iBOS

Broadband Operating System

iBOS 7.1 Configuration Guide

waystream

iBOS 7.1 Configuration Guide

iBOS 7.1 Configuration Guide

WS-DOC-iBO7.1-CONFIG. Published May 16, 2018.

Copyright and Legal notice

Copyright 2012-2017 Waystream AB. All rights reserved.

iBOS, the ASR Bootloader, program models and other software software content and this documentation ("the Intellectual Property Rights") are protected by the Swedish Copyright Act (Sw: Upphovsrättslagen) and, if applicable, the Swedish Patents Act (Sw: Patentlagen). All and any copying and distribution of the Intellectual Property Rights, without authorization by Waystream is prohibited. The prohibition includes every form of reproduction and distribution.

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, information is subject to change without notice. Waystream assumes no liability for damages incurred directly or indirectly from errors, omissions or discrepancies between the software and this document.

iBOS, ServiceEngine, MS4000-series switches and ASR are trademarks of Waystream.

All other trademarks, service marks and trade names are proprietary to their respective owner.

Purchasers, licensees and users accept and acknowledge that the products contain components (including components carrying certain firmware) and combinations of components that constitute trade secrets protected by Waystream or its partners. Purchasers, licensees and users warrant that the delivered products will not be opened or dismantled, copied, altered or in any other way modified. Furthermore, purchasers, licensees and users agree not to attempt to reverse engineer, disassemble, modify, translate, create derivate works, rent, lease, loan, or without written permission distribute or sublicense the software, in whole or in part.

The products and its hardware, firmware and software, including technical data, may be subject to EU and U.S export control laws, including the U.S Export Administration Act and its associated regulations and the International Traffic in Arms Regulations administerd by the US Department of State, and may be subject to export or import regulations in other countries. Purchasers and licensees agree to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import hardware, firmare and software.

Purchasers and licensees are not entitled to, and Waystream is not in any event liable to pay, compensation for damages which delivered products or software has caused to other property or to persons or any other consequential damages, including but not limited to loss of profit, loss of production or any other indirect damages.

Contents

Introduction 11

Conventions used in this Manual 12

Overview 13

What is iBOS 14
iBOS Interfaces 14
Layer 2 Interfaces 14
Layer3 Interfaces 14
VLAN Interfaces 15
Loopback Interface 15
System0 15
Downstream0 15
Remote-vid 16
Ethernet0 16
iBOS Routing Engine 16
Routing Information Base (RIB) 17
Distance and Metric 17
iBOS Forwarding 17
Forwarding Table (FIB) 18
About the iBOS Command Line Interface 18
Connecting to the CLI 18
The CLI Prompt 19
Changing to Admin Mode 19
Changing to Configuration Mode 19
Contexts 20
Exit from Configuration Mode 20
Exit from Admin Mode 21
Exit from User Mode 21
Understanding the Command Line Interface 21
Command History 21
Command Completion 21
Command Help 21
Typing Commands 22
Basic Command Line Options 22
Piping Output 24
Storing Configuration 25
Rebooting the System 25
User Accounts and Remote Access 26
Managing Local User Accounts 26
Remote Login 26
Support for Telnet Clients 27
Basic CLI Commands 28

System Management 31
System Commands 32
Setting the System Hostname 32
Setting Domain Name Server 32
Setting System Domain Name 33
Setting System Configuration Registers 33
Authentication, Authorization and Accounting (AAA) 34
Configuring AAA. 34
DHCP RADIUS Authentication 35
DHCP Server 35
Radius Server 35
DHCP Relay Agent 35
Configuring DHCP-RADIUS with BNG 36
Configuring DHCP-RADIUS with BNG and DHCP Snooper 37
Policy-based RADIUS 39
RADIUS with Redundancy 44
Environment Monitor 55
File System 55
System Time and Network Time Protocol (NTP) 56
System Time 56
Setting System Clock 56
Setting Time Stamp Format 56
NTP 57
Setting System Time with NTP 57
Logging 57
Logging Levels. 58
Logging Destinations 58
Logging to a Remote Host 59
Local Logging 59
Remote Management 61
Simple Network Management Protocol 62
Network Management Systems 62
The Agent 63
The Structure of Management Information 63
The Management Information Base 63
The History of SNMP 64
SNMPv1 64
SNMPv2 64
SNMPv3 65
Configuring SNMP 65
Configuring Agent Parameters 65
Configuring View Access 67
Configuring SNMP-Groups 67
Agent Information 69
CSAR Protocol 70
Configuring CSAR 70
PFDP Protocol 70
Protocol Overview 70
Protocol Operation 71
PFDP Firmware Upgrade 74
Active PFDP CPE Configuration 75
Active PFDP Support for DRG CATV 76
Error Disable 77
Link Layer Discovery Protocol 81
Purpose 81
Frame Structure. 81
Destination MAC 81

Source MAC 82
Ethertype 82
Mandatory TLV's 82
Optional TLV's 82
Special TLV (End of LLDPDU) 83
Frame Check Sequence 83
Operating Modes 83
LLDP Timers 83
LLDP Configuration 84
Monitoring and Maintenance Commands 84
Basic LLDP Configuration Example 85
Netflow 88
Flow Contents 88

Layer 2 Specific Configuration 89

Layer 2 Interfaces 90
Physical Interfaces 90
QoS 90
Layer 2 Logical Interfaces 90
VLAN Interfaces 90
Basic VLAN Configuration Example 91
L2TPv3 Tunnel 95
Configuring L2TP 100
L2TP Redundancy 102
Link Aggregation 102
Forwarding 103
Layer 2 Unicast Support 103
UNI Isolated Ports/VLANs 103
MAC Forced Forwarding 104
Layer 2 Multicast Support 104
Access Control of Multicast Groups 105
Spanning Tree 106
Bridge Roles 106
Election Mechanism 107
STP Ports and States 107
STP Topology Changes 108
RSTP 111
RSTP Port States 111
RSTP Port Roles 111
RSTP Enhancements 112
Multiple Spanning Tree (MSTP) 116
Configuring Spanning Tree 116
Spanning Tree Configuration Examples 117

PPP over Ethernet (PPPoE) 119

PPPoE Intermediate Agent 120
PPPoE Server 121
Configure PPPoE Access Concentrator 122
PPPoE Authentication 124
Configuring PPPoE Address Assignment 126
Configure PPPoE Client IP Address 126
Configure DNS Server for PPPoE Clients 128
LCP Negotiation 129
LCP Echo Requests 130
Collect PPPoE Server Statistics 131
PPPoE Data Forwarding 132
PPPoE Session Parameters 133
Terminate PPPoE Sessions 134
PPPoE Redundancy 134

Layer 3 Specific Configuration 137
Layer 3 Standard System 138
Layer 3 Customer Interfaces for Residential Services 141
Layer 3 Services 142
Service Definition 142
Service Templates 142
IPv4 145
IPv4 Addressing 146
iBOS BOOTP Client 146
DHCP 146
DHCP Operation 147
DHCP Relay Agent 147
Configuring BOOTP/DHCP Relay 148
BOOTP/DHCP Relay for BECS 148
DHCP and PXE Relay 149
ARP 150
Understanding ARP 150
Proxy ARP 151
Controlling ARP Traffic in iBOS 151
Static ARP Entries 151
Sharing Subnets 152
ARP Configuration Example 153
IPv4 Routing 154
OSPF v2 154
Understanding OSPF 154
Link State 154
Shortest Path Tree Algorithm 154
Network Hierarchy 155
Subnets and Authentication 155
Protocol Implementation 155
Packet Types 155
Link-state Advertisement 156
Areas 157
Virtual Links 158
Configuring OSPF 159
Area Configuration 159
OSPF Configuration Example 159
PIM 160
Definitions 161
Rendezvous Point (RP) 161
Designated Router (DR) 161
Multicast Routing Information Base (MRIB) 161
RPF Neighbour 161
Upstream 161
Downstream 161
PIM Routing 161
Phase One: RP Tree 162
Joining and Pruning 162
Source Sending Data 162
Phase Two: Register Stop 163
Register Stop Message 163
Phase Three: Shortest Path Tree 163
Multicast Traffic from Two Directions 164
Finding the RP 164
Static RP 164
Bootstrap Router 164
Bootstrap Router Overview 165

PIM-SM Overview 165
PIM Rules for Mapping 165
PIM-SM BSR Mechanism 166
Overview of Bootstrap and RP Discovery 167
Administratively Scoped Multicast and BSR 168
Configuring PIM 168
Interface Configuration 168
PIM Configuration Example 169
IGMP 170
Querier 170
General Query 170
Membership Report 171
Joining Groups 171
Leaving Groups 171
Router Response to Leave Group 171
Changes to Version 1 172
IGMPv3 Message Format 172
Packet Formats for IGMPv3 Messages 173
IGMPv3 Snooping 174
SSM Mapping and IGMP Reports 174
Extended Output 175
IPv4 Forwarding 176
GRE Tunnels 176
Virtual Router Redundancy Protocol (VRRP) 177
IP MTU Size 178
DNAT Forwarding 179

IPv6 181

IPv6 Overview 182
SLAAC 182
Stateful Address Assignment 182
IPv6 Addressing 183
Representation of IPv6 Addresses 184
Address Type Identification 184
Unicast Addresses 185
The Unspecified Address 185
The Loopback Address 185
IPv4-Compatible IPv6 Address 185
IPv4-Mapped IPv6 Address 186
Link-local Addresses 186
Site-local IPv6 Unicast Addresses (Deprecated) 186
Anycast Addresses 186
Multicast Addresses 187
Pre-defined Multicast Addresses 189
Required Addresses 189
DHCPv6 190
DHCPv6 Overview 190
Two Message Exchange 191
Four Message Exchange 191
DHCPv6 Operation 191
DHCP Unique Identifier (DUID) 192
Identity Association (IA) 193
DHCP Client-Server Operation 193
Relay Agent Behaviour 195
The Address Lease 196
DHCP Options 196
Neighbour Discovery 197
Purpose of Neighbour Discovery 198
Neighbour Discovery Packet Types 198

Router Advertisements 199
Neighbour Solicitation 199
Stateless Address Auto Configuration 199
SLAAC Automatic Configuration 200
SLAAC Operation 200
Creating Link-Local Addresses 201
Using the MAC Address 201
Duplicate Address Detection 202
Creating Global Addresses 203
IPv6 Addressing Commands 203
IPv6 Routing 204
IPv6 Static Routing 204
OSPFv3 204
OSPFv3 Commands 204
OSPFv3 Configuration Example 205
IPv6 Forwarding 205
IPv6 Unicast 206
IPv6 GRE Tunnels 206
IPv6 DHCP Relay 206

Configuring MPLS 207

What is MPLS 208
MPLS Forwarding 209
MPLS Standards 211
MPLS LDP 211
LDP Adjacencies 211
LDP Session 212
Forwarding Equivalence Class 212
Label Generation 212
Label Distribution Modes 212
Label Retention Mode 212
LSP Control 213
LDP Configuration Examples 213
Enable Label Switching 213
MPLS Layer-3 VPNs 218
Requirements for MPLS L3 VPN 218
MPLS VPN Topology 219
MPLS-VPN Routing Process 220
Configure MPLS Layer 3 VPN 220
Establish Connections between PE Routers 221
Configure PE Routers as BGP Neighbours 222
Create VRF 223
Associate Interfaces to VRFs 223
Configure VRF—RD and Route Targets 223
Configure CE Neighbour for the VPN using BGP and OSPF 224
Verify the MPLS-VPN Configuration 225
MPLS Layer 2 VPN 227
EoMPLS 227
Pseudowires 228
Virtual Circuits 228
Attachment Circuits 229
Traffic Control using a Policy Map on Pseudowires 229
Targeted LDP 230
Configuring EoMPLS 230
EoMPLS Configuration Example 231
MPLS Traffic Class 233
Configure MPLS Traffic Class Trust 234
Configure MPLS TC to Priority Mapping 234
Configure MPLS TC to IP DSCP Mapping 234

Configure MPLS TC to VLAN PCP Mapping 235
MPLS Traffic Class Example 236

BGP Configuration 239

Introduction to BGP 240
BGP Routing 241
Internal BGP Routing 241
External BGP Routing 242
Comparison BGP and IGP 242
BGP Synchronization 242
BGP Filtering of Routes 243
BGP Parameters. 243
AS Paths 243
BGP Route Origin 244
BGP next-hop 244
BGP local-preference 245
BGP Multi Exit Discriminator 246
BGP Community 246
Selecting the Best Route 248
Advanced iBGP 248
Router Reflectors 248
BGP Confederation. 249
Asymmetrical Traffic 250
MD5 Verification 250
Graceful Restart. 251
Example Configuration for Graceful Restart and Graceful Reset 252
BGP Peer Reset and Graceful Restart Events 253
BGP and MPLS. 253
Inter AS BGP MPLS VPNs 253
CE-PE Interaction 253
PE-PE Interaction 254
MPLS LSP 254
BGP Messages 254
OPEN Messages. 254
UPDATE Messages. 254
NOTIFICATION Messages 254
KEEPALIVE Messages. 255
BGP Configuration Examples 256
Enable BGP Routers in the Same Autonomous System 257
Enable BGP Between Different Autonomous Systems. 257
Configure AS Path Filtering 258
Configure Route-Map. 258
Redistribution of Routing into BGP 260
Configuring Route Reflector 260
Configure Route-Maps 261
Configure Route Reflectors 263
Configure Route Reflectors to Reduce iBGP Mesh Inside AS 263
Configure Route Reflector to be Transparent to Clients 264
Configure Route Server 266
Configure a Route Server with Multiple Views 269
Configure BGP Confederations 271
Configure BGP Confederation with Same Member AS 271
Configure Confederation between Members and Non-Members 272
BGP Authentication 274
Enable eBGP Multihop 275
Enable Peer Groups 277
Redistribute with OSPF 278
Add Multiple Instances of Same AS to Administer Route Selection 279
Remove Multi-Exit Disc Attribute from Update Messages 281

Configure BGP Four-Byte AS 282
4-Octet ASN Capability Enabled on R1 and R2 283
4-Octet ASN Capability Enabled on R1 and Disabled on R2 284
Configure BGP Extended Community Attribute 285
Configure Extended Community with 2-Byte ASN 285
Configure Extended Community with 4-Byte ASN 288
Configure IPv6 Route Refresh. 291
Configure Nexthop Tracking. 294
Configure Nexthop Tracking Delay Timer 297
Configure BGP Graceful Reset 300
Configure BGP-MPLS Graceful Restart 301
Configure BGP Distance 306
Configure BGP Weight per Peer 307
Configure BGP-OSPF as PE-CE Protocol for VPNs 309
BGP Multipath for IPv4 and IPv6 312
Configure Multipath iBGP for IPv4 313
Configure Multipath iBGP for IPv6 316
Configure Multipath eBGP for IPv4 320
Configure Multipath eBGP for IPv6 323
BGP Graceful Shutdown 326
Configure BGP Graceful Shutdown on Interface 326
Configure BGP Graceful Shutdown on Router 328
Extended BGP 330
Enable iBGP Peering Using Link-local Address 332
Enable eBGP Peering Between Different Autonomous Systems 334
Configure BGP4+ Route-Map 335
BGP4+ Route Reflector 338
BGP4+ Confederations. 340
Configure BGP4+ Graceful Restart 342
Configure BGP4+ Distance. 344
Configure BGP4+ Graceful Reset 346

Configuring VRF 349

Simple VRF Implementation 350
VRF Configuration Example 351
Full VRF Implementation 354
Multicast VRF. 357
Configure PIM-SM in VRF. 357
Check PIM-SM VRF Configuration 363
Check PIM Interface and Status 363
Check PIM Neighbour and Status 364
Check PIM RP Mapping Status 365
Check PIM Multicast Route 365
Check Multicast Forwarding Status 368
Check PIM Local Member Status 368
Configure PIM-SSM in VRF with Downstream Map 369
Configure PIM-SSM for VRF 370
Configure SSM Group Access-list for VRF 370
Check PIM-SSM for VRF Configuration 370
Check IGMP SSM Map 374
Check PIM Multicast Route Status 376
Check Status of iGMP Groups 376
Configure PIM-SM in VRF with SSM Map 377
Configure PIM-SSM on VRFs with Same S,G 377
Check igmp ssm-Map Status 382
Check PIM Multicast Route Status: 382
Check PIM Local Member Status: 384
Check IGMP Group Status 384
VRF Commands used by iBOS 385

Configuring IS-IS 387
Introduction to IS-IS 388
Basis IS-IS Configuration 389
Enable IS-IS on an Interface 390
Set Priority 391
Redistribute Routes into IS-IS 393
Configure Metric 394
L1-L2 Area Routing with Single Instance 397
L1-L2 Area Routing with Multiple Instances 399
Route Leaking 402
Route Summarization 403
Configure IS-IS Distance 404
Configure Overload Bit 407
Configure Passive Interface 410
IS-ISv6 Configuration 413
Basic IS-ISv6 Configuration Example 413
Set IS-ISv6 Priority 414
Redistribute Routes into IS-ISv6 416
Configure Metric for IS-ISv6 417
L1-L2 Area IPv6 Routing 420
L1-L2 IPv6 Routing with Single Instance 421
L1-L2 Area IPv6 Routing with Multiple Instances 422
Configure IS-IS IPv6 Distance 424

Access-lists 427
Understanding iBOS Access-lists 427
Access-list Configuration Example 428

Traffic Control 431
Rate-limiting 432
Policy Mapping for VLAN and Ports 432
Automatic Creation of Policy-Map for Snooped DHCP Clients 432
VLAN Translation 432
Policy Map EtherTypes 434
Load Balancing 435
Policy Routing 435
Configure PBR 436
Flow Inspection 438
Configure Flow Inspection 438
Flow Inspection for Troubleshooting 441

Extended iBOS 445
Realtime Protocol Monitoring 446
Realtime Protocol Monitoring Logging 446
Scripts 447
802.1X Authentication Script 447
DHCP Relay Script 448
DHCPv6 Relay Script 449
DHCPv6 Snooper Script 450
PPPoE Intermediate Agent Script 450
Static RADIUS Script 451
Mirrored Traffic 451
Lawful Interception 452
Overview 452
Architecture 452
LI User Configuration and Global LI Settings 453
LI Task Configuration [x1 Interface] 454
Packet Interception and Reporting [x3 interface] 455
State of LI Task 456

SNMP Cold/Warm Start Trap Notification 456
Configuration Examples 456
USE CASE 1 457
USE CASE 2 458
Selective Q-in-Q 460

iBOS Licenses 461

Pay as you Grow 461
Grace Period 461
Verify Licenses 462
Fetch your License Files. 463
Generate Licenses 464
Installing Licenses 465

Appendix A – Copyright Statements 467

NetBSD 467
Net-SNMP 468
Internet Standards Document Copyright. 469

Introduction

Waystream hardware runs the intelligent broadband operating system (iBOS).

This manual is intended for use by individuals who are responsible for configuring Waystream hardware using the iBOS CLI in a broadband network.

To understand this manual, the user should have expert knowledge of broadband networking, data communication protocols and be familiar with Waystream hardware products.

The following is described in this document:

- “Overview” on page 13
- “System Management” on page 31
- “Remote Management” on page 61
- “Layer 2 Specific Configuration” on page 89
- “PPP over Ethernet (PPPoE)” on page 119
- “Layer 3 Specific Configuration” on page 137
- “IPv4” on page 145
- “IPv6” on page 181
- “Configuring MPLS” on page 207
- “BGP Configuration” on page 239
- “Configuring VRF” on page 349
- “Configuring IS-IS” on page 387
- “Access-lists” on page 427
- “Traffic Control” on page 431
- “Extended iBOS” on page 445
- “iBOS Licenses” on page 461
- “Copyright Statements” on page 467

Conventions used in this Manual

The following typeface conventions apply

Table 1 Typeface Conventions

Typeface	Description
show commands	Denotes iBOS commands
iBOS Configuration Guide	Denotes a document title
stdio.c	Denotes a file name
show	Input from keyboard
ibos#	Output to screen

The following command syntax conventions apply:

Table 2 Command Syntax Conventions

Character	Description
[]	Denotes optional keywords or arguments
{keyword | keyword}	Denotes mandatory keywords separated with |, you must select one of the keywords.
<>	Denotes user entered data
a.b.c.d	Denotes standard 4-octet ip address format

Overview

The intelligent broadband operating system (iBOS) manages access hardware products from Waystream. This software customizes the operational functionality depending on the network infrastructure. The versatility of iBOS allows Waystream access products to be deployed in many different types of broadband networks.

The iBOS operating system is described in:

- "What is iBOS" on page 14
- "iBOS Interfaces" on page 14
- "iBOS Routing Engine" on page 16
- "iBOS Forwarding" on page 17
- "About the iBOS Command Line Interface" on page 18

What is iBOS

The functionality of iBOS can be split into the following areas:

- "iBOS Interfaces" on page 14
- "iBOS Routing Engine" on page 16
- "iBOS Forwarding" on page 17

iBOS Interfaces

An interface is a physical or logical device that is used for packet forwarding.

Physical interfaces are TenGigabitEthernet, GigabitEthernet and FastEthernet ports, and the Management port (called ethernet0).

Logical interfaces are VLAN interfaces, tunnel interfaces and system loopback interfaces.

Layer 2 Interfaces

Some interface types, such as TenGigabitEthernet ports, can operate both as Layer 2 and Layer 3 interfaces in iBOS.

Layer 2 interfaces operate on Layer 2 of the OSI model. An interface configured as a Layer 2 interface does not perform IP routing. Only the content of the packet Layer 2 header is considered when traffic is forwarded.

Layer 2 interfaces in iBOS must be connected to a Layer 3 interface in order for routed IP communication to occur. For physical interfaces, this is typically done by assigning the interfaces to be a member of a VLAN. If multiple interfaces are assigned to the same VLAN, they are also in the same broadcast domain and thereby any Layer 2 traffic between hosts connected on those interfaces is possible. Hosts can communicate using IP between themselves but if the router is to forward traffic to a destination in another broadcast domain, a Layer 3 interface must be created and configured.

Layer3 Interfaces

Simply speaking, a Layer 3 (L3) interface in iBOS can be described as any interface where an IP address can be configured. Both physical and logical interfaces can be Layer 3 interfaces in routers. For switches, the model and version of iBOS determines if an interface can be a L3 interface. Traffic is routed between L3 interfaces that belong to the same Virtual Router Forwarding (VRF) instance. By default, L3 interfaces in a system belong to the same VRF and therefore IP routing is automatically enabled between interfaces with a configured IP address.

In addition to configuring one or multiple IP addresses, the L3 interface also has configuration commands for activating routing protocol operations, multicast traffic, and other features on the interface.

VLAN Interfaces

A VLAN interface is a logical interface used to set the routers IP address on a VLAN. A VLAN interface number corresponds to the VLAN id configured on the Layer 2 interface. Any host connected on a Layer 2 interface that is a member of the VLAN may send traffic to the router and have it forwarded to a different broadcast domain.

There are no VLAN interfaces by default in iBOS, but they are easily created with the **interface vlan** configuration command.

If all Layer 2 interfaces associated with a VLAN are down (i.e. no link), the VLAN interface will also be down.

Loopback Interface

A loopback interface is a logical interface that exists “in the middle” of the system. Unlike VLAN interfaces, a loopback interface is not connected to any Layer2 physical interface.

There is only one host connected on a loopback interface, and that is the iBOS system itself.

Loopback interfaces are useful as the source and destination for traffic directed to the system itself (Telnet sessions, BGP source/destination address for peering, etc) as these interfaces are always up and unaffected by any network failures.

If the loopback interface is redistributed by routing protocols and used as the source interface for the routing protocols, and the iBOS router is connected by two or more links to other parts of the network, the connection will continue to run in the event of a link outage. This is due to that rerouting will find an alternative route between the loopback interface and the remote peer. If a VLAN interface is used instead, the link outage may cause that VLAN interface to go down which will also close the protocol connection.

System0

There is a special internal loopback interface in iBOS called *system0*. The *system0* interface has the traditional loopback interface address of 127.0.0.1 and can not be used as a source or destination interface. It exists for internal reasons only.

Downstream0

A key feature of iBOS is its ability to allow multiple clients connected on multiple physical or logical interfaces in the same or in different adjacent systems to share the same IP subnet.

Clients are either connected on physical interfaces or are present within VLANs that are bridged to the iBOS device. It is possible to configure an IP address directly on the interface of the client if it supports L3 operation. In addition, when **arp proxy** is enabled, it is also possible to not configure an IP address on the interface and still have iBOS respond to any ARP requests.

When the DHCP relay agent is configured to relay DHCP packets from such an interface, and to replace the server-id information so that all DHCP traffic goes

through the local system, the relay agent will install the default-gateway address assigned by the DHCP server to a requesting client as an ip address on the *downstream0* interface. When clients make an ARP request for their default-gateway address, the proxy ARP function in iBOS makes it respond if the address is present on the downstream0 interface.

The *downstream0* interface provides additional functionality for multicast. Interface configuration commands allows all IGMP traffic received on client interfaces to be redirected to the downstream0 interface. For the iBOS routing engine, it appears as if all clients are present on the *downstream0* interface, which reduces the state and complexity involved in providing multicast services to hundreds of interfaces (see section on remote-vid interfaces below).

Remote-vid

Remote-vid interfaces are L3 interfaces that represent a VLAN on a port. It is possible to create interfaces representing double-tagged VLANs on a port. The remote-vid interface is typically used for customer-VLANs and allows service related configuration for traffic to clients in that VLAN. VLANs associated with a remote-vid interface are not bridged; this means that the same VLAN number can be used on multiple physical ports without any L2 bridging occurring when the VLAN is associated with the remote-vid interface.

Ethernet0

In some products, there is a physical port reserved for management. iBOS designates this port as the *ethernet0* management interface. This interface operates as a normal Layer 3 interface in the system and is connected directly to the system CPU. The performance and features available on the interface are therefore limited and *ethernet0* should only be used for out-of -band (OOB) management.

OOB management offers a dedicated connection to the system that is separate from the actual network that the system runs on. This ensures a single entry point for the management interface.

iBOS Routing Engine

The iBOS routing engine is a robust and standards-based Layer3 carrier class routing software solution. It is modular, meaning that different protocols are implemented as separate processes that interact with a central module and the system kernel an subsystems.

Routing capability is a fundamental feature of any router. The iBOS routing engine is responsible for all routing protocol exchanges with neighbouring routers. In addition, the routing engine controls the local routing (static and connected networks). The iBOS routing engine handles both unicast and multicast routing protocols, such as OSPF and PIM.

Configuration in the CLI controls the operation of the routing protocols.

Routing Information Base (RIB)

The routing engine maintains a routing information base (RIB). The RIB contains a combination of routing information assembled by the various routing protocols. This means that a router with configured static routes and OSPF running will have a combination of the static routes and the OSPF learnt routes in the appropriate RIB.

Both unicast and multicast routing protocols use the RIB for their routing decisions.

Distance and Metric

Sometimes the same route may be learnt by different routing protocols, or even from different sources by the same routing protocol. The router needs to have a method of determining which of the possible routes is preferred.

The distance parameter that can be configured on static routes and elsewhere - and which is displayed for routing information - determines how information learnt by a certain protocol (static routes included) is weighted compared to the same information learnt by another protocol. The lower the distance value, the more the route is preferred. The following table shows the default protocol distance values for iBOS routing protocols.

Table 3 Routing Protocol Distance

Protocol	Distance
Directly connected	0
Static routes	1
Kernel routes	Inherits distance of type of route (static/connected etc.)
OSPF routes	110
BGP distance	TBD
IS-IS	TBD

A kernel route is a route learnt by means other than through the routing engine, for instance by automatic actions from other subsystems. Such routes are typically static or connected routes and have the same distance as manually configured static or connected routes unless the subsystem creating the kernel route sets a different distance value.

The metric value is protocol related. The metric value of a route determines how good that route is compared to other routes learnt by the same routing protocol. Two equal routes learnt by OSPF may have different metric values, and if so, the OSPF protocol's way of handling metric determines which of the routes is selected as the active route.

iBOS Forwarding

Forwarding is the process in which iBOS receives a packet, looks up the packet destination in the forwarding table, determines any actions to take on the packet in terms of header rewrites, queueing, priority, determining the egress interface and then transmitting the packet out the egress interface.

Forwarding Table (FIB)

The iBOS kernel contains an optimised version of the routing information base. Each protocol maintains a set of parameters for routes learnt through that protocol. As described in the previous section, this includes metric values, distance parameters and many other parameters that are essential for the selection of the best route to each destination.

However, once that selection has been made, the router requires a subset of information to carry out the actual forwarding. This includes the layer3 destination (the route), the nexthop address, the physical address of the nexthop and the interface over which to transmit the packet to the destination.

Since routes are learned and constantly evaluated against each other, the routing engine creates and updates entries in the forwarding table. The forwarding table (or forwarding information base, FIB) only contains the information required for traffic forwarding.

About the iBOS Command Line Interface

The iBOS command line interface (CLI) meets industry standards for networking operating systems. Commands are written as English words or phrases, and are sometimes combined with numbers, IP addresses and other variables.

The iBOS CLI is described in the following:

- "Connecting to the CLI" on page 18
- "Understanding the Command Line Interface" on page 21
- "Basic Command Line Options" on page 22
- "Rebooting the System" on page 25
- "User Accounts and Remote Access" on page 26

Connecting to the CLI

To connect to a machine using IBOS, use a serial cable to the console port. Use a PC or other computer with a terminal software such as HyperTerminal. Configure the terminal software to connect directly on the serial port at 115200 bps with the 8N1 settings for data bits, parity and stop bits.

Once the boot sequence is completed, the username prompt appears on the console. If no user account has been set up, simply press enter after the username and subsequent password prompt to gain access to the system.

Example 1 The Login Prompt

```
username:
password:
asr>
```

You have now entered the iBOS CLI.

The CLI Prompt

The CLI will present the user with a prompt, which tells the you the system name and user mode. If the system is not configured, the default prompt will be displayed.

Example 2 The ASR Default Prompt

```
asr>
```

The prompt gives the user the system name (asr) and the user mode. In this case the prompt displays the character '>', which indicates that the system is operating in user mode.

There are three different user modes.

Table 4 iBOS CLI Modes

Prompt	User Modes
>	Default user mode. The available commands are generally "show and tell" commands. The user cannot change configuration or affect system operation.
#	Admin mode. The user can issue all commands, including commands that re-start the system or clear important tables, such as the ARP table. The user may access and alter configuration information.
(config)#	Configuration mode. In this mode the user can change system configuration.

To configure the command line prompt, use the **service terminal prompt format** command.

There is no default security between the user modes, as the user freely may change mode to admin mode with the **enable** command and exit it with the **disable** command.

If desired, the network administrator can set the system up to use remote authentication, as well as local usernames and passwords, in order to restrict access to the system. Password verification for access to admin mode can also be enabled.

Changing to Admin Mode

In order to set up the system, it is necessary to enter the administrator (admin) mode. This is done with the **enable** command.

Example 3 Changing to Admin Mode

```
ibos> enable
ibos#
```

The change of the prompt from '>' to a '#' indicates that the system is now in admin mode.

Access to the admin mode can be password protected, using the **enable secret** configuration command to set the password. If successful, the user must enter the password after running the enable command in order to get into admin mode.

Changing to Configuration Mode

Configuration mode can only be accessed from admin mode. The command to enter configuration mode is **configuration terminal**.

Example 4 Changing to Configuration Mode

```
ibos# configuration terminal
ibos(config)#
```

The addition of '(config)' to the prompt indicates that the user now is in configuration mode.

Contexts

In configuration mode, configuration commands are entered one by one. Configuration statements are grouped into contexts of related commands.

For instance, all configuration commands related to a physical interface are grouped into the interface configuration context.

It is not possible to enter a configuration command that belongs to the router ospf context in, for example, the interface context. Only commands belonging to the root context are available to the user.

When moving into a new context, the prompt will change to reflect this.

Example 5 Contexts

```
ibos(config)#interface fastethernet1
ibos(config-if-fa)#exit
ibos(config)#router ospf
ibos(config-router-ospf)#
```

The example shows how the prompt changes as the user moves from interface context into router ospf context.

As indicated, the **exit** command moves back to the root context.

Exit from Configuration Mode

At anytime while in configuration mode, the user may revert from configuration mode to admin mode by issuing the **end** command. The CTRL-C and CTRL-Z key combinations can also be used for this purpose.

Example 6 Exit from Configuration Mode

```
ibos(config-router-ospf)#end
ibos#

ibos(config-router-ospf)#^Z
ibos#

ibos(config-router-ospf)#^C
ibos#
```

The prompt indicates that the user is back in admin mode.

Note: *CTRL-C aborts and exits from configuration mode immediately. CTRL-Z first attempts to execute any command typed in on the line before exiting from configuration mode.*

Exit from Admin Mode

The user may revert from admin mode to default mode by issuing the **disable** command.

The user may also log out of the system by issuing the **quit**, **exit** or **logout** command.

Exit from User Mode

There is only one exit from default mode, and that is to log out of the system. This is done by issuing the **quit**, **exit** or **logout** command.

Understanding the Command Line Interface

Command History

All commands that have been typed in are stored in a history buffer, accessible with the up and down arrow keys. This allows easy repetition of commands. There is one history buffer for configuration commands and one for executable commands. Certain commands may also have individual history buffers.

Command Completion

Command names may be abbreviated to their shortest unique form. The **show ip route** command for example may be shortened to **sh ip rou**, which is sufficient for the router to uniquely identify which command to execute.

To make the router complete the current command word the TAB key can be pressed. Typing **sh** and then pressing the TAB key will cause the router to complete the word to **show**.

If there are multiple matching command words, the word will be expanded to the longest mutual character and a list of available command words is displayed.

Using the TAB key enables the user to quickly find the correct command - even if the final command sentence has many words and arguments.

Command Help

At any time it is possible to press the '?' key and have all available commands presented with a short help text.

Example 7 Using the '?' key for help

```
ibos# show ip ?
  bootp        BOOTP client information
  connections  Active Internet connections
  dhcp         Dynamic Host Configuration Protocol
  flow-cache   Display IPv4 flow cache
  forwarding   IP forwarding table
  icmp         Internet control message protocol
  igmp         IGMP information
  interface    IP interface status and configuration
  mroute       PIM Tree Information Base
  multicast    IP multicast
  ospf         Open Shortest Path First routing protocol
```

```
  pim          Protocol Independent Multicast (PIM)
  protocols    IP routing protocol process parameters and statistics
  relay        Display information on relay configuration
  route        IP routing table
  tcp          Transport control protocol
  udp          User datagram protocol
  <cr>         (Executes 'show ip')
ibos# show ip
```

In the above example, pressing the '?' key after typing **sh ip** will display all **show ip** commands.

Typing Commands

Commands are typed in on the command line interface prompt. A command consists of one or multiple unique keywords to identify the command and any options. When the command has been entered the <ENTER> key is used to execute the command. The result takes effect immediately. It is possible to type multiple commands on the same command line if each command is separated from the next with a semi-colon, ';'. The command and all of its options and parameters is written normally, then a semi-colon followed by the next command and its parameters and options.

Basic Command Line Options

The command line interface accepts different types of control and output manipulation commands. For cursor control and positioning, the following tables list iBOS CLI key shortcuts.

Table 5 Cursor Positioning

Key	Key Code	Function
CTRL-A	1	Position cursor at beginning of line
CTRL-B	2	Step cursor back
CTRL-E	5	Position cursor at end of line
CTRL-F	6	Step cursor forward
CTRL-L	12	Vertical tab
Enter	13	Execute command
ESC-B	ESCB	Move cursor back one word
ESC-F	ESCF	Move cursor forward one word
Right arrow	ESC[C	Step cursor forward
Left arrow	ESC[D	Step cursor back

Table 6 Text Deletion

Key	Key Code	Function
CTRL-D	4	Delete character under cursor
Backspace	8	Delete character to the left of cursor
CTRL-K	11	Delete from cursor to end of line
CTRL-U	21	Delete from cursor to beginning of line
CTRL-W	23	Delete word to the left of cursor
Del	127	Delete character to the left of cursor
Meta-Backspace	136	Delete word to the left of cursor
ESC-Backspace	ESC8	Delete word to the left of cursor
ESC-D	ESCD	Delete word to the right of cursor
ESC-K	ESC[K	Delete from cursor to end of line

Table 7 Special Purpose

Key	Key Code	Function
CTRL-C	3	Abort command and exit from context
TAB	9	Expand command or display options
CTRL-N	14	Step down in history table
CTRL-P	16	Step up in history table
CTRL-Z	26	Execute command and exit from context
?	63	List commands or display options
INS	126	Toggle insert mode

Key	Key Code	Function
Up arrow	ESC[A	Step up in history table
Down arrow	ESC[B	Step down in history table

Piping Output

Some commands, especially the show command, display a lot of information. To manipulate the output, iBOS provides commands to format output.

Commands can be piped to a set of utilities. Piping means that the vertical-bar '|' character is added between the issued command and the utility.

Grep

The **grep** command is a popular UNIX utility for looking for string matches in files and output.

Example 8 Using Grep

```
ibos# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       CL - Client
       * - candidate default

Gateway of last resort is 10.0.0.254 to network 0.0.0.0

Codes   Destination
------------------------------------------------------------------
-------------
S*      0.0.0.0/0 [1/0] via 10.0.0.254, ethernet0
C       10.0.0.0/24 is directly connected, ethernet0
C       10.40.10.1/32 is directly connected, loopback1
S       10.40.180.0/24 [1/0] via 10.40.10.1, loopback1
C       127.0.0.0/8 is directly connected, system0
S       192.168.1.0/24 [1/0] via 10.0.0.254, ethernet0
ibos# show ip route | grep ethernet
S*      0.0.0.0/0 [1/0] via 10.0.0.254, ethernet0
C       10.0.0.0/24 is directly connected, ethernet0
S       192.168.1.0/24 [1/0] via 10.0.0.254, ethernet0
ibos#
```

In the above example, the second execution of the **show ip route** command is piped to **grep** where the argument 'ethernet' means that **grep** should discard all lines of output not matching the regular expression 'ethernet'. In this case, 'ethernet' matches the word ‘ethernet0’, which is found in the three lines of output

Include, exclude, begin, stop

Some of the common functions of **grep** are implemented as separate commands.

Include includes any line matching the regular expression.

Exclude excludes any line matching the regular expression.

Begin starts the output on the line matching the regular expression.

Stop stops the output on the line matching the regular expression.

Pausing Output

The **more** command is another popular UNIX utility for paging through text one screen full at a time. For example, when displaying the configuration with **show running-configuration**, the output will probably not fit on the screen. By piping the command through **more**, the CLI will only display the output one page at a time, waiting for the user to press the space bar to display further output.

Example 9 Pause Output using more

```
ibos# show running-configuration | more
```

Storing Configuration

Waystream routers has an onboard flash, where system images and configuration are stored.

If configuration is not stored, a reboot or power cycle of the system will revert the system to the previously stored – if any – configuration.

To save the configuration, use the **write** command.

Example 10 Writing System Configuration

```
ibos# write
Wrote 568 bytes compressed configuration (1895 bytes uncompressed)
ibos#
```

Rebooting the System

The system is rebooted if the power is turned off and on. The system may also be rebooted by a user in admin mode by using the **reload** command.

Reloads can be scheduled to occur at a specified time or after a specified interval.

Example 11 Delayed Reload of the System

```
ibos# reload in 30

Message from test at Wed Oct 8 21:11:10 2003
System will be reloaded in 30 seconds!
ibos# reload cancel

Message from test at Wed Oct 8 21:11:13 2003
Pending reload aborted.
ibos#
```

User Accounts and Remote Access

The factory default configuration of the system contains no user account. To gain access to the system, all that is required is a PC with terminal software and a serial console cable to the systems console port. When the system is operating in a live network, it should be protected by at least a local user account. It is not possible to connect to the system from a remote workstation without a configured local user account (or remote authentication).

In a network with more than a few routers, a central user account and authentication server using the RADIUS authentication protocol can simplify the management of the network. All users that have access to network equipment have accounts configured in one location. The network equipment then verifies the user name and password entered by the user when they connect using the console port, or using a Telnet session with the authentication server. Remote authentication makes it easier to add, change, or delete user accounts in the network, rather than configuring local user accounts on each network device.

Managing Local User Accounts

To create or delete local user accounts, you must be in the configuration context (enter iBOS command config terminal). To create, delete or modify local user accounts, use the **username** configuration command.

Example 12 Create User Account

```
ibos(config)# username test password cat13
```

Example 13 Change User Account

```
ibos(config)# username test password oi812
```

Example 14 Delete User Account

```
ibos(config)# no username test
```

The password is stored in the configuration in a ciphered form, so that users cannot see each other's passwords simply by listing the configuration.

Example 15 Display User Account Configuration

```
ibos# show running-config | grep username
username test password Wnr2XvkS/.iQU encrypted
ibos#
```

Remote Login

iBOS supports remote login through the well-known Telnet and SSH services. To login, a user account must be configured on the system. If no user account is configured, iBOS will deny any remote access.

Example 16 Remote Login using Telnet

```
[daveva@kiwi:/export/users/daveva]$ telnet 10.0.0.1
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
Waystream terminal
Localtime is Fri Jul 29 22:07:10 2011


username: test
password:

iBOS 6.7.0-R Copyright (C) 2001-2016 Waystream AB
All rights reserved.
asr>
```

Example 17 Remote Login using SSH

```
[daveva@kiwi:/export/users/daveva]$ ssh 10.0.0.1
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
Waystream terminal
Localtime is Fri Jul 29 22:07:10 2011


username: test
password:

iBOS 6.7.0-R Copyright (C) 2001-2015 Waystream AB
All rights reserved.
asr>
```

The user is given three attempts to enter the correct password before the connection terminates. The maximum number of login attempts, and other parameters, can be configured using the **login** with the **telnetd** or **sshd** configuration commands.

Remote access to the system can also be restricted by access-filters attached to the Telnet service.

Example 18 Restricting Remote Login Network Access

```
ibos(config)# access-list telnet
ibos(config-acl)# seq 10 permit ip host 10.11.12.13
ibos(config-acl)# exit
ibos(config)# telnetd access-group telnet
```

In the example above, Telnet access will be permitted only from a host with the IP address 10.11.12.13. All other connections will be denied. The change will take effect on the next connection attempt to the system.

Support for Telnet Clients

If an ASR becomes unreachable due to incorrect routing, a network operator may wish to log on to an ASR closest to the failing ASR and start a remote session from iBOS. To do this, the **telnet** command has been added.

Basic CLI Commands

The following commands are used for basic CLI functionality:

Table 8 Basic CLI Commands

Command	Description
begin	Search for pattern in output and begin printing when found
clear line	Terminate session
configure terminal	Enter system configuration mode
disable	Go to user mode
do	Run executable command in configuration context
enable	Change to admin mode
enable secret	Configure the enable password
end	Exit configuration mode
exclude	Search for pattern in output and omit matching lines
exit	Logout from system
grep	Search for pattern in output
head	Print a number of lines of output read from the pipe
help	Display CLI help summary
include	Search for pattern in output
license	Enable/disable license
logout	Logout from system
more	Display one screen of output at a time
reload	Reload the operating system/cancel a reload
telnet	Start a telnet session
quit	Logout from iBOS
service terminal prompt format	Configures the format of the command line prompt
show history	Display the content of the command history buffer
show terminal	Display terminal configuration parameters
show users	Display information about users in the system
stop	Search for pattern in output and stop printing when found
taiil	Print the last number of lines read from pipe
telnetd	Configure Telnet session
terminal	Set terminal screen length and width
terminal monitor	Toggle display of logging information to the terminal
terminal reset	Reset all terminal settings
terminal screen	Set terminal screen parameters
terminal timeout	Set terminal idle timeout.

Command	Description
username	Configure a local user
watch	Repeat command after interval seconds until interrupted
wc	Count words in output
who	Display information about users in the system
write	Write/erase running configuration

System Management

System management of Waystream products is performed by defining the configuration parameters in the iBOS operating system.

System management configuration is described in the following:

- "System Commands" on page 32
- "Authentication, Authorization and Accounting (AAA)" on page 34
- "Environment Monitor" on page 55
- "File System" on page 55
- "System Time and Network Time Protocol (NTP)" on page 56
- "Logging" on page 57

System Commands

There are system commands used to configure basic parameters that affect the look and feel of the system. These configurations are described in:

- "Setting the System Hostname" on page 32
- "Setting Domain Name Server" on page 32
- "Setting System Domain Name" on page 33
- "Setting System Configuration Registers" on page 33

Setting the System Hostname

The system hostname is displayed as a prompt in the iBOS command line interface. The hostname has no other purpose than to enable the user to identify which router he is logged into, thereby avoiding confusion or configuration of the wrong unit.

Example 19 Setting System Hostname

```
ibos# conf t
ibos(config)# hostname BB12-GW
BB12-GW(config)# end
BB12-GW#
```

Setting Domain Name Server

In order to perform DNS-resolutions and enable the use of hostnames in commands such as **ping** and **traceroute**, iBOS needs to know the IP address of the DNS server(s) to be used. This is configured with the **ip name-server** command.

Example 20 Setting DNS Server

```
ibos# conf t
ibos(config)# ip name-server 10.130.5.14
ibos(config)# ip name-server 10.130.8.14
```

Configuring multiple addresses for the DNS will make iBOS try each one in order until an authoritative response is received.

The name-server configuration is local for iBOS only. It does not affect DNS server information in relayed DHCP packets or other traffic.

The **host** command can be used to test DNS resolution.

Setting System Domain Name

On the Internet, a host typically belongs to a domain. The iBOS system can be configured with a domain name. The primary purpose of the domain name is to simplify DNS resolution.

When performing **ping**, **traceroute** or other activities where a remote host is specified, if no other valid domain name has been provided a configured domain name is automatically added to the specified hostname in these commands

Example 21 Setting Domain Name

```
ibos(config)# ip domain-name waystream.com
ibos(config)# end
ibos# ping kista
PING kista.waystream.com (172.16.9.247): 36 data bytes (IP packet
64 bytes)
!!!!!
----kista.waystream.com PING Statistics----
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max = 1.556/1.591/1.648 ms
```

Setting System Configuration Registers

iBOS has a set of configuration registers that control basic operations of system function that also affect the system Bootloader operation. Configuration registers are shared between iBOS and the system Bootloader. When the system starts and is interrupted into the Bootloader it is possible to set configuration registers to a temporary value that is used for the next startup of iBOS. In iBOS, the permanent setting of configuration register occurs in configuration mode.

Example 22 iBOS Configuration Registers

```
Config register     Running value      Stored value
-------------------------------------------------------------
Baud rate                  115200           115200
Ignore config                  no               no
Net boot                      yes              yes
Save core                     yes              yes
Self test                    none             none
Watchdog T1                    25               25
Watchdog T2                    10               10
```

For more information, see the **config-register** command.

Authentication, Authorization and Accounting (AAA)

Authentication, Authorisation and Accounting (AAA) is an established term for functionality that authenticates and authorises user activities, and that keeps accounting records of these activities.

The AAA module in iBOS operates as a middleware between service (features that allow an action) and a verification of that action.

Function	Description
Authentication	Authentication verifies that the user, object or system attempting to perform an action really is the user, object or system that it claims to be. If authentication is successful, access to the network is granted. If authentication fails, access is denied.
Authorisation	Authorisation verifies that the authenticated user, object or system is allowed to perform specific actions, to initiate specific processes, or to use other system functionality. The authorisation enforces policies for each individual user, object or system. Usually, authorisation is performed within the context of authentication where the authentication credentials can also authorize different types of access and/or activities within the network.
Accounting	Accounting records the activity of a user, object or system. Accounting also keeps track of AAA requests and sending logged information that can be used for billing, security, or other purposes.

In addition to enabling local user authentication, (refer "User Accounts and Remote Access" on page 26, a real-life network will use a remote AAA server. For AAA, iBOS supports the Remote Authentication Dial-In User Service (RADIUS) protocol.

Configuring AAA

By default, AAA is disabled. To enable the AAA connection to a RADIUS server, use the **aaa authentication radius priority** command:

Example 23 Configure AAA for RADIUS Server

```
ibos(config)# aaa authentication radius priority 8
ibos(config)# aaa request-timeout 15
ibos(config)# radius server 10.10.2.4 secret wAysTrEam
ibos(config)# ip name-server 10.10.2.1
ibos(config)# radius server radius.waystream.com retry 5 timeout 2
```

The RADIUS authentication must have a higher priority (i.e. lower value) than the local authentication to be executed first. The RADIUS server must reply within 15 seconds, or local authentication will be performed.

To verify the configuration, use the **show aaa** command:

```
ibos> show aaa
AAA configuration

 Request-timeout: 30
  Authentication: radius, priority: 8
  Authentication: local, priority: 10  (statically set)
         Service: login
```

Note: *When the RADIUS priority is equal or lower (i.e. higher value), iBOS will only perform local authentication.*

Available iBOS commands to configure AAA include:

Command	Description
aaa	Configure AAA settings
radius	Configure RADIUS server
port authentication 802.1X	Enable 802.1X authentication on an interface
ip static script	Enable script for service deployment by RADIUS server on static IPv4 client interfaces

DHCP RADIUS Authentication

There are three components for the DHCP-Radius authentication solution:

- DHCP Server
- Radius Server
- DHCP Relay Agent

DHCP Server

The Dynamic Host Configuration Protocol (DHCP) provides configuration parameters to Internet hosts.

DHCP consists of two components:

- Protocol for delivering host-specific configuration parameters from a DHCP server to a host.
- Mechanism for allocation of network addresses to hosts.

DHCP is built on a client-server model, where designated DHCP servers allocate network addresses and deliver configuration parameters to dynamically configured clients.

Radius Server

The Radius server authenticates client hosts via RADIUS protocol and provides service authorization, accounting, and dynamically updates a service for client host using CoA request messages. The Radius server can also dynamically disconnect clients using CoA disconnect requests.

A Captive portal (Login host) server provides a web page so that the client can input username and password for authenticating access rights via the Radius server.

DHCP Relay Agent

In an iBOS IPv4 network, a client host configuration that includes assignment of IP addresses normally uses DHCP (RFC 2131). If a server and a client are not on the same physical subnet, the DHCP messages need to be relayed.

iBOS can act as a DHCP relay agent. In such a scenario, clients get the IP address from an external DHCP server via the iBOS device, and then iBOS automatically enforces the service for the client on the interface (using a group of policy/policy-map).

DHCP RADIUS authentication is described in the following:

- "Configuring DHCP-RADIUS with BNG" on page 36
- "Configuring DHCP-RADIUS with BNG and DHCP Snooper" on page 37

Configuring DHCP-RADIUS with BNG

The simple DHCP-RADIUS use case includes a relay server (BNG), DHCP Server and Radius server to provide the service (policy/policy-map) to the client using a 2-stage RADIUS AAA (Authentication, Authorization, Accounting) method. The two authentication stages are:

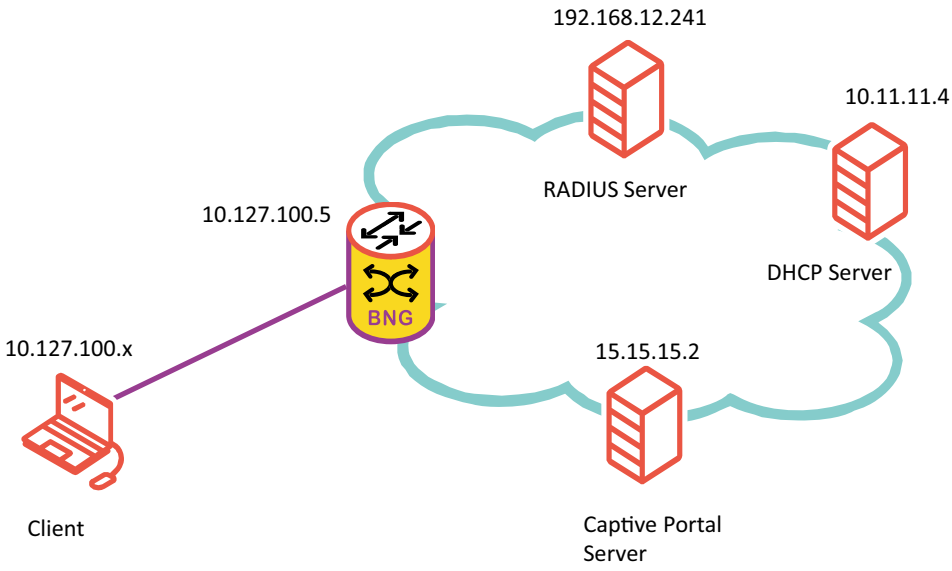
- Get the service CAPTIVE-PORTAL
- Get the service CAPTIVE-PORTAL

The message flow for this single level authentication is:

1. The Client sends an DHCP-DISCOVER to the BNG server which triggers iBOS to send an access request to the Radius server
2. The Radius server replies with a Access Accept with respective service definition e.g. CAPTIVE-PORTAL
3. BNG records the Access-Accept packet to process after DHCP_ACK is received from the server.
4. Then BNG server forwards the DHCP-DISCOVER towards DHCP-SERVER, and subsequently forwards the reply (DHCP_OFFER) from the DHCP-SERVER towards the client
5. After the client receives DHCP_OFFER from server, it sends out an DHCP-REQUEST to BNG server.
6. After getting DHCP_ACK from DHCP-SERVER:
   - BNG applies the authentication parameter received earlier (see step 3)
   - BNG starts accounting and also starts the Interim Interval timer for periodic accounting updates.

In following topology, the BNG is acting both as a DHCP-RELAY agent (relays DHCP messages between the Client and DHCP server) and as a RADIUS client (authenticates the Client host towards Radius Server).

Figure 1 DHCP-Radius BNG Topology



To configure the BNG server towards the DHCP-Server and Radius server, do the following:

1. Configure DHCP on BNG:

```
iBOS(config)# interface gigabitethernet7
   parameter dhcp service service-definition gen-static-sd service-
selection gen-dhcp-yy
    ip dhcp relay script
```

2. Configure Radius on BNG:

```
iBOS(config)# service-selection gen-dhcp-yy
iBOS(config)# parameter dhcp relay-to dhcp-server 10.11.11.4
source-interface gi7/vid100
iBOS(config)# parameter dhcp match relay
iBOS(config)# parameter dhcp radius source-interface

gigabitethernet1 server-ip 192.168.12.241 server-port 1812 secret
waystream
```

Configuring DHCP-RADIUS with BNG and DHCP Snooper

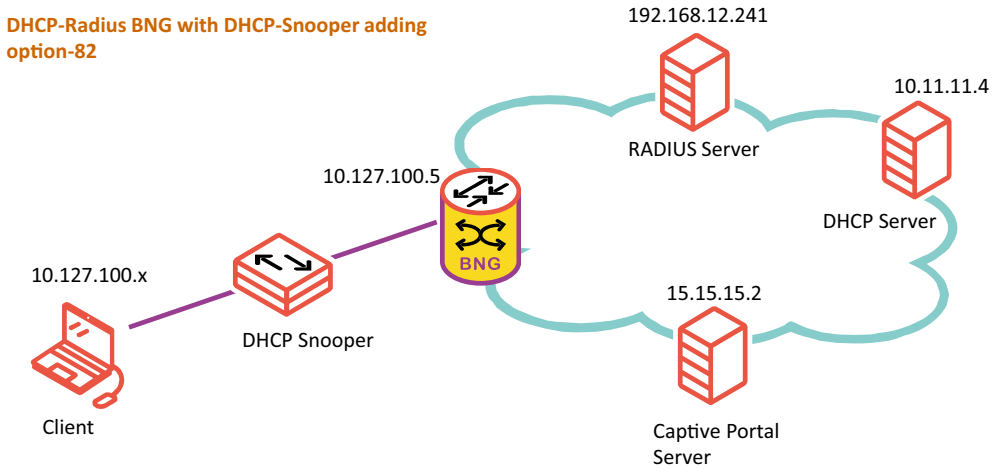
In the following example, the BNG and DHCP-RADIUS implementation is extended with the DHCP snooper in the network access switch that includes option-82 in the client information which is an improved security feature offered by iBOS.

The Authentication process is as follows:

1. The Client sends an DHCP-DISCOVER to the BNG server which triggers iBOS to send an access request to the Radius server
2. The Radius server replies with a Access Accept with respective service definition e.g. INTERNET
3. BNG records the Access-Accept packet to process after DHCP_ACK is received from the server.
4. Then BNG server forward the DHCP-DISCOVER towards DHCP-SEREVER and again forward reply (DHCP_OFFER) from DHCP-SERVER towards client

5. After the client receives DHCP_OFFER from server, it sends out an DHCP-REQUEST to BNG server.
6. After getting DHCP_ACK from DHCP-SERVER:
   - BNG apply the authentication parameter received earlier (see step 3)
   - BNG starts accounting and also starts the Interim Interval timer for periodic accounting updates
7. After the DHCP_CLIENT gets the IP, the traffic will be redirected to the Captive Portal server where the user enters a new username and password for the second level of authentication. The BNG then sends this information to Radius server.
8. The Radius server will send CoA request to IBOS to inform change in authentication parameters. iBOS stops existing accounting and Interim accounting.
9. Upon receiving CoA request IBOS will send second authentication request to RADIUS_SERVER with new username and password.
10. Then the RADIUS server will respond with Access Accept with new services (INTERNET, HTTP). iBOS sends request to start new accounting.

Figure 2 DHCP-Radius with Snooper Topology



To configure the BNG server towards the DHCP-Server and Radius server, do the following:

1. Configure DHCP relay script and service selection on BNG:

```
iBOS(config)# interface gi7/vid100
iBOS(config-if)# ip dhcp relay script
iBOS(config-if)# parameter dhcp service service-definition gen-
static-sd service-selection gen-dhcp-yy
iBOS(config-if)# no shutdown
```

2. Configure Radius and DHCP servers on BNG to relay:

```
iBOS(config)# service-selection gen-dhcp-yy
iBOS(config)# parameter dhcp relay-to dhcp-server 10.11.11.4
source-interface gi7/vid100
iBOS(config)# parameter dhcp match snpa any relay
iBOS(config)# parameter dhcp radius source-interface
gigabitethernet1 server-ip 192.168.12.241 server-port 1812 secret
```

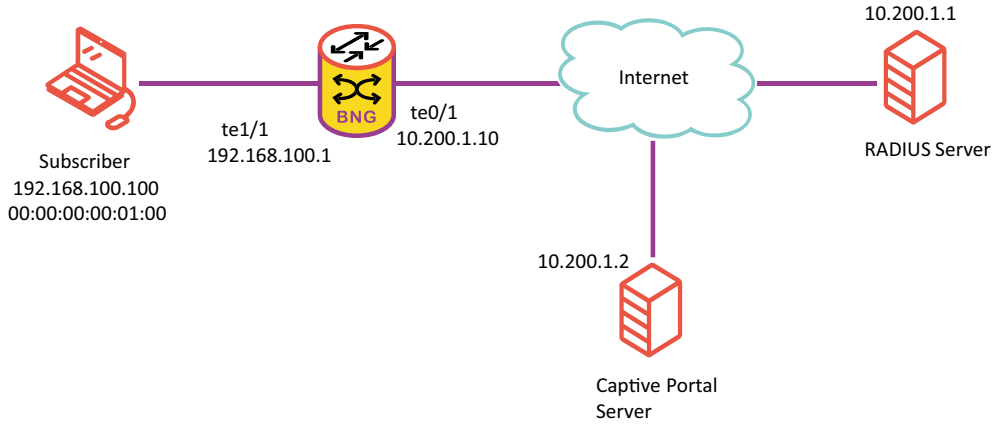
Policy-based RADIUS

The policy-based (aka IP packet-trigger) RADIUS feature enables the authentication of subscribers and application of dynamic policies that are controlled by the RADIUS server.

Subscriber instantiation is triggered for ingress packets on interface tengigabitethernet1/1 on the BNG.

In the following network example,

Figure 3 Policy-based RADIUS Network Example



The procedure for policy-based Radius authentication is:

1. IP packet is sent from Subscriber to BNG
2. RADIUS Access-Request(1) is sent from BNG to Radius server
3. RADIUS Access-Accept is sent from radius server to BNG with:
   - Captive portal service for Captive Portal users
   - Internet services for other users (skip steps 4 through 7)
4. Captive Portal users are redirected to Captive Portal server and perform login on the server login webpage.
5. Authentication is sent from Captive Portal to RADIUS Server
6. Radius CoA-Request with username/password is sent from RADIUS server to BNG
7. RADIUS Access-Request(2) is sent from BNG to RADIUS server
8. RADIUS Access-Accept is sent from RADIUS server to BNG with Internet services (GOOGLE, AKAMAI, INTERNET, etc).
9. BNG will apply those Internet services from Radius server on the Subscriber and Radius COA-ACK is sent from BNG to radius server.

Note: *BNG, RADIUS server, and the optional Captive-Portal Server do not need to belong to the same subnet. All devices must be L3 reachable via routing.*

To setup policy-based RADIUS authentication:

1. Configure the default ACL on BNG server. First, add ACL to allow DNS packet, so the client can resolve the Captive-Portal server address:

```
BNG# configure terminal
BNG(config)# access-list DNS-ACL
BNG(config-acl)# permit tcp any 53
BNG(config-acl)# permit udp any 53
```

2. Add HTTP stream filter, so client can access the Captive-Portal login webpage:

```
BNG# configure terminal
BNG(config)# access-list CAPTIVE-PORTAL-ACL
BNG(config-acl)# permit tcp any 80
```

3. # Add HTTP stream filter for HTTP-CACHE service:

```
BNG# configure terminal
BNG(config)# access-list HTTP-CACHE-ACL
BNG(config-acl)# permit tcp any 80
BNG(config-acl)# permit tcp any 8080
```

4. Add torrent stream filter for TORRENT service:

```
BNG# configure terminal
BNG(config)# access-list TORRENT-ACL
BNG(config-acl)# permit tcp any range 1024 65535
(config-acl)# permit udp any range 54 65535
```

5. Add service-selection for RADIUS, and specify RADIUS configure:

```
BNG# configure terminal
BNG(config)# service-selection bng-ss
BNG(config-service-selection)# parameter bng match ip any radius
BNG(config-service-selection)# parameter bng aaa-source-interface
loopback0
BNG(config-service-selection)# parameter bng auth server-ip
10.200.1.1 server-port 1812 secret waystream
BNG(config-service-selection)# parameter bng acct server-ip
10.200.1.1 server-port 1813 secret waystream
BNG(config-service-selection)# parameter bng coa secret testing1234
```

6. Enable policy-based RADIUS on the client interface:

```
BNG# configure terminal
BNG(config)# interface tengigabitethernet1/1
BNG(config-if-te)# ip address 192.168.100.1 255.255.255.0
BNG(config-if-te)# ip strict-clients
BNG(config-if-te)# policy radius to-script shortcut
BNG(config-if-te)# policy-map 192.168.100.0/24 policy radius
BNG(config-if-te)# parameter bng service service-selection bng-ss
BNG(config-if-te)# no shutdown
```

To verify the configuration:

```
BNG# show running-config
! version ibos-mpc480-7.0.1-ED-RC24 (1706051919) (ibos/ibos-mpc480-
7.0.1-ED-RC24.bz2)
interface ethernet0
 ip address 10.192.0.183 255.255.255.0
 no shutdown

interface loopback0
 ip address 10.0.0.21 255.255.255.255
 no shutdown

interface tengigabitethernet0/1
```

```
 port uplink
 ip address 10.200.1.10 255.255.255.0
 no shutdown

interface tengigabitethernet0/2
 port uplink

interface tengigabitethernet0/3
interface tengigabitethernet0/4

interface tengigabitethernet1/1
 ip strict-clients
 policy radius to-script shortcut
 policy-map 192.168.100.0/24 policy radius
 ip address 192.168.100.1 255.255.255.0
 parameter bng service service-selection bng-ss
 no shutdown

interface tengigabitethernet1/2

interface tengigabitethernet1/3

interface tengigabitethernet1/4

interface downstream0
 no shutdown

hostname asr1

exception dump 172.19.36.4 10001

logging console debugging
logging monitor debugging

ip route 0.0.0.0/0 10.192.0.1

ntp server 172.19.36.4
ntp interval 60

clock timezone Asia/Shanghai

username t password sFMEJcTWQwZOc encrypted

access-list DNS-ACL
 seq 10 permit tcp any 53
 seq 20 permit udp any 53

access-list CAPTIVE-PORTAL-ACL
 seq 10 permit tcp any 80

access-list HTTP-CACHE-ACL
 seq 10 permit tcp any 80

access-list TORRENT-ACL
 seq 10 permit tcp any range 1024 65535
 seq 20 permit udp any range 54 65535

service-selection bng-ss
 parameter bng match ip any radius
```

```
 parameter bng source-interface loopback0
 parameter bng auth server-ip 10.200.1.1 server-port 1812 secret
waystream
 parameter bng acct server-ip 10.200.1.1 server-port 1813 secret
waystream
 parameter bng coa secret testing1234 disconn-delay 10
!end
```

To show client status after 1st-stage authentication (got Captive-Portal service):

```
BNG# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
-----------------------------------------------------------------
---------------

192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted CAPTIVE-PORTAL

asr# script run bng show client detail
           Interface: tengigabitethernet1/1
       Physical port: tengigabitethernet1/1
          IP Address: 192.168.100.100
                SNPA: 00:00:00:00:01:00
             On-line: No
              Uptime: 40s
          Expires in: 0h8m31s
               lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:29711621
        Authenticate: Access-Accepted
     Session-timeout: 511 of 600 seconds
      Captive-Portal: 10.200.1.2
             Service: CAPTIVE-PORTAL
          Input rate: 6000 kbit/s
         Output rate: 11000 kbit/s
```

To show Captive-portal service (policy/policy-map) assigned for online clients:

```
BNG# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
ingress shape 6000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
egress shape 11000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 accounting
CAPTIVE-PORTAL-192-168-100-100
   !volatile policy CAPTIVE-PORTAL-192-168-100-100 ip destination-
nat host 10.200.1.2
    policy radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy DNS
access-group DNS-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy CAPTIVE-
PORTAL-192-168-100-100 access-group CAPTIVE-PORTAL-ACL
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-selection bng-ss
    no shutdown
```

To show client status after 2nd-stage authentication (actual service assigned):

```
BNG# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
-------------------------------------------------------------------
---------------
192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted2 INTERNET, HTTP-CACHE, TORRENT

asr# script run bng show client detail
           Interface: tengigabitethernet1/1
      Physical port: tengigabitethernet1/1
         IP Address: 192.168.100.100
               SNPA: 00:00:00:00:01:00
            On-line: Yes
             Uptime: 67s
         Expires in: 0h9m34s
              lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:14129552
        Authenticate: Access-Accepted2
     Session-timeout: 600 seconds
      Captive-Portal: 10.200.1.2
             Service: INTERNET
          Input rate: 5000 kbit/s
         Output rate: 10000 kbit/s
             Service: HTTP-CACHE
          Input rate: 7000 kbit/s
         Output rate: 12000 kbit/s
             Service: TORRENT
          Input rate: 6000 kbit/s
         Output rate: 11000 kbit/s
```

To show actual services (policy/policy-map) assigned for online clients:

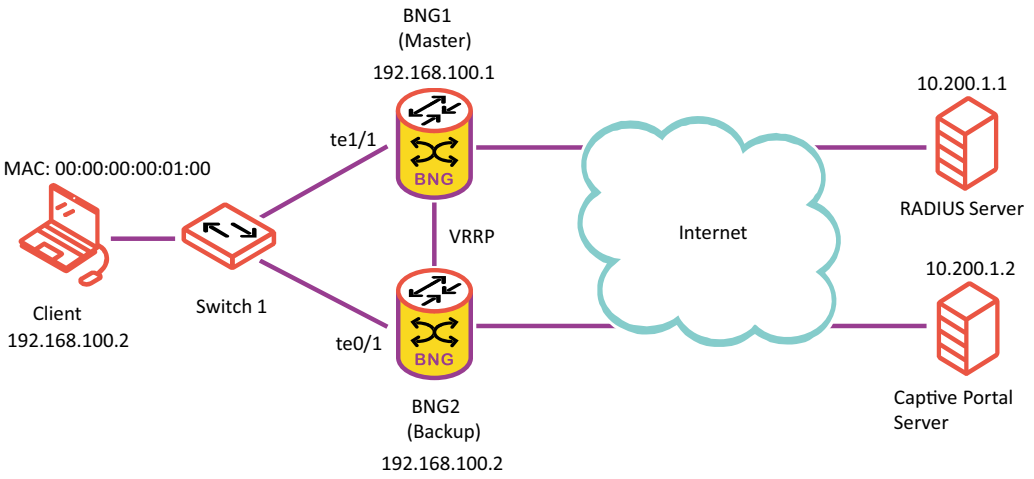
```
BNG# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy TORRENT-192-168-100-100 conditioner ingress
shape 6000 green pass
    !volatile policy TORRENT-192-168-100-100 conditioner egress
shape 11000 green pass
   !volatile policy TORRENT-192-168-100-100 accounting TORRENT-192-
168-100-100
    !volatile policy INTERNET-192-168-100-100 conditioner ingress
shape 5000 green pass
    !volatile policy INTERNET-192-168-100-100 conditioner egress
shape 10000 green pass
    !volatile policy INTERNET-192-168-100-100 accounting INTERNET-
192-168-100-100
   !volatile policy HTTP-CACHE-192-168-100-100 conditioner ingress
shape 7000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 conditioner egress
shape 12000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 accounting HTTP-
CACHE-192-168-100-100
    policy radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy HTTP-
CACHE-192-168-100-100 access-group HTTP-CACHE-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy TORRENT-
192-168-100-100 access-group TORRENT-ACL
```

```
    !volatile policy-map 192.168.100.100/32 seq 30 policy INTERNET-
192-168-100-100
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-definition bng-sd service-
selection bng-ss
    no shutdown
```

RADIUS with Redundancy

In the following example, two BNG servers have VRRP running to provide redundancy for the client host. In the event that BNG1 goes down, VRRP will make the Backup server BNG2 take over as the new VRRP Master.

Figure 4 Radius Redundancy Network Example



Note: *BNG, RADIUS server, and Captive-Portal Server do not need to be in the same subnet. The only requirement is that all devices are L3 reachable via routing.*

To setup policy-based RADIUS with redundancy:

1. Configure default ACLs on both BNGs (BNG1/BNG2):

a. Add ACL to bypass DNS packet, so client can resolve the Captive-Portal server address:

```
BNG1# configure terminal
BNG1(config)# access-list DNS-ACL
BNG1(config-acl)# permit tcp any 53
(config-acl)# permit udp any 53
```

b. Add HTTP stream filter, so client can access the Captive-Portal login webpage:

```
BNG1# configure terminal
BNG1(config)# access-list CAPTIVE-PORTAL-ACL
```

c. Add HTTP stream filter for HTTP-CACHE service:

```
BNG1# configure terminal
BNG1(config)# access-list HTTP-CACHE-ACL
BNG1(config-acl)# permit tcp any 80
BNG1(config-acl)# permit tcp any 8080
```

d. Add torrent stream filter for TORRENT service:

```
BNG1# configure terminal
BNG1(config)# access-list TORRENT-ACL
BNG1(config-acl)# permit tcp any range 1024 65535
BNG1(config-acl)# permit udp any range 54 65535
```

e. Add service-selection for RADIUS, and specify RADIUS configure:

```
BNG1# configure terminal
BNG1(config)# service-selection bng-ss
BNG1(config-service-selection)# parameter bng match ip any radius
BNG1(config-service-selection)# parameter bng source-interface
loopback0
BNG1(config-service-selection)# parameter bng auth server-ip
10.200.1.1 server-port 1812 secret waystream
BNG1(config-service-selection)# parameter bng acct server-ip
10.200.1.1 server-port 1813 secret waystream
BNG1(config-service-selection)# parameter bng coa secret
testing1234
```

f. Repeat steps a-e for BNG2

2. Enable VRRP Master on BNG1:

```
BNG1# configure terminal
BNG1(config)# interface tengigabitethernet1/1
BNG1(config-if-te)# ip address 192.168.100.1 255.255.255.0
BNG1(config-if-te)# ip vrrp 100 address 192.168.100.1
BNG1(config-if-te)# ip vrrp 100 priority 254
BNG1(config-if-te)# ip vrrp 100 prohibit-preemption
BNG1(config-if-te)# ip vrrp 100 accept
BNG1(config-if-te)# no shutdown
```

3. Enable VRRP Backup on BNG2:

```
BNG2# configure terminal
BNG2(config)# interface tengigabitethernet1/1
BNG2(config-if-te)# ip address 192.168.100.2 255.255.255.0
BNG2(config-if-te)# ip vrrp 100 address 192.168.100.1
BNG2(config-if-te)# ip vrrp 100 priority 253
BNG2(config-if-te)# ip vrrp 100 prohibit-preemption
BNG2(config-if-te)# ip vrrp 100 accept
BNG2(config-if-te)# no shutdown
```

4. Enable policy-based RADIUS on BNG1 and BNG2:

```
BNG1# configure terminal
BNG1(config)# interface tengigabitethernet1/1
BNG1(config-if-te)# ip strict-clients
BNG1(config-if-te)# policy radius to-script shortcut
BNG1(config-if-te)# policy-map 192.168.100.0/24 policy radius
BNG1(config-if-te)# parameter bng service service-selection bng-ss
BNG1(config-if-te)# no shutdown
BNG2# configure terminal
BNG2(config)# interface tengigabitethernet1/1
BNG2(config-if-te)# ip strict-clients
BNG2(config-if-te)# policy radius to-script shortcut
BNG2(config-if-te)# policy-map 192.168.100.0/24 policy radius
```

```
BNG2(config-if-te)# parameter bng service service-selection bng-ss
BNG2(config-if-te)# no shutdown
```

To see the detailed configuration for each BNG, use the command **show running-config**:

```
BNG1# show running-config
! version ibos-mpc480-7.0.1-ED-RC26 (1706091525) (ibos/ibos-mpc480-
7.0.1-ED-RC26.bz2)
interface ethernet0
 ip address 10.192.0.183 255.255.255.0
 no shutdown

interface loopback0
 ip address 10.0.0.21 255.255.255.255
 no shutdown

interface tengigabitethernet0/1
 port uplink
 ip address 10.200.1.10 255.255.255.0
 no shutdown

interface tengigabitethernet0/2
 port uplink

interface tengigabitethernet0/3

 tengigabitethernet0/4

interface tengigabitethernet1/1
 ip strict-clients
 policy radius to-script shortcut
 policy-map 192.168.100.0/24 policy radius
 ip address 192.168.100.1 255.255.255.0
 parameter bng service service-selection bng-ss
 ip vrrp 100 address 192.168.100.1
 ip vrrp 100 priority 254
 ip vrrp 100 prohibit-preemption
 ip vrrp 100 accept
 no shutdown

interface tengigabitethernet1/2

interface tengigabitethernet1/3

interface tengigabitethernet1/4

interface downstream0
 no shutdown

hostname BNG1

exception dump 172.19.36.4 10001

logging console debugging
logging monitor debugging

ip route 0.0.0.0/0 10.192.0.1

ntp server 172.19.36.4
ntp interval 60
```

```
clock timezone Asia/Shanghai

username t password sFMEJcTWQwZOc encrypted

access-list DNS-ACL
 seq 10 permit tcp any 53
 seq 20 permit udp any 53

access-list CAPTIVE-PORTAL-ACL
 seq 10 permit tcp any 80

access-list HTTP-CACHE-ACL
 seq 10 permit tcp any 80

access-list TORRENT-ACL
 seq 10 permit tcp any range 1024 65535
 seq 20 permit udp any range 54 65535

service-selection bng-ss
 parameter bng match ip any radius
 parameter bng aaa-source-interface loopback0
 parameter bng auth server-ip 10.200.1.1 server-port 1812 secret
waystream
 parameter bng acct server-ip 10.200.1.1 server-port 1813 secret
waystream
 parameter bng coa secret testing1234 disconn-delay 10
```

```
BNG2# sh run
 ibos-mpc480-7.0.1-ED-RC26 (1706091525) (ibos/ibos-mpc480-7.0.1-
ED-RC26.bz2)
interface ethernet0
 address 10.192.0.184 255.255.255.0
 shutdown

interface loopback0
 ip address 10.0.0.22 255.255.255.255
 shutdown

interface tengigabitethernet0/1
 port uplink
 ip address 10.200.2.10 255.255.255.0
 no shutdown

interface tengigabitethernet0/2
 port uplink

interface tengigabitethernet0/3

interface tengigabitethernet0/4

interface tengigabitethernet1/1
 ip strict-clients
 policy radius to-script shortcut
 policy-map 192.168.100.0/24 policy radius
 ip address 192.168.100.2 255.255.255.0
 parameter bng service service-selection bng-ss
 ip vrrp 100 address 192.168.100.1
```

```
 ip vrrp 100 priority 253
 ip vrrp 100 prohibit-preemption
 ip vrrp 100 accept
 no shutdown

interface tengigabitethernet1/2

interface tengigabitethernet1/3

interface tengigabitethernet1/4

interface downstream0
 no shutdown

hostname BNG2

exception dump 172.19.36.4 10001

logging console debugging
logging monitor debugging

ip route 0.0.0.0/0 10.192.0.1

ntp server 172.19.36.4
ntp interval 60

clock timezone Asia/Shanghai

username t password sFMEJcTWQwZOc encrypted

access-list DNS-ACL
 seq 10 permit tcp any 53
 seq 20 permit udp any 53

access-list CAPTIVE-PORTAL-ACL
 seq 10 permit tcp any 80

access-list HTTP-CACHE-ACL
 seq 10 permit tcp any 80

access-list TORRENT-ACL
 seq 10 permit tcp any range 1024 65535
 seq 20 permit udp any range 54 65535

service-selection bng-ss
 parameter bng match ip any radius
 bng source-interface loopback0
 parameter bng auth server-ip 10.200.1.1 server-port 1812 secret
waystream
 parameter bng acct server-ip 10.200.1.1 server-port 1813 secret
waystream
 parameter bng coa secret testing1234 disconn-delay 10
!end
```

To verify the configuration, there are a number of show commands.

To show VRRP state for vrid:

```
BNG1# show ip vrrp
Interface             Grp  State   Pri   IP Addresses    Adv  MDwn
-------------------------------------------------------------------
-------
tengigabitethernet1/1    100  Master  254   A192.168.100.1     1s
0.0s

BNG2# show ip vrrp
Interface             Grp  State   Pri   IP Addresses    Adv  MDwn
-------------------------------------------------------------------
-------
tengigabitethernet1/1    100  Backup  253   A192.168.100.1      -
2.6s
```

To show the client status after 1st-stage authentication (got Captive-Portal service on BNG1):

```
BNG1# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
-------------------------------------------------------------------
-------------

192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted  CAPTIVE-
 PORTAL

BNG1# script run bng show client detail
           Interface: tengigabitethernet1/1
       Physical port: tengigabitethernet1/1
          IP Address: 192.168.100.100
                SNPA: 00:00:00:00:01:00
VRID: 100
             On-line: No
              Uptime: 40s
          Expires in: 0h8m31s
               lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:29711621
        Authenticate: Access-Accepted
     Session-timeout: 511 of 600 seconds
      Captive-Portal: 10.200.1.2
             Service: CAPTIVE-PORTAL
          Input rate: 6000 kbit/s
         Output rate: 11000 kbit/s

BNG2# script run bng show client
BNG2#
```

To show the Captive-portal service (policy/policy-map) is assigned for online clients:

```
BNG1# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
ingress shape 6000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
egress shape 11000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 accounting
```

```
CAPTIVE-PORTAL-192-168-100-100
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 ip destination-
nat host 10.200.1.2
    policy radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy DNS
access-group DNS-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy CAPTIVE-
PORTAL-192-168-100-100 access-group CAPTIVE-PORTAL-ACL
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-definition bng-sd service-
selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 254
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown

BNG2# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    policy radius to-script shortcut
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.2 255.255.255.0
    parameter bng service service-selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 253
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
```

To show client status after 2nd-stage authentication (actual service assigned on BNG1):

```
BNG1# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
------------------------------------------------------------------
--------------
192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted2
 INTERNET, HTTP-CACHE, TORRENT

BNG1# script run bng show client detail
           Interface: tengigabitethernet1/1
       Physical port: tengigabitethernet1/1
          IP Address: 192.168.100.100
                SNPA: 00:00:00:00:01:00
VRID: 100
             On-line: Yes
              Uptime: 67s
          Expires in: 0h9m34s
               lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:14129552
        Authenticate: Access-Accepted2
     Session-timeout: 600 seconds
      Captive-Portal: 10.200.1.2
```

```
            Service: INTERNET
         Input rate: 5000 kbit/s
        Output rate: 10000 kbit/s
            Service: HTTP-CACHE
         Input rate: 7000 kbit/s
        Output rate: 12000 kbit/s
            Service: TORRENT
         Input rate: 6000 kbit/s
        Output rate: 11000 kbit/s
BNG2# script run bng show client
```

To show actual service (policy/policy-map) assigned for online clients:

```
BNG1# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy TORRENT-192-168-100-100 conditioner ingress
shape 6000 green pass
    !volatile policy TORRENT-192-168-100-100 conditioner egress
shape 11000 green pass
   !volatile policy TORRENT-192-168-100-100 accounting TORRENT-192-
168-100-100
    !volatile policy INTERNET-192-168-100-100 conditioner ingress
shape 5000 green pass
    !volatile policy INTERNET-192-168-100-100 conditioner egress
shape 10000 green pass
    !volatile policy INTERNET-192-168-100-100 accounting INTERNET-
192-168-100-100
   !volatile policy HTTP-CACHE-192-168-100-100 conditioner ingress
shape 7000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 conditioner egress
shape 12000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 accounting HTTP-
CACHE-192-168-100-100
    policy radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy HTTP-
CACHE-192-168-100-100 access-group HTTP-CACHE-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy TORRENT-
192-168-100-100 access-group TORRENT-ACL
    !volatile policy-map 192.168.100.100/32 seq 30 policy INTERNET-
192-168-100-100
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-definition bng-sd service-
selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 254
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
BNG2# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    policy radius to-script shortcut
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.2 255.255.255.0
    parameter bng service service-selection bng-ss
    ip vrrp 100 address 192.168.100.1
```

```
    ip vrrp 100 priority 253
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
```

To show redundancy after te1/1 link is down on BNG1

(BNG1 goes to Backup for vrid 100, BNG2 becomes Master for vrid 100):

```
# show VRRP state for vrid:
BNG1# show ip vrrp
Interface             Grp  State   Pri    IP Addresses    Adv  MDwn
-------------------------------------------------------------------
-------
tengigabitethernet1/1   100  INIT   254   A192.168.100.1     -  1.3s

BNG2# show ip vrrp
Interface             Grp  State   Pri    IP Addresses    Adv  MDwn
-------------------------------------------------------------------
-------
tengigabitethernet1/1    100  Master  253   A192.168.100.1     2s
0.0s
```

To show client status after 1st-stage authentication (i.e. Captive-Portal service on BNG2):

```
BNG1# script run bng show client
BNG1#

BNG2# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
-------------------------------------------------------------------
-------------------
192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted  CAPTIVE-
 PORTAL

BNG2# script run bng show client detail
          Interface: tengigabitethernet1/1
       Physical port: tengigabitethernet1/1
          IP Address: 192.168.100.100
                SNPA: 00:00:00:00:01:00
VRID: 100
             On-line: No
              Uptime: 40s
          Expires in: 0h8m31s
               lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:29711621
        Authenticate: Access-Accepted
     Session-timeout: 511 of 600 seconds
      Captive-Portal: 10.200.1.2
             Service: CAPTIVE-PORTAL
          Input rate: 6000 kbit/s
         Output rate: 11000 kbit/s
```

To show Captive-portal service (policy/policy-map) is assigned for online clients:

```
BNG1# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    policy radius to-script shortcut
```

```
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    parameter bng service service-selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 254
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown

BNG2# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
ingress shape 6000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 conditioner
egress shape 11000 green pass
    !volatile policy CAPTIVE-PORTAL-192-168-100-100 accounting
CAPTIVE-PORTAL-192-168-100-100
   !volatile policy CAPTIVE-PORTAL-192-168-100-100 ip destination-
nat host 10.200.1.2
 radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy DNS
access-group DNS-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy CAPTIVE-
PORTAL-192-168-100-100 access-group CAPTIVE-PORTAL-ACL
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.2 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-definition bng-sd service-
selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 253
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
```

To show client status after 2nd-stage authentication (actual service assigned on BNG2):

```
BNG1# script run bng show client
BNG1#

BNG2# script run bng show client
Address       SNPA           Lease Type   Interface  Radius
Service
-------------------------------------------------------------------
--------------
192.168.100.100 00:00:00:00:01:00 600   static te1/1      Access-
Accepted2
 INTERNET, HTTP-CACHE, TORRENT

BNG2# script run bng show client detail
           Interface: tengigabitethernet1/1
       Physical port: tengigabitethernet1/1
          IP Address: 192.168.100.100
                SNPA: 00:00:00:00:01:00
VRID: 100
             On-line: Yes
```

```
             Uptime: 67s
          Expires in: 0h9m34s
               lease: 600
====================Radius Status====================
     Acct-Session-Id: tengigabitethernet1/1:14129552
        Authenticate: Access-Accepted2
     Session-timeout: 600 seconds
      Captive-Portal: 10.200.1.2
             Service: INTERNET
          Input rate: 5000 kbit/s
         Output rate: 10000 kbit/s
             Service: HTTP-CACHE
          Input rate: 7000 kbit/s
         Output rate: 12000 kbit/s
             Service: TORRENT
          Input rate: 6000 kbit/s
         Output rate: 11000 kbit/s
```

To show actual service (policy/policy-map) is assigned for online clients:

```
BNG1# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    policy radius to-script shortcut
    policy-map 192.168.100.0/24 policy radius
    ip strict-clients
    ip address 192.168.100.1 255.255.255.0
    parameter bng service service-selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 254
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
BNG2# show running-config context interface tengigabitethernet1/1
interface tengigabitethernet1/1
    !volatile policy TORRENT-192-168-100-100 conditioner ingress
shape 6000 green pass
    !volatile policy TORRENT-192-168-100-100 conditioner egress
shape 11000 green pass
   !volatile policy TORRENT-192-168-100-100 accounting TORRENT-192-
168-100-100
    !volatile policy INTERNET-192-168-100-100 conditioner ingress
shape 5000 green pass
    !volatile policy INTERNET-192-168-100-100 conditioner egress
shape 10000 green pass
    !volatile policy INTERNET-192-168-100-100 accounting INTERNET-
192-168-100-100
   !volatile policy HTTP-CACHE-192-168-100-100 conditioner ingress
shape 7000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 conditioner egress
shape 12000 green pass
    !volatile policy HTTP-CACHE-192-168-100-100 accounting HTTP-
CACHE-192-168-100-100
    policy radius to-script shortcut
    !volatile policy-map 192.168.100.100/32 seq 10 policy HTTP-
CACHE-192-168-100-100 access-group HTTP-CACHE-ACL
    !volatile policy-map 192.168.100.100/32 seq 20 policy TORRENT-
192-168-100-100 access-group TORRENT-ACL
    !volatile policy-map 192.168.100.100/32 seq 30 policy INTERNET-
192-168-100-100
    policy-map 192.168.100.0/24 policy radius
```

```
    ip strict-clients
    ip address 192.168.100.2 255.255.255.0
    !volatile interface arp entry 192.168.100.100 0000.0000.0100
client
    parameter bng service service-definition bng-sd service-
selection bng-ss
    ip vrrp 100 address 192.168.100.1
    ip vrrp 100 priority 253
    ip vrrp 100 prohibit-preemption
    ip vrrp 100 accept
    no shutdown
```

Environment Monitor

The environmental monitor (envmon) process in iBOS system manages and monitors the operating environment of the system. This includes monitoring of temperature sensors, voltage levels, operation of air cooling fans etc.

The environment requirements, and thereby the monitoring features, are different for different types of devices. Some temperature- or voltage monitors may not exist in all systems. In some systems, fans are continuously operating while in others they are controlled by iBOS and the operating speed is automatically adjusted to fit the current conditions.

The primary purpose of the envmon subsystem in iBOS is to set and monitor environmental sensor levels. If the defined thresholds are exceeded, envmon will create log entries in the system log that can be sent to remote logging servers with the **logging** configuration commands. This allows a central network management station to receive alarms related to environmental situations and the network operator can react accordingly.

Commands used to configure envmon are:

Command	Description
environment	Configure environment settings
fans	Configure fans
show environment	Display envmon settings

File System

Waystream devices that run iBOS has a flash or NVRAM storage device integrated into the product. This storage device is used to store system images, configuration and other vital system components that are required for system operation. Some systems also have USB ports to allow the use of external storage devices, such as USB memory sticks.

The iBOS file system can store multiple files. Files can be downloaded to the system using several different protocols, the most common are TFTP and HTTP.

The most common files kept in the file system are system images. They are also perhaps the most frequently upgraded or modified files.

When an iBOS system starts, the first software executed in the system is the Bootloader. The bootloader does not contain a complete operating system but rather a subset of the most necessary utilities and features to start the system. The primary purpose of the Bootloader is to initialise system components and in particular to access the storage device file system to start the system image.

The system image is stored in an compressed format in the file system in order to reduce the size of the image. It contains all the subsystems and programs started and running in an operating iBOS system.

iBOS contains several commands to view and modify the local file system and to configure which file to use as the system image during startup.

System Time and Network Time Protocol (NTP)

The system has an internal clock which maintains system time in seconds from EPOCH (which is 00:00:00 GMT January 1st 1970). The counter is converted to the current date and time and is used to put timestamps on logging and debug events.

The network time protocol (NTP) is a protocol that allows system clocks in network devices to be synchronised to a central timing device, such as an atom clock.

System Time

The system has an internal clock used to time stamp logging information and debug output and other information in the system. It can prove quite useful to have an accurate clock in the system.

iBOS allows configuration of the correct time zone. If no timezone has been configured, iBOS assumes that it is running in UTC (GMT timezone).

Setting System Clock

To set the system clock manually, use the **clock set** command. This command is not a configuration command

Example 24 Setting System Clock

```
ibos# clock set 15:07:00 Mar 6 2002
```

The current time can be displayed using the **show clock** command.

The system time can also be set and continuously updated using the Network Time Protocol.

Setting Time Stamp Format

In the system log and debug output, the time stamp of the event can be displayed in two formats: either as the exact date and time, or as the time from system start. The default is to display the time stamp of events as the time from system start. To change this, use the **service timestamps** configuration command.

NTP

The NTP protocol is currently at version 3, specified in RFC1305 as a Draft Standard. NTP version 2, specified in RFC1119 is also known as Internet Standard 12. The original version of NTP was specified in RFC 958.

Waystream supports all three versions of NTP, although the system default is version 3.

In NTP, a number of primary clock sources are connected to the network. NTP conveys timekeeping information from these sources to time servers using IP. The protocol allows clocks to crosscheck and compensate for errors due to equipment or propagation failure (delays in links, outages etc.).

The typical implementation is a client/server model where a client transmits NTP messages to the server. The server replies to these messages and based on the information in the replies, the client is able to determine the server time in relation to its own time and adjust the local clock accordingly. It also allows the client to calculate the accuracy of the timekeeping effort.

Setting System Time with NTP

iBOS can be configured to transmit NTP messages to a central NTP server at regular intervals. This is done using **ntp** configuration commands.

Example 25 NTP Configuration

```
ibos(config)# ntp source-interface loopback1
ibos(config)# ntp server 10.0.0.42
ibos(config)# ntp interval 3600
ibos(config)# ntp version 2
```

In the above example, the system is configured to obtain the accurate system time from NTP server 10.0.0.42 once every hour, using the primary IP address of interface Loopback1 as the source-address and version 2 of the NTP protocol in the communication with the NTP server.

NTP uses UTC, and in order for the time to be correctly displayed the **clock timezone** configuration command should be used to configure the system to the correct time zone.

Logging

iBOS has an internal logging system that stores important events in a round-robin memory. This information can often be a crucial source of information for determining what is happening in the network.

The iBOS logging subsystem collects important events from other parts of the system. This information is classified into categories and levels by the originating subsystem.

The logging subsystem can store the event in a local logging buffer found in RAM, and/or forward the event to a remote logging server using the syslog protocol.

Information can also be dynamically presented to a user logged in to the system through telnet. Finally, it can be printed to the serial console port.

Logging Levels

Not all information generated by a system is important. Therefore, different levels have been introduced to define the importance of a particular log message.

When setting the logging level in iBOS, all messages at that level or lower (more important) are displayed, and higher-level messages are ignored.

Setting a high level may cause the internal logging buffer to become full. Since the buffer keeps the most recent information, detailed logging over a longer period of time should be syslogged to a remote logging server where hard disk space allows for the storage of extensive logs.

Table 9 Logging Levels

Severity Level	Level Name	Message Type
0	emergencies	System unusable
1	alerts	Immediate action required, system may be unstable
2	critical	Critical condition
3	errors	Error condition, the command was not executed
4	warnings	Warning, something went wrong but the command completed
5	notifications	A significant event that is not an error condition occurred
6	informational	Standard information
7	debug	Debugging information

The default logging level is set to **notifications** for all destinations.

Logging Destinations

There are four possible destinations for an event.

- The event can be logged to the console and thereby presented to any user connected to the serial console port. The user does not have to be logged in to the system on the console port for logging to be output on the port. The **terminal no monitor** command can be used to turn off logging events when connected on the console port.
- The event can be logged to the terminal. This means that any active remote telnet connection to the system receives the event - if the connection is set to monitoring mode with the **terminal monitor** command. If the terminal monitor is not enabled, a remote user will not see events logged to the terminal.
- The event can also be logged to the local RAM buffer. Up to 4096 lines of logging can be stored in the system RAM memory. The information is not stored beyond a system reload, so if the system loses power or is restarted any logging information in the buffer is lost.
- The event can be sent to a remote server with the syslog protocol. The server can typically be a network management station that collects logging information from multiple sources and stores them locally on a hard drive. Remote logging is preferred if it is important to maintain logging information or to use it to alert network

operators about potential error conditions in the network.

Logging to a Remote Host

To set up iBOS to forward all events to a remote logging server using the syslog protocol, the remote logging server address or hostname and the syslog logging facility to which iBOS should forward events (default is to log to the local0 facility) must be set up. The logging level of events to forward should also be configured.

Example 26 Remote Logging

```
ibos(config)# logging trap debugging
ibos(config)# logging server 192.168.5.1
ibos(config)# logging facility local5
```

In the example above, the system is configured to log all events (including debug output if debugging is activated) to the remote host 192.168.5.1 as syslog facility local5. On that system, the local5 facility should be set up with a file that will contain all logging events from this particular system. The file can be shared between multiple network devices.

The optional **logging source-interface** configuration command can be used to set the interface from which to use the primary IP address as the source for any log event sent from the system. This allows all logged events to be clearly related to the system, regardless of link status on other interfaces. Otherwise, iBOS will select the first available interface as source for logging messages. This may lead to inconsistent source address for logged messages due to link outages.

Local Logging

The information logged to the local buffer can be displayed with the **show logging** command. The log can be up to 4096 rows long. This will of course consume internal memory, but under normal conditions the system should be able to maintain a log of that size. The default buffer length is 128 rows.

The local buffer can also be displayed in reversed order, with the most current event displayed first. This is useful if the log contains many messages. To display the log in reversed order, use the **show logging reverse** command.

The buffer can be emptied using the **clear logging** command.

Remote Management

iBOS uses various tools for enabling remote management. This is described in the following sections:

- “Simple Network Management Protocol” on page 62
- “CSAR Protocol” on page 70
- “PFDP Protocol” on page 70
- “Link Layer Discovery Protocol” on page 81
- “Netflow” on page 88

Simple Network Management Protocol

The Simple Network Management Protocol, SNMP, is the Internet standard network management protocol. SNMP allows management information to be exchanged between network devices and network management stations.

The protocol was originally defined in RFC1157, but has evolved into the current version 3. In addition to the actual protocol used between network devices, there are also documents defining the description language for management information objects (SMI) and the management information base (MIB) tree.

Waystream iBOS supports SNMP version 1, 2c and 3.

In order to be manageable with SNMP, a network device must contain a SNMP agent. The agent listens to incoming SNMP requests for object identifiers within the management information base. If the request is valid, the agent responds to the request.

Network Management Systems

Many network management systems (NMS) use SNMP to find information about the devices in the network. This includes the collection of statistics, such as traffic passing through interfaces, CPU load and other parameters. Often, the network management system combines the information retrieved by SNMP with other network monitoring tools, such as ICMP echo request and reply messages (ping).

It is possible to set system configuration parameters by using SNMP, but most routers and switches today offer either customised configuration GUIs or industry-standard CLIs to allow the network administrator to set system parameters. It is therefore rare for NMS to rely entirely upon SNMP for the configuration of network devices.

The NMS is typically placed at a central location in the network. Since network management with SNMP takes place in-band (over the same connections and links as normal traffic), the NMS requires security measures to be protected from attacks. Because the NMS is able to get, and sometimes set, parameters with SNMP, an attacker may gain valuable information if the NMS system is compromised. SNMP also facilitates a notification method, in earlier versions of the protocol in the form of SNMP traps, whereby a network device with a SNMP agent may send unsolicited messages to the NMS to alert network administrators of an important condition in the network. Standard trap messages include coldstart/warmstart and interface up/down events.

The version 2 and 3 of the protocol also include the inform notification type, which require an acknowledgement from the NMS (the inform receiver) that the message has been received.

The Agent

The SNMP agent within the network device listens for SNMP messages and is typically also responsible for the transmission of SNMP notification messages (traps or informs).

The version 1 and version 2 implementations of the protocols offer a simple authentication scheme, in the form of a community string, to verify that a received SNMP set or get request should be handled. The community string is no more than a clear-text password used between the NMS and the agent.

Version 3 of the protocol has enhanced the security and offers additional options for authentication of the message.

The Structure of Management Information

SNMP is not only a protocol for exchange of information. The management framework includes a description of how each piece of information is organised and where it may be accessed.

In order for SNMP to work between NMS and network devices from different vendors, a common language for describing management information is required. In SNMP this is defined as the Structure of Management Information (SMI), which uses the syntax of Abstract Syntax Notation One (ASN.1).

SMI defines how parameters and objects that are manageable (settable, gettable or notify able) are described; what their name is, if it is a string (such as the system hostname), a value (such as the interface speed), part of a table (such as the routing table), etc.

Each object is uniquely named in a tree structure with a nameless root. The management information for IP based devices is found under .iso.org.dod.internet (numerically .1.3.6.1)

The Management Information Base

The Management Information Base (MIB) is the collection of manageable objects in a device described by SMI. The MIB often refers to the collection of all possible MIBs, while a MIB refers to the MIB definition for a particular part of the tree.

Together with SNMP, a standard MIB known as MIB-II (the Internet MIB), was defined to specify a set of management objects that every SNMP device must support. In addition to MIB-II, there are many other MIB definitions for individual protocols (OSPF, DHCP, AAA), different types of devices (hosts, routers, switches, bridges), media-types (Ethernet, fddi) and many more.

There are also MIBs that are vendor-unique devices.

Because of SMI and the structure of the MIB, any NMS that supports SNMP typically also supports the loading of MIB definitions. Therefore it is possible to identify, get and set parameters in MIB definitions custom to the managed device.

The History of SNMP

Today, version 3 of the Internet Standard Management Framework has been approved by the IESG. But it all began with SNMP version 1.

SNMPv1

The original definition for SNMP is known as Internet Standard 16 and consists of a number of RFCs.

- RFC1155 defines the SMI language (SMIv1)
- RFC1212 defines a more concise description mechanism for MIB objects
- RFC1157 defines the SNMP protocol used between agents and NMS

These documents were accompanied by Internet Standard 17, RFC1213 (MIB-II), and RFC1215 (trap notifications).

After the publication of MIB-II, it was decided that management information should be defined in many mini-MIB documents - rather than a single large Internet-standard MIB. This allowed the working-groups of the IETF to produce MIB documents focused on information relevant for the protocol developed.

SNMPv1 and Security

Already when SNMP version 1 was defined, the issues of authentication were important. SNMP version 1 therefore defines an authentication service whereby different authentication schemes could be added to ensure data authentication.

However, due to various reasons, no such authentication schemes were defined, except for a simple, rudimentary scheme known as communities. The result was that security was a known problem in SNMP, which would not be solved until the introduction of SNMP version 3.

SNMPv2

The second version of the management framework contained enhancements such as:

- expanded data types
- improved efficiency and performance
- fine tuning of the data definition language
- inform notification type

The IETF working group failed to reach all goals set up for the second generation of SNMP. The major points that failed to be resolved were authentication and privacy of data, authorisation and access-control and remote configuration capabilities for such features.

There was also some controversy within the SNMP community about how these issues should be solved. During the early 1990:s, a number of documents related to SNMP security were published. They contributed to the development of SNMP v2, but in the end, the SNMP v2 management framework failed to include a standards-based security solution.

Instead, multiple security and administrative frameworks appeared, including:

- The Community-based SNMPve (SNMPv2c) [RFC1901]
- SNMPv2u [RFC1909-1910]
- SNMPv2*

SNMPv2c was the official IETF protocol, but with no security and administration. Both SNMPv2u and SNMPv2* had security solutions, but without the endorsement from the IETF.

SNMPv3

When the SNMP version 3 working group was created, it was chartered to produce a single set of specifications for the next generation of SNMP. The specifications were based on a combination of SNMPv2u and SNMPv2*.

The working group focused on security and administration, including authentication and privacy, authorisation and view-based access control and standards-based remote configuration of these features.

SNMPv3 is built on SNMPv2 draft standard documents. With the necessary additions of security and administration lacking in earlier versions of the protocol, SNMPv3 could be thought of as SNMPv2 with additional security and administration capabilities.

Configuring SNMP

SNMP configuration in iBOS is divided into two major sections.

The first is the general agent related configuration parameters, including settings for the agent itself, the definition of views for security control and specific MIB parameter settings.

The second is the protocol version related configuration parameters, found in the **snmp-group** configuration contexts.

Configuring Agent Parameters

The **snmp agent** configuration command control many of the iBOS agent settings.

UDP-Port

To set the UDP port to which the agent listens for SNMP requests to anything but the default, use the **snmp agent udp-port** configuration command. The default is for the agent to listen for SNMP requests on UDP port 161.

Retransmit-Queue

To configure the SNMP notification message retransmit-queue parameters use the **snmp agent retransmit-queue** configuration command.

The retransmit-queue is used to temporarily store notification messages that were not possible to send due to routing lookup failure for the notification message destination. This can happen if a link event, either on a directly connected interface or somewhere else in the network, causes iBOS to loose the route to the destination host.

The retransmit-queue allows iBOS to store the notification message in memory for some duration of time (configurable to 1-60 seconds). If the route to the destination host reappears, iBOS may retransmit the notification message. The result is that the trap or inform message is received at the notification host, although with some delay (1-60 seconds depending on configuration and when the route becomes available).

It is possible to set a retransmit-queue length of 0, indicating that no notifications should be stored in the retransmit-queue (effectively disabling this feature).

Example 27 SNMP Retransmit-Queue

```
ibos# conf t
ibos(config)# snmp agent retransmit-queue length 10
ibos(config)# snmp agent retransmit-queue timeout 20
ibos(config)# end
ibos#
```

In the example above, the notification retransmit-queue is set to contain a maximum of 10 messages for a maximum of 20 seconds. If an 11th message is created and not possible to send, the oldest message of the 10 already in the queue will be dropped and the new message entered into the queue. Any message in the queue older than 20 seconds will also be dropped.

Inform Messages

The SNMP agent also has an internal queue for inform messages that are not successfully delivered to the destination host. The retransmit-queue described above handles queueing of messages if the destination host is unreachable in the routing.

The inform message queue, with parameters for **retries** and **timeout**, is used to store inform messages that were successfully transmitted (no routing failure or other indication of error) but for which there was no response from the destination host. If the destination host fails to acknowledge the reception of the inform message, the message can be temporarily stored in the inform message queue.

To configure the maximum number of attempts to retransmit an inform message, use the **snmp agent inform retries** configuration command.

To set the timeout between inform messages when retransmitting to a destination host, use the **snmp agent inform timeout** configuration command.

Configuring MIB-II System Parameters

In the Internet-standard MIB, MIB-II, there are parameters for system name, location and contact that can be set by the network administrator.

iBOS uses the configured hostname in response to any SNMP requests for the system.sysName MIB variable.

To set the system.sysLocation or system.sysContact MIB variables, use the **snmp mib system** configuration command.

Example 28 SNMP MIB-II System Parameters

```
ibos(config)# snmp mib system location “Building 10, West Tasman
Drive”
```

Configuring View Access

When a read or write request is received by the agent, the agent authenticates the request. Among the things checked is if access to the MIB variable is allowed. This is done by the mapping between a user/community and a MIB view.

The MIB view is created with the **snmp view** configuration command. The view consists of a number of include- and exclude statements for branches of the MIB tree that the view either permits or denies access to. It is not necessary to specify each MIB variable individually, but possible to specify a certain branch of the tree, such as the system section of MIB-II if access to variables in that section are permitted or denied.

Example 29 Permitting Access to the Entire MIB Tree

```
ibos(config)# snmp view ALLTREE include .1
```

Example 30 Access to MIB-II System only

```
ibos(config)# snmp view SYSONLY include .1.3.6.1.2
```

Example 31 Complex MIB Access

```
ibos(config)# snmp view COMPLEX include .1.3.6.1.2
ibos(config)# snmp view COMPLEX exclude .1.3.6.1.2.1.1.9
ibos(config)# snmp view COMPLEX exclude .1.3.6.1.2.1.2
```

The SNMP view COMPLEX configured in the example above grants access to everything in the standard MIB-II tree except the .1.3.6.1.2.1.1.9 and .1.3.6.1.2.1.2 branches of that tree.

Configuring SNMP-Groups

To set up configuration for notifications, communities, users and other protocol and version related configuration, the **snmp-group** configuration command is used.

Snmp-groups can be named, for instance to separate configuration if different service provider have SNMP access to the system, but it is not required.

Regardless of snmp-group name, community names, user names and similar parameters must be unique in the system.

The configuration parameters for snmp-group version 1 and snmp-group version 2 (version 1 or version 2c of the SNMP protocol) are virtually the same, separated only by the version to distinguish the protocol version used.

Version 1/2c Communities

To configure a version 1 or version 2c community for read or read/write access to the system, use the community configuration command

Example 32 version 1 or 2c Community

```
ibos(config)# snmp view PUBLIC include .1
ibos(config)# snmp-group version 1
ibos(config-snmp-group)# community public read-only view PUBLIC
```

This example creates a read-only view named PUBLIC to which any request with the community public will be checked. Because the entire MIB is included in the view, this configuration grants any SNMP request with community public read-only access to any MIB variable the agent understands.

Version 1/2c Notification

To configure version 1 or version 2c notification, use the **notify** configuration command.

Version 1 of the protocol can handle notification messages of type trap only.

Version 2 can send both trap and inform messages.

Example 33 Configuring version 1 Notification

```
ibos(config)# snmp-group version 1
ibos(config-snmp-group)# notify 192.168.1.11 community public trap
```

The configuration in the example above configures iBOS to send SNMP traps for version 1 to the host 192.168.1.11 whenever a trap is generated by the system.

Version 3 Authentication

For SNMP version 3, authentication of messages can be configured as no authentication, no encryption or authentication, no encryption or authentication, encryption.

To configure the authentication method to use, use the **authentication** configuration command in snmp-group version 3 context.

Version 3 User

In SNMP version 3, the community system with clear text passwords has been replaced by a much more powerful user system that allows multiple levels of security and authentication.

An SNMP user is a remote system that is allowed to access a defined part of the MIB tree in the system. It can also be the local systems user for notifications to a remote network management system.

The user is configured with a password which is used to authenticate the communication.

To configure an SNMP user, use the **snmp-user** configuration command. The username together with a password of at least 8 characters must be configured.

For each **snmp-user** it is also possible to specify what **snmp view** to use for read and write access to the MIB.

Example 34 Configuring SNMP version 3 User

```
ibos# configuration terminal
ibos(conf)# snmp-group version 3
ibos(conf-snmp-group)# snmp-user nocmgmt password afUvkl34g
ibos(conf-snmp-group)# snmp-user nocmgmt readview all
ibos(conf-snmp-group)# snmp-user nocmgmt writeview all
```

The configuration in the example creates the snmp-user nocmgmt, sets a clear text password to be encrypted using the md5 algorithm (default) and assigns the snmp view all to both read and write access.

Version 3 Notification

SNMP version 3 notifications can be of both trap and inform type. The snmp-user to use by the system when notifying the remote host is configured with the **notify** command.

Example 35 Configuring version 3 Notification

```
ibos# configuration terminal
ibos(config)# snmp-group version 3
ibos(config-snmp-group)# notify 192.168.1.1 user asrsys password
dmIL8wj
ibos(config-snmp-group)# notify 192.168.1.1 type inform remote-
engine-id 0x80002457030008ae000220
```

The configuration example configures inform notifications to the remote host 192.168.1.1 with engine-id 0x80002457030008ae000220, using the asrsys user. The remote-engine-id is optional and allow the otherwise probed engine-id for the remote management system to be set directly in the system.

Agent Information

The **show snmp** CLI command displays information about the agent activity in iBOS.

The output includes the system engine-id, the number of packets of different types sent and received, and information about any configuration errors the current configuration may include. Such errors may result from inconsistent configuration, such as configuring a community with a view that has not been created.

Example 36 Show SNMP Information

```
ibos# show snmp
SNMP Agent Information
  Engine-ID : 0x80002457030008ae0000a0

  Received packets
    Incoming packets : 3720
      Type: 3720 total request vars, 0 total set vars,
            0 total get requests, 3720 total get next requests,
            0 set requests, 0 get responses, 0 traps
      Bad : 0 bad version, 0 bad community names
            1374 bad community uses, 0 ASN parse errors,
            0 too big, 0 no such name, 0 read-only,
            0 bad values, 0 generic error
    Outgoing packets : 3720
      Type: 0 get requests, 0 get next requests
            0 set request, 3720 get responses, 0 traps
      Bad : 0 too big, 0 no such name,
            0 bad values, 0 generic error
            0 silent drops
   No warnings for snmp configuration.
```

CSAR Protocol

The CSAR protocol is used by the PacketFront Software Solutions network control system BECS™ version 3 to control iBOS configuration. CSAR is a protocol supported by all Waystream hardware.

The CSAR protocol is a proprietary protocol.

Configuring CSAR

By default, the system hostname is used to identify the iBOS system with the PDP. If other names than the hostname wish to be used, this can be configured with the **csar server** configuration command.

The configuration command **csar recovery** triggers a BOOTP retry if the contact to BECS is lost.

The configuration command **csar source-interface** configures the CSAR source interface.

The configuration command **csar tos** configures the termination of service setting.

The **ip dhcp csar** command is used to intercept DHCP DISCOVER packets and to open a client context to a BECS version 3.0 (or later) system.

The **show csar** command displays the current connection status for the CSAR protocol.

PFDP Protocol

The PacketFront Device Protocol (PFDP) is a legacy, stateless link-local protocol that provides means for Waystream devices to share limited system information. The protocol is used by legacy PacketFront Network Products networks, as well as other third party CPE devices, such as the DRG products from Genexis.This information is useful for troubleshooting, network mapping, statistics and similar purposes. The protocol also controls firmware upgrade functionality.

The protocol is still implemented in Waystream products, such as the ASR and MS4000-series switches.

Protocol Overview

PFDP is a non-routable protocol. It is only local to the link it is operating on. Packets are addressed to all devices on the link by usage of the layer2 multicast address 0108.AE00.0013 (allDevices), but may also be unicasted directly between hosts.

The protocol use a SNAP header with OIU set to 0x0008AE and EtherType 0x0001.

Protocol Operation

Each PFDP speaking device regularly transmits a hello-message with information about the device. Parameters include: hostname, serial number, uptime, known PFDP neighbours, version of operating system running in the device and other parameters.

The transmitting device specifies how long the information is valid by setting a timeout in seconds for the data.

By default, PFDP is only used on interfaces designated as uplink interfaces. An uplink interface is assumed to lead towards the network core (part of the transport network). PFDP is normally only sent on uplink ports.

The announcement of PFDP on an interface can be enabled or disabled with the **pfdp enable** interface command.

The information received with PFDP is stored in an internal database in iBOS. The **show pfdp** command can be used to display configuration and statistics for PFDP operation in the system.

Example 37 show pfdp Output

```
Hello PDU Transmission interval is 30 seconds
 Default Lifetime for received PDUs (if not supplied) is 180 seconds
 Lifetime of transmitted PDUs from this system is 180 seconds

...

 Interface gigabitethernet32
  PFDP is enabled
  Firmware push is enabled
  1 neighbour(s)
  Maximum neighbours: 16
  Maximum database entries: 64  (per neighbour)
  908 packets in, 0 erroneous packets in
  8347 packets out
```

The output displayed in the example above is a sample of the output from the **show pfdp** command. It displays basic PFDP settings such as the hello packet interval, lifetime of information received and then list information about each interface configured with pfdp in the system. In this example, interface fastethernet22 has one active PFDP neighbour and that the firmware update option is enabled.

The **show pfdp neighbour** command display a list of all known PFDP speaking devices directly connected on any interface in the system.

Example 38 Output from show pfdp neighbour

```
asr> show pfdp neighbour
Hostname      LocalIf   RemotIf         Platform   LastAct      OS
Version
-------------------------------------------------------------------
-----------
CPS             fa22      fa1     CPS TX rev: B      20s      CPS
v1.1.2-R
tnv-ring7        gi1      gi2         ASR4124C       6s      iBOS
v2.1.0
tnv-ring9        gi2      gi1         ASR4124C      13s      iBOS
v2.1.0
```

In the example above, the output from the **show pfdp neighbour** command is provided. The output shows that this system has three PFDP speaking devices on connected interfaces. Information includes the hardware type of those devices and the operating system version running, the local interface connecting to each device and which interface on each device that connects to this system.

Additional information about each known neighbour can be displayed with the **show pfdp neighbour detail** command. This command display any specific information sent by all neighbours, such as interface statistics, serial number and other information.

Example 39 Output from show pfdp neighbour detailed

```
Interface fastethernet22
  Neighbour 0008.ae02.0171 state unidirectional
   Entry created 3h11m36s ago
   Last Activity 26s ago
   768 packets in
   26 database entries

   Hostname   : CPS
   OS Version : CPS v1.1.2-R
   HW Platform: CPS TX rev: B
   Serial No. : 0008ae020171
   Uptime     : 15h53m5s

    Transmit Interface: fa1

    Port fa1 State: Up Full-Duplex 100Mbit VLAN-Info 0
    Port fa2 State: Down VLAN-Info 0
    Port fa3 State: Down VLAN-Info 0
    Port fa4 State: Down VLAN-Info 0
    Port fa5 State: Down VLAN-Info 0
    Port fa6 State: Down VLAN-Info 0
    Port fa7 State: Down VLAN-Info 0
    Port fa8 State: Down VLAN-Info 0

    Port fa1 Statistics
     TxOctets 1764730, TxDropPkts 0, TxCollisions 0
     TxMulticastPkts 3782, TxUnicastPkts 0, TxBroadcastPkts 0
     TxDeferredTransmit 0, TxFrameInDisc 0

     RxOctets 0, RxUndersizePkts 0, RxOversizePkts 0
     RxJabbers 0, RxAlignmentErrors 0, RxFCSErrors 0
     RxGoodOctets 0, RxDropPkts 0, RxFragments 0
```

```
    RxUnicastPkts 0, RxMulticastPkts 0, RxBroadcastPkts 0
    RxExcessSizeDisc 0, RxSymbolError 0
   Packet Sizes In:       0-64      65-127     128-255     256-511
                           0           0           0           0
    Packet Sizes In:   512-1023   1024-1522
                           0           0

  Port fa2 Statistics empty
  Port fa3 Statistics empty
  Port fa4 Statistics empty
  Port fa5 Statistics empty
  Port fa6 Statistics empty
  Port fa7 Statistics empty
  Port fa8 Statistics empty
```

The following table describes the counters for the number of packets (including error packets) for packet size distributions within each size range.

Table 10 PFDP Counters

Counter	Description
TxOctets	The total number of good bytes of data transmitted by a port (excluding preamble but including FCS).
TxDropPkts	This counter is incremented every time a transmit packet is dropped due to lack of resources (e.g., transmit FIFO underflow), or an internal MAC sublayer transmit error
TxCollisions	The number of collisions experienced by a port during packet transmissions.
TxMulticastPkts	The number of good packets transmitted by a port that are directed to a multicast address. This counter does not include errored multicast packets or valid broadcast packets.
TxUnicastPkts	The number of good packets transmitted by a port that are addressed to a unicast address.
TxBroadcastPkts	The number of good packets transmitted by a port that are directed to a broadcast address. This counter does not include errored broadcast packets or valid multicast packets.
TxDeferredTransmit	The number of packets transmitted by a port for which the first transmission attempt is delayed because the medium is busy.
TxFrameInDisc	The number of valid packets received which are discarded by the forwarding process due to lack of space on an output queue. Not maintained or reported in the MIB counters. This attribute only increments if a network device is not acting in compliance with a flow control request.
RxOctets	The number of bytes of data received by a port (excluding preamble, but including FCS), including bad packets.
RxUndersizePkts	The number of good packets received by a port that are less than 64 bytes long (excluding framing bits, but including the FCS).
RxOversizePkts	The number of good packets received by a port that are greater than 1518 or 1522 bytes inclusive, depending on whether IEEE 802.1Q is disabled or enabled respectively (excluding framing bits but including the FCS).
RxJabbers	The number of packets received by a port that are longer than 1522 bytes and have either an FCS error or an alignment error.
RxAlignmentErrors	The number of packets received by a port that have a length (excluding framing bits but including FCS) between 64 and 1522 bytes, inclusive, and have a bad FCS with a nonintegral number of bytes.
RxFCSErrors	The number of packets received by a port that have a length (excluding framing bits, but including FCS) between 64 and 1522 bytes inclusive, and have a bad FCS with an integral number of bytes.

Table 10 PFDP Counters

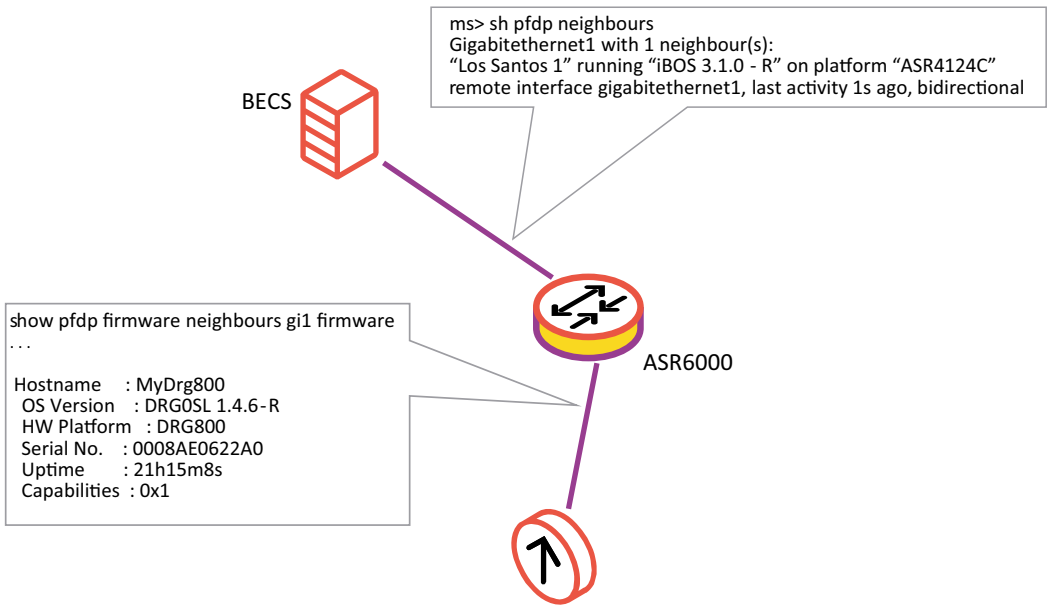
Counter	Description
RxGoodOctets	The total number of bytes in all good packets received by a port (excluding framing bits, but including FCS).
RxDropPkts	The number of packets received by a port that were dropped due to security conflict when MAC address security Control is enabled
RxFragments	The number of packets received by a port that are less than 64 bytes (excluding framing bits) and have either an FCS error or an alignment error.
RxUnicastPkts	The number of good packets received by a port that are addressed to a unicast address.
RxMulticastPkts	The number of good packets received by a port that are directed to a multicast address. This counter does not include errored multicast packets or valid broadcast packets.
RxBroadcastPkts	The number of good packets received by a port that are directed to the broadcast address. This counter does not include errored broadcast packets or valid multicast packets
RxExcessSizeDisc	The number of good packets received by a port that are greater than 1536 bytes (excluding framing bits but including the FCS) and were discarded due to excessive length. The RxOversizePkts counter alone is incremented for packets in the range 1523–1536 bytes inclusive, whereas both this counter and the RxOversizePkts counter are incremented for packets of 1537 bytes and higher
RxSymbolError	The total number of times a valid length packet was received at a port and at least one invalid data symbol was detected. Counter only increment once per carrier event and does not increment on detection of collision during the carrier event.

PFDP Firmware Upgrade

Legacy support of PFDP includes the feature upgrade firmware and control certain features in legacy CPE products, such as Genexis DRG and PacketFront CPS. The information in this section is provided for reference purpose only.

PFDP adds software upgrade functionality for PFDP enabled CPE products to the ASR routers. BECS can upgrade firmware for CPE hardware via the ASR. The ASR can push the firmware to a CPE that is directly connected.

Figure 5 BECS upgrades DRG devices connected to Waystream MS4000 using PFDP



For more information, see **show pfdp** and **pfdp firmware.**

Active PFDP CPE Configuration

Active PFDP is a way to configure PFDP enabled CPE products, (CPS and DRG models), so that the device behaves differently from the default behavior. Active PFDP can be used for:

- debugging
- tuning the CPS
- workarounds for malfunctioning devices

This functionality has the following features:

- Enable/Disable individual ports on CPS
- Setting media 10-half/full and 100-half/full
- Configure the QoS settings in the CPS, per port
- Enable/disable port protect on each CPS port
- IGMP robustness
- Immediate-leave
- Enable/disable/configure ingress- and egress shaping per port
- Force Reboot of CPS

Configuration to the CPS will be pushed out with a timeout. A configuration is only valid during this timeout. When the timeout expires, the CPS returns to its default values. iBOS has to therefore periodically resend the preferred configuration. The timeout follows the normal pfdp lifetime configuration.

The interface names for the uplink port (both FX or TX) is “uplink1” and the other interfaces are “fa1” to “fa7” (TX model), or “fa1” to “fa8” (for FX models).

Note: *For this functionality, the CPS units must have firmware version 2.2.0 or 3.1.0, or later. Some limited support is also provided in version 1.4.0.*

For more information on configuring CPE settings, see **remote unit cps** and **reset remote pfdp**.

Active PFDP Support for DRG CATV

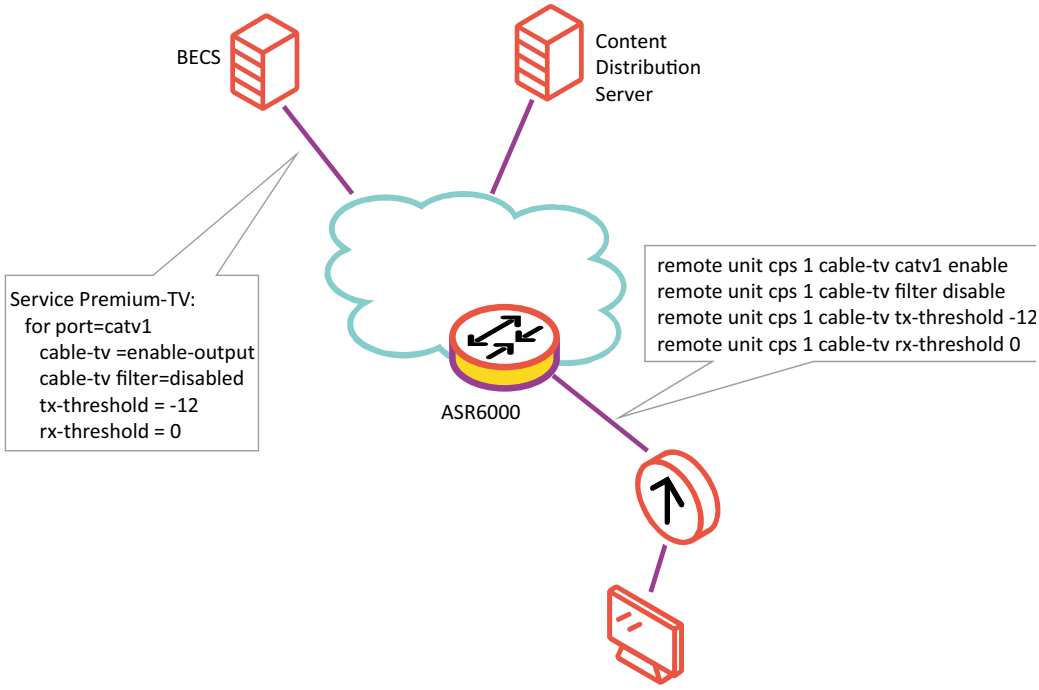
Some DRG products can include an optical-electrical cable TV converter module that can be turned on or off by the DRG.

The module also contains a filter that can be used to block part of the RF spectrum which makes it possible to turn on and off premium cable TV content located in blockable parts. Via PFDP, iBOS can configure the following:

- the output can be turned on or off (default is off)
- the filter can be turned on or off (default is on)
- the input and output power levels can have thresholds. This makes it possible to trigger an alarm or some other notification event when the signal is outside the defined interval

This added support in iBOS also makes it possible for BECS to define services that control cable-TV channels for RF video customers.

Figure 6 CATV Management for DRG 600 using Active PFDP



To configure the CATV module on the DRG 600, the following commands have been implemented for PFDP-aware interfaces:

- **remote unit <n> cps cable-tv <port> output-enable**
- **remote unit <n> cps cable-tv <port> filter-disable**
- **remote unit <n> cps cable-tv <port> rx-power-thresholds <min> <max>**
- **remote unit <n> cps cable-tv <port> tx-power-thresholds <min> <max>**

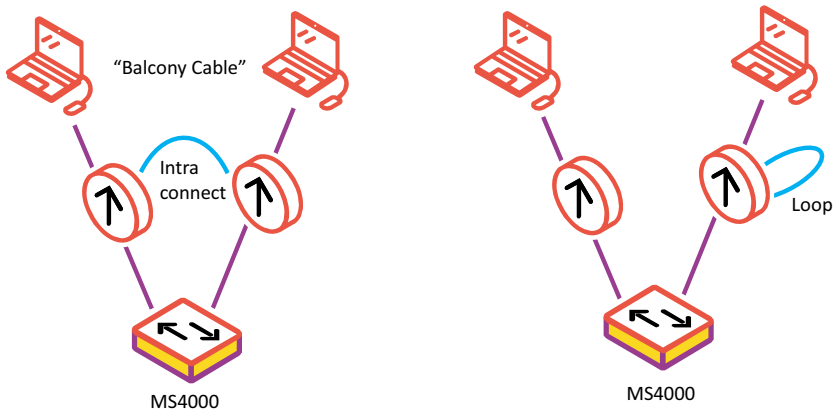
Error Disable

iBOS contains a port error-disable function that can be used to protect the network against loops. The spanning-tree protocol is the primary solution to avoid loop topologies, but on edgeports that connect hosts and servers, it may not be a requirement to operate spanning-tree as it is not likely that loops occur on such ports.

However, these topologies can experience network loops due to user intervention. In broadband networks, the concept “balcony cable” is well known; two users in different apartments in the same building connect a cable between the home networks for faster gaming and unknowingly create a loop between the two ports in the building switch that connect their apartments to the city network.

Another example is a user with a simple desktop switch where two ports on the same switch are connected together and inadvertently create a loop connection.

Figure 7 Network Loops



Both of these errors can cause significant impact on the switch and network as Ethernet frames will be looping between ports or on the same port.

The PFDP protocol has a feature to detect this error and to log or disable the affected port should the problem occur. If the ports are enabled to transmit PFDP frames, the switch will be able to detect its own PFDP frame coming back on the same port (in the loop scenario) or on another port (in the intraconnect scenario), both of which are a sign of something not being right in the network.

The errdisable command turns own the detection of either loop or intraconnect, or both. It is a global configuration command which means it turns on the functionality for the whole switch. You can then disable the feature on the specific interfaces where it is not wanted or required, once turned on.

```
R3(config)# errdisable ?
  default     Default config interfaces
  nni         NNI interfaces
  uni         UNI interfaces
  uplink      Uplink interfaces
```

The errdisable feature can be turned on related to interface types. The default type covers all downlink interface, usually the gige ports on a switch, right out of the box. In service provider networks using service VLAN topologies, the UNI and NNI port types are used. These types are covered in a different course, but in this context if determines how errdisable operates on those port types. Finally, uplink refers to ports that are defined as uplinks, enabling errdisable only on such port types.

```
R3(config)# errdisable default ?
  pfdp-intraconnect  Local interface intraconnect detection by PFDP
  pfdp-loop          Single-interface loop detection by PFDP
```

You then get to select the two options to detect; the loop and the intraconnect topology.

```
 R3(config)# errdisable default pfdp-intraconnect ?
  log         Only log ErrDisable events
  shutdown    Shutdown ErrDisable interfaces (and log)
```

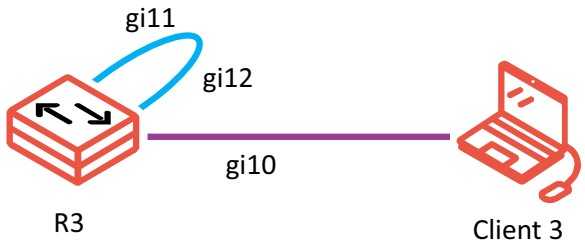
Finally, there are the log or shutdown option. If you select log, then upon detection of a errdisable condition the system will only generate a log message. With shutdown, the port is put into errdisable state and a log message is generated.

It is also possible to configure a timeout. When the port enters errdisable state, the timer is launched. When the timer expires, the port will recover from errdisable state automatically. If a new errdisable situation is detected, the port goes back into errdisable state. So the timeout mechanism allows your users to detect the problem and get reconnected without any intervention from you. The default timeout is 5 minutes.

So looking at our example topology, we have deliberately connected a cable between gi11 and gi12 on R3. This is the equivalent of an intraconnect loop or "balcony-cable".

R3 is a L3 switch, so by default is does not have VLAN memberships, but that does not matter for the errdisable feature, it can operate on both native and VLAN ports. They key criteria, however, is to have pfdp enable on the port so that PFDP frames are sent.

Figure 8 Errdisable Example




```
R3(config)# errdisable default pfdp-intraconnect shutdown timeout 5

R3(config)# int gi11
R3(config-if-gi)# pfdp enable
R3(config-if-gi)# vlan mem 1
R3(config-if-gi)# vlan un 1
R3(config-if-gi)# no shut
R3(config-if-gi)# int gi12
R3(config-if-gi)# no shut
R3(config-if-gi)# vlan mem 1
R3(config-if-gi)# vlan un 1
R3(config-if-gi)# pfdp enable
R3(config-if-gi)#

R3(config-if-gi)# 3w0d: %INTERFACE-3-ERRDISABLE:
gigabitethernet12: pfdp-intraconnect: Intra-connect detected with
interface gigabitethernet11: shutting down interface
R3(config)# 3w0d: %INTERFACE-3-RECOVERY: gigabitethernet12:
Recovering from pfdp-intraconnect: Intra-connect detected with
interface gigabitethernet11
3w0d: %INTERFACE-3-ERRDISABLE: gigabitethernet11: pfdp-
intraconnect: Intra-connect detected with interface
gigabitethernet12: shutting down interface


R3# show errdisable
State legend: active   - ErrDisable activated
              disabled - ErrDisable triggering disabled

Interface                     Facility    State   Action    Expire
-------------------------------------------------------------------
--------------
```

```
gigabitethernet12        pfdp-intraconnect   active shutdown
5s

R3# show errdisable detailed
State legend: active   - ErrDisable activated
              disabled - ErrDisable triggering disabled

Interface                      Facility    State   Action    Expire
-------------------------------------------------------------------
--------------
gigabitethernet12       pfdp-intraconnect   active shutdown
5s
  "Intra-connect detected with interface gigabitethernet11"



R3# sh int gi12
gigabitethernet12 is ErrDisabled, administratively up, line
protocol is down (17s ago)
  Duplex auto (n/a), Speed auto (n/a)
  Hardware address 0008.ae8c.9c80
  Interface availability is 0.1%
  Created 3W14h ago, 41m21s of accumulated uptime
  IP MTU 1500 bytes, Ethernet MTU 1538 bytes
  Inet6 address fe80:c::208:aeff:fe8c:9c80/64
  Vlan membership: 1
  10 seconds input rate 0 bits/sec, 0 packets/sec
  10 seconds output rate 0 bits/sec, 0 packets/sec
  55 packets input, 32466 bytes, 0 input errors
    0 input queue drops, 0 unknown protocols
    34 broadcasts, 21 multicasts, 0 pause
    0 runts, 0 oversized, 0 CRC
    0 jabbers, 0 fragments, 0 symbol, 0 alignment
  62 packets output, 46127 bytes, 0 output errors
    0 output queue drops
    31 broadcasts, 31 multicasts, 0 pause
    0 collisions, 0 deferred

   ErrDisable:
   pfdp-intraconnect (shutdown) triggered 17s ago:
     "Intra-connect detected with interface gigabitethernet11"
```

Since the transmit interval of PFDP is slow, like very minute or so, it can take a while before the errdisable situation is detected, but reaction is immediate.

You can also force an interface to recover from errdisable state earlier by the clear errdisable command.

```
R3# clear errdisable gigabitethernet12
*Feb 25 03:03:22: %INTERFACE-3-RECOVERY: gigabitethernet12:
Recovering from pfdp-intraconnect: Intra-connect detected with
interface gigabitethernet12
```

It is not possible to determine which of the ports involved in the loop that gets into errdisable state. It is the interface where the PFDP frame is wrongly received that will be disabled.

Link Layer Discovery Protocol

The Link Layer Discovery Protocol (LLDP) is a stateless, vendor-independent, layer 2 neighbour discovery protocol (as detailed in 802.1AB-2009). It provides means for Waystream networked devices to advertise their information and capabilities to their directly connected neighbours. The iBOS LLDP is supported on physical interfaces and can discover up to one device per port. It supports only the nearest bridge agent.

Purpose

- Used by networked devices to advertise their identity, capabilities as well as their neighbours.
- Stores the device information received from other network devices in the management domain.
- Enhances the ability to facilitate management of network resources and simplify networking tasks for administrators to easily map the network.

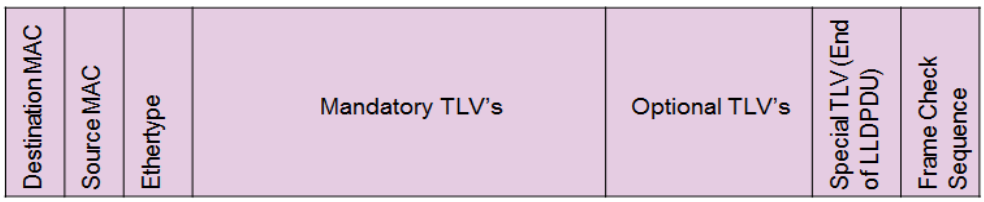
Frame Structure

LLDP frames are sent at fixed intervals on each port that runs LLDP. LLDP data units (LLDPDU's) are sent inside Ethernet frames.

An LLDP device shares its information such as chassis identification, port identification, port description, system name and description, device capability (as router and bridge), IP/MAC address etc., at fixed intervals in the form of Ethernet frames. The information is advertised by grouping attributes in the form of Type-Length-Value (TLV) structures.

There are some mandatory Type-Length-Value (TLV) fields that mark the beginning of every Ethernet frame. The mandatory TLV's include chassis identifier, port identifier and time-to-live. Apart from these mandatory TLV's, there are even few optional TLV's and a special TLV called end of LLDP Data Unit (LLDPDU) TLV that marks the end of frame.

Figure 9 Frame Structure



Destination MAC

The destination multicast address is defined as 01:80:c2:00:00:0e

Source MAC

Advertises the stations address

Ethertype

The Ethertype is defined as 0x88CC

Mandatory TLV's

Mandatory TLV's are advertised once the LLDP is enabled. These include,

Table 11 Mandatory TLV's

Name	Type	Description
Chassis Identifier	1	Advertises the system Media Access Control (MAC) address of the transmitting station to uniquely identify the device
Port Identifier	2	Advertises the port ID i.e., name of the interface to uniquely identify the port that sends the TLV
Time-to-Live	3	Indicates the time in seconds that the LLDP agent to store the information in database. The record or information should be deleted once the value reaches 0 from the default 120 sec.

Optional TLV's

Optional TLV's are advertised only when so enabled by default of during configuration. These include,

Table 12 Optional TLV's

Name	Type	Description
Port Description	4	Advertises the port description i.e., display-string that is configured to the port
System Name	5	Advertises the system name, usually referred to as the hostname
System Description	6	Advertises the information about the software and current version
System Capability	7	Advertises the primary functions of the system including bridge and router capabilities.
Management Address	8	Advertises the first IPV4 address from the IP address table. The preference order for advertising the IPv4 address is gigabitethernet > tengigabitethernet > loopback
Organizational Specific TLV's	127	Advertises the organizational specific TLV's. iBOS LLDP supports, • IEEE 802.1 optional TLV's – port VLAN identifier, and VLAN name. • IEEE 802.3 optional TLV's – MAC/PHY configuration or status, link aggregation, and maximum frame size.

Note: *The size of the LLDP Data Unit (LLDPDU) exceeds the MTU size, if more number of vlans (>900) are configured on any lldp enabled interface. In this case, some of the VLAN Name TLV's are excluded and a warning log message is displayed as shown.*

```
00:00:53: %LLDP-4-SEND: packet size is more than MTU size on
interface gigabitethernet15. So some of the TLV's are excluded!!!
```

Note: *The VLAN Name string is kept empty in the VLAN Name TLV, since iBOS does not support vlan names.*

Special TLV (End of LLDPDU)

This field marks the end of Ethernet frame.

Frame Check Sequence

The purpose of this field is to determine whether the frame experienced transmission errors in its journey through the network. It is 4 bytes in length

Operating Modes

LLDP operates in three operating modes,

- Transmit only
- Receive only
- Both transmit and receive

By default, LLDP is disabled on all supported interfaces. One has to enable LLDP on an interface to send or receive LLDP packets.

Table 13 Operating Modes

Mode	Command	Purpose	Description
Transmit only	lldp transmit	Enables interface to transmit lldp packets	In this mode, packets which holds the information about the capabilities and current status of local system are just transmitted but the received packets are discarded
Receive only	lldp receive	Enables interface to receive lldp packets	In this mode, packets which holds the information about the capabilities and current status of local system are just received and processed
Transmit and Receive	lldp transmit-and- receive	Enables interface to transmit and receive lldp packets	In this mode, packets which hold the local system capabilities are being transmitted and the packets which specify the remote system capabilities are received and processed

LLDP Timers

The timers associated with LLDP are,

- *LLDP Packet Frequency Timer (tx-interval):* This timer specifies the time gap maintained while sending the LLDP updates to neighbour.
- *Hold Time:* This timer specifies that a receiving device should store the information about its neighbour before aging it.

  Hold Time or Time-to Live (TTL) = tx-interval * hold-multiplier

LLDP Configuration

By default, iBOS LLDP is disabled on all supported interfaces. The default LLDP global and interface configurations are as shown,

Table 14 Default LLDP Configuration

Configuration	Feature	Default Setting
Global Configuration	lldp hold-multiplier	4 sec
	lldp tx-interval (packet update frequency)	30 sec
Interface Configuration	lldp transmit-and-receive	Disabled on all interfaces
	lldp tlv-select-basic-tlv	Enabled to send all basic TLV's
	lldp tlv-select dot1-tlv	Disabled to send all IEEE 802.1 organizationally specific TLV's
	lldp tlv-select dot3-tlv	Disabled to send all IEEE 802.3 organizationally specific TLV's

Monitoring and Maintenance Commands

The commands used for monitoring and maintaining LLDP packets include,

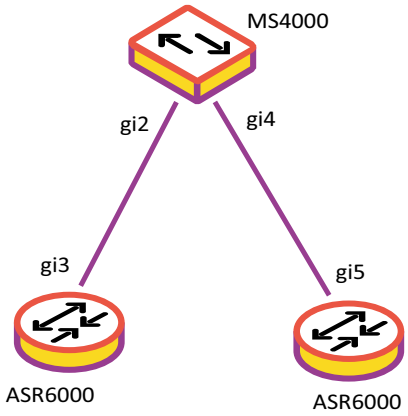
Table 15 Monitoring and Maintenance Commands

Command	Description
show lldp [detailed]	Displays transmission delay and hold multiplier values. With detailed command, it displays system details like Chassis ID, System Name, System Description, Management IP and Capability
show lldp interface brief	Displays a table of LLDP Transmit and Receive status for all interfaces
show lldp neighbours [interface<ifname> [detail]	Displays information about neighbours, including Chassis information, Hold time settings, Capabilities, and Port ID
show lldp statistics [interface <ifname>] [brief]	Displays LLDP related statistics like the number of LLDPDU transmitted, received, discarded or unrecognised. When specifying ports, only the statistics from the given port are reported
clear lldp neighbours [interface <ifname>]	Deletes neighbour information
clear lldp statistics [interface <ifname>]	Resets the traffic counter to zero on interfaces
debug lldp	Turns ON lldp debugging

Basic LLDP Configuration Example

In the following configuration example, one MS4K and two ASRs are connected in a small network. On ASR1, all basic-tlv's and dot1-tlv's are configured on gigabitethernet3 port and on ASR2, all basic-tlv's and dot3-tlv's are configured on gigabitethernet5 port. On MS4K, all TLV's are configured on gigabitethernet2 and gigabitethernet4 ports.

Figure 10 LLDP Configuration Topology



Example 40 Basic LLDP Configuration Example

MS4K:

```
hostname MS4k

interface gigabitethernet2
 lldp transmit-and-receive
 lldp tlv-select dot1-tlv all
 lldp tlv-select dot3-tlv all
 vlan member 50
 port aggregator join aggregator1
 no shutdown

interface gigabitethernet4
 lldp transmit-and-receive
 lldp tlv-select dot1-tlv all
 lldp tlv-select dot3-tlv all
 no shutdown

interface gigabitethernet5
 no shutdown

lldp tx-interval 50
lldp hold-multiplier 2
```

ASR1:

```
hostname ASR1

interface gigabitethernet3
 lldp transmit-and-receive
 lldp tlv-select dot1-tlv all
 vlan member 100
```

```
 no shutdown

interface gigabitethernet4
 no shutdown

lldp tx-interval 20
lldp hold-multiplier 2
```

ASR2:

```
hostname ASR2

interface gigabitethernet5
 lldp transmit-and-receive
 lldp tlv-select dot3-tlv all
 port aggregator join aggregator5
 mtu 2360
 no shutdown

interface gigabitethernet6
 no shutdown
```

From the basic LLDP configuration example, the output from the show lldp neighbour detailed command on MS4k is shown in example 38. The output shows that this system has two LLDP speaking devices on connected interfaces.

Information displayed on gigabitethernet ports are:

- gigabitethernet2 port: basic-tlv's and dot1-tlv's
- gigabitethernet3 port: basic-tlv's and dot3-tlv's.

Example 41 Output from show lldp neighbour detailed

```
ms4k# show lldp neighbours detailed
Total neighbours: 2

Interface:    gigabitethernet2, via: LLDP, RID: 13, Time: 0 day,
00:08:03
  Chassis:
    ChassisID:    mac 00:08:ae:87:15:c0
    SysName:      asr1
    SysDescr:     ASR6126-AC, 2 1/10GE SFP+, 20 10/100/1000BaseTX
ports, 4 BaseTX/SFP combo, AC
    MgmtIP:       192.168.12.102
    Capability:   Bridge, on
    Capability:   Router, on
  Port:
    PortID:       ifname gigabitethernet3
    PortDescr:    gigabitethernet3
    TTL:          40
  VLAN:         100 vlan100

Interface:    gigabitethernet4, via: LLDP, RID: 1, Time: 0 day,
00:00:03
  Chassis:
    ChassisID:    mac 00:08:ae:87:3b:c0
    SysName:      asr2
    SysDescr:     ASR6126-AC, 2 1/10GE SFP+, 20 10/100/1000BaseTX
ports, 4 BaseTX/SFP combo, AC
```

```
    MgmtIP:       192.168.12.103
    Capability:   Bridge, on
    Capability:   Router, on
  Port:
    PortID:       ifname gigabitethernet5
    PortDescr:    gigabitethernet5
    TTL:          120
    MFS:          2360
    Port is aggregated. PortAggregID: 5
    PMD autoneg:  supported: yes, enabled: yes
      Adv:          10Base-T, HD: yes, FD: yes
      Adv:          100Base-TX, HD: yes, FD: yes
      Adv:          1000Base-T, HD: yes, FD: yes
      MAU oper type: 1000BaseTFD - Four-pair Category 5 UTP, full
duplex mode
```

Netflow

Flow Export is used to collect information about certain types of traffic propagating in the network. The information collected is widely configurable and depends on the protocol used.

The protocol supported by iBOS is Netflow v9 is described in RFC3954 and follows the IP-FIX requirements as described in RFC3917 (Requirements for IP Flow Information Export (IPFIX).

The flow exporter's main functionality is to extract the data needed from the intercepted packets. The data that is considered valuable is based on the Netflow v9 protocol and the IP-FIX requirements. Once the data is collected for a flow (see flow definition), the data is then packed and sent towards the configured collector.

Flow Contents

The data kept (and exported) for each flow is:

- Source and destination IP addresses
- IP protocol, for example TCP or UDP
- Source and destination port numbers, if protocol is TCP or UDP
- Timestamps when flow started and ended
- Number of packets in a flow
- Number of octets in a flow

Commands used for flow export are

- **destination**
- **flow-export-group**
- **max-packet-size**
- **show flow-export-group**
- **source-interface**
- **template-refresh**

Layer 2 Specific Configuration

iBOS supports Layer 2 specific configuration. This is described in the following sections:

- “Layer 2 Interfaces” on page 90
- “Forwarding” on page 103

Layer 2 Interfaces

An interface is a logical or physical device within the system that is used for packet forwarding. Physical interfaces can be tengigabitethernet, gigabitethernet, fastethernet ports and the administrative ethernet0 interface. Logical interfaces include the VLAN interface, tunnel and system loopback interfaces, downstream and aggregator interfaces.

Interfaces can be divided into Layer 2 and Layer 3 interfaces. Layer 2 interfaces operate on layer2 of the OSI model.

Physical Interfaces

Some interface types, such as fastethernet, gigabitethernet and tengigabitethernet, can operate both as Layer 2 and Layer 3 interfaces in iBOS. Using VLAN tagged packets and tagged vlan membership, the same interface can operate in both Layer 2 and Layer 3 mode simultaneously. An interface configured as a Layer 2 interface does not perform IP routing. Only the content of the packet Layer 2 header is considered when traffic is forwarded.

Layer 2 interfaces in iBOS must be connected to a Layer 3 interface in order for IP routing to occur. For tengigabitethernet, gigabitethernet and fastethernet type interfaces, this is done by assigning the interfaces to be a member of a virtual LAN (VLAN). If multiple interfaces are assigned to the same VLAN, they are also in the same broadcast domain and thereby any Layer 2 traffic between hosts connected on those interfaces is possible. This of course also means that the hosts can communicate using IP between themselves but if the router is to forward traffic to a destination in another broadcast domain, a Layer 3 interface must be created and configured.

QoS

Quality of service (QoS) is the ability to provide different priority to different data flows, and to guarantee a certain level of performance to a data flow. By either raising the priority or limiting the priority of defined flows, QoS is used to ensure the quality of services over the broadband network to the end user.

Layer 2 Logical Interfaces

VLAN Interfaces

A virtual LAN (VLAN) interface is a logical, not physical, group of interfaces defined by iBOS. VLAN configuration makes it possible to resegment the physical device network without having rearranging the physical network interfaces, and makes it possible to separate network traffic logically in the same switch by using VLAN trunking.

A VLAN interface is used to set the routers IP address on a VLAN. A VLAN interface number corresponds to the VLAN id configured on the Layer 2 interface. Any host connected on a Layer 2 interface that is a member of the VLAN may send traffic to the router and have it forwarded to a different broadcast domain.

For Waystream routers running iBOS, there are no default VLAN interfaces, but these are easily created with the **interface vlan** configuration command.

If all Layer 2 interfaces associated with a VLAN are down (i.e. no link), the VLAN interface will also be down.

Basic VLAN Configuration Example

In the following configuration example, two Waystream ASRs are connected in a small network. On ASR1, two separate VLANs have been configured on the FastEthernet ports and one GigabitEthernet port is used to connect to a GigabitEthernet port on ASR2. ASR2 has one VLAN configured but also operates local interfaces in normal mode.

Each system also has a loopback interface, which is redistributed by OSPF

Example 42 Basic VLAN Configuration Example

ASR1:

```
hostname ASR1

interface fastethernet1
 vlan member 1
 vlan untagged 1

interface fastethernet2
 vlan member 1
 vlan untagged 1

interface fastethernet3
 vlan member 1
 vlan untagged 1

interface fastethernet4
 vlan member 1
 vlan untagged 1

interface fastethernet5
 vlan member 1
 vlan untagged 1

interface fastethernet6
 vlan member 20
 vlan untagged 20

interface fastethernet7

interface fastethernet8

interface gigabitethernet1
 ip address 10.0.0.1 255.255.255.252

interface gigabitethernet2
 shutdown

interface vlan1
```

```
 ip address 192.168.1.1 255.255.255.0

interface vlan20
 ip address 172.16.1.1 255.255.255.0

interface loopback1
 ip address 192.168.10.1 255.255.255.255

router ospf
 network 192.168.10.1/32 area 1
 network 172.16.1.0/24 area 1
 network 192.168.1.0/24 area 1
 network 10.0.0.0/30 area 1
```

ASR2:

```
hostname ASR2

interface fastethernet1
 ip address 172.16.2.1 255.255.255.0

interface fastethernet2
 vlan member 20
 vlan untagged 20

interface fastethernet3

interface fastethernet4

interface fastethernet5

interface fastethernet6

interface fastethernet7

interface fastethernet8

interface gigabitethernet1
 ip address 10.0.0.2 255.255.255.252

interface gigabitethernet2
 shutdown

interface vlan20
 ip address 172.16.128.1 255.255.255.0

interface loopback0
 ip address 192.168.10.2 255.255.255.255

router ospf
 network 10.0.0.0/30 area 1
 network 172.16.2.0/24 area 1
 network 172.16.128.0/24 area 1
 network 192.168.10.2/32 area 1
```

ASR1 has vlan1 and vlan20 configured on the FastEthernet interfaces. FastEthernet interfaces 1 through 5 are members of the same VLAN and operating in untagged mode (allowing direct connection to hosts on the configured subnet). The IP of vlan1 is 192.168.1.1 which should be used by all connected hosts as default gateway if they which to reach hosts connected on ASR2.

vlan 20 is only configured on fastethernet6 with the IP address 172.16.1.1. Because the interface operates in untagged mode, the ip address could just has well been assigned directly to the fastethernet interface, removing the requirement of a vlan interface to make the router part of the broadcast domain.

The ASR1 gigabitethernet1 interface connects to the ASR2 gigabitethernet1 interface. The 10.0.0.0/30 subnet is used on the link.

On ASR2, fastethernet1 is configured with an ip address. The interface connects to the 172.16.2.0/24 subnet. Fastethernet2 is configured into vlan20 that has IP subnet 172.16.128.0/24 as interface address. vlan20 was also used on ASR1, but they are not the same broadcast domain as the connection between the two ASRs does not trunk. If the gigabitethernet interfaces had been configured with vlan member statements for vlan20, the fastethernet interfaces configured into vlan20 on both ASRs would have been part of the same broadcast domain. But since that is not the case, the vlan is local to the ASR itself. Hosts attached to vlan20 on ASR2 are not in the same broadcast domain as hosts connected to vlan20 on ASR1. Any communication between hosts on the FastEthernet ports of the two ASRs must go through one or multiple routers, and in this network that would be the ASRs.

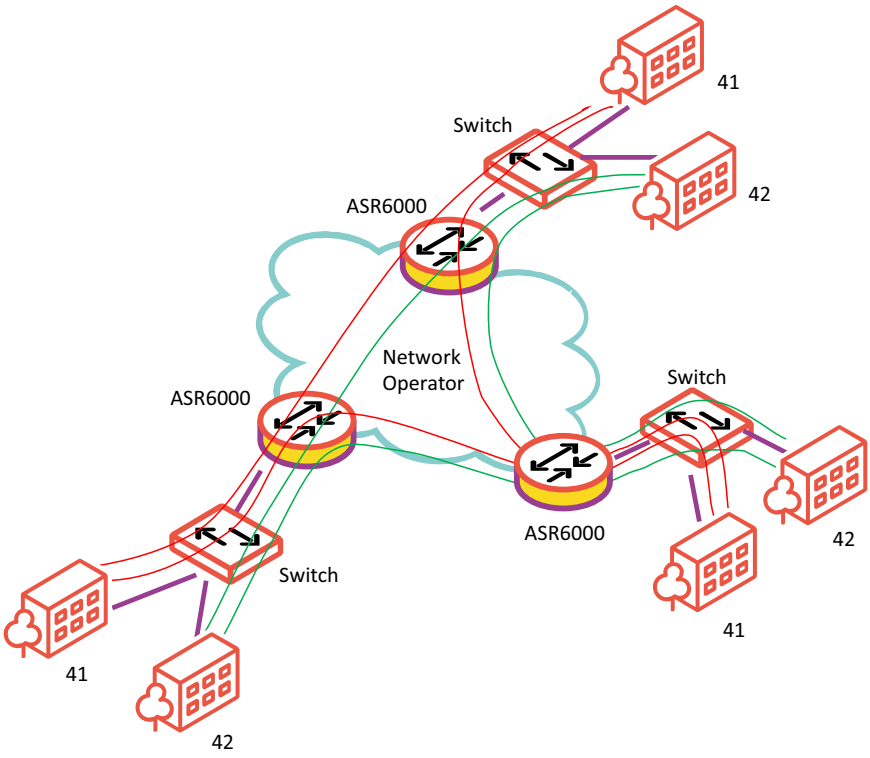
The loopback interfaces exist to make communication with the ASRs easier in case of a link failure. In a small network such as this, they might not be needed, but if the network is more complex with multiple routes between routers, loopback interfaces simplify network management and operation.

Q-in-Q

Q-in-Q (see IEEE802.1ad) was originally introduced to increase the number of VLANs. Today, the main use of the Q-in-Q functionality is to provide a Layer 2 transparency for customers that use VLANs. For example, the inner tag is set by customers, and the outer tag is set by service providers. The Q-in-Q packet traverses the carrier network and transmits the inner tag transparently, achieving a simple and practical VPN technology.

In the following figure, end users at vlan 41 sites can only communicate with end users at other vlan 41 sites. Similarly, vlan 42 end users can only communicate with vlan 42 users. There is full isolation and privacy between the different VLANs.

Figure 11 Standard VPN using Q-in-Q VLANs



Q-in-Q packets use C-vlan and the S-vlan tags:

- C-Vlan= Customer VLAN. This VLAN tag is set by the customer.
- S-vlan = Service-provider VLAN. This VLAN tag is used to encapsulate all C-vlans.

To be able to use so-called "q-in-q" tunneling, iBOS must act as if ingress port packets are always untagged (i.e, the router will always push the configured tag). When the packets are received from the customer, they are simply prepended with the S-vlan tag.

To configure Q-in-Q for a port, the following command is used:

ibos(config-if-fa)# vlan q-in-q <vid>

Note: *When* **vlan q-in-q** *is configured on a port, all effects from the* **vlan untagged** *command will be disabled.*

In addition, the interface also needs to be a member of the vid to be pushed to the packet. So, a correct configuration for Q-in-Q looks like this:

```
ibos(config)# interface fastethernet4
 ibos(config-fa4)# vlan q-in-q 42
 ibos(config-fa4)# vlan member 42
 ibos(config-fa4)# no shutdown
```

If you just want to "transit" the q-in-q tag, do not detag the packets:

```
ibos(config)# interface gigabitethernet1
ibos(config-gi1)# vlan member 40-7 !all my q-in-q S-VLANs goes here
ibos(config-gi1)# exit
ibos(config)# interface gigabitethernet2
ibos(config-gi2)# vlan member 40-70 !all my q-in-q S-VLANs goes here
```

L2TPv3 Tunnel

Layer 2 Tunnelling Protocol (L2TP) provides a dynamic tunnelling mechanism for multiple Layer 2 circuits across a packet-oriented data network. L2TP was originally defined in RFC2661 as a protocol to tunnel Point-to-point protocol (PPP) but extensions allow the protocol to be used to tunnel a number of other layer 2 protocols.

Waystream's implementation is based on *draft-ietf-l2tpext-l2tp-base-07.txt* which is the February 2003 draft for version 3 of L2TP and the information in this chapter is largely based on that draft. The L2TP feature is available on some Layer 3 iBOS devices as a mechanism to create Layer 2 connectivity across a routed network.

L2TP Terminology

The following table defines common L2TP terminology

Table 16 L2TP Terminology

Terminology	Description
Control connection	An L2TP control connection is a reliable control channel that is used to establish, maintain, and release individual L2TP sessions as well as the control connection itself. Control Messages are passed in the control connection.
Data channel	The channel for L2TP-encapsulated data traffic that passes between two LCCEs over a packet switched network. Data Messages are passed in the data channel.
L2TP Control Connection Endpoint (LCCE)	An LCCE that tunnels a circuit (either physically connected or logically connected, as via another L2TP session) to another location using L2TP, without performing any native L2 packet processing on the circuit. The LAC may tunnel to either an LNS or another LAC.
L2TP Network Server (LNS)	An LCCE that logically terminates a tunnelled circuit locally and that processes the tunnelled traffic as though the circuit was physically connected to the device. The LNS may tunnel to either an LAC or another LNS.
Peer	Peer refers to the far end of an L2TP control connection (i.e. the remote LCCE). An LAC's peer may be either an LNS or another LAC. Similarly, an LNS's peer may be either an LAC or another LNS.
Pseudowire (PW)	An emulated circuit as it traverses a packet switched network. There is one Pseudo wire per L2TP session. The Pseudo wire Type is the payload type being carried within an L2TP session (such as Ethernet, PPP and Frame Relay).
Session	An L2TP session is created by a particular L2TP control connection between two LCCEs when a circuit is successfully established. The circuit may either pass through (LAC) or terminate locally (LNS) on the LCCEs, which maintain state for the circuit. There is a one-to-one relationship between established L2TP sessions and their associated circuits.

L2TP Topology

L2TP operates between two L2TP Control Connection Endpoints (LCCEs), tunnelling circuit traffic across a packet network. An L2TP Network Server (LNS) is an LCCE that:

- decapsulates tunnelled traffic
- removes any Layer 2 data which may be present
- processes the encapsulated Layer 3 payload without forwarding to other hardware (i.e. routing an IP packet within an HDLC frame tunnelled by L2TP)

An L2TP Access Concentrator (LAC) is an LCCE that processes tunnelled traffic based primarily on Layer 2 information, either by directly switching session traffic to an attached circuit or to another L2TP session, or by handing off to a virtual bridging

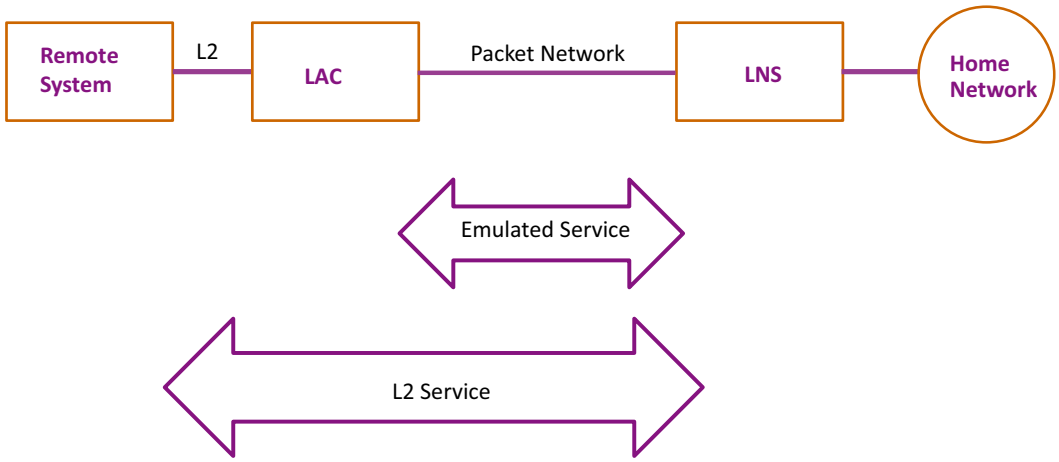
function. A single piece of network equipment may serve as both an LAC and LNS.

There are three predominant tunnelling models in which L2TP operates:

- LAC-LNS (or vice versa)
- LAC-LAC
- LNS-LNS

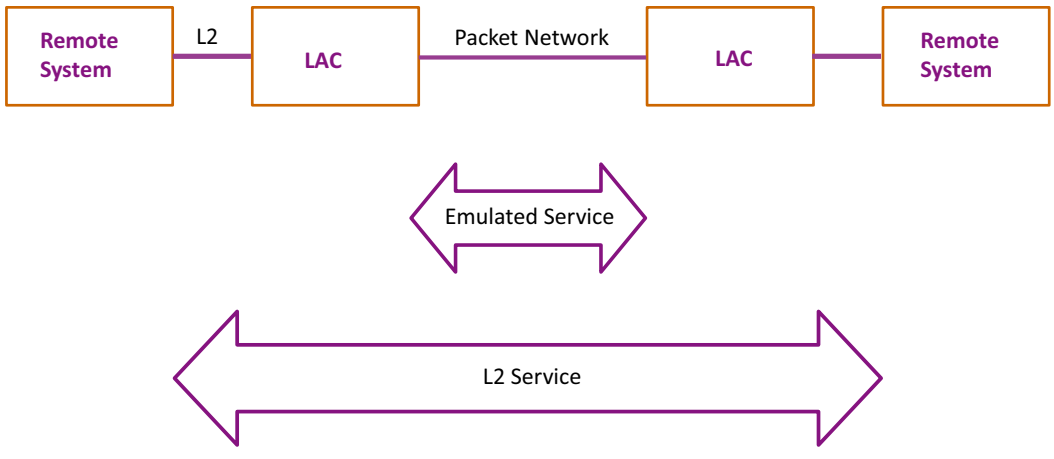
These models are diagrammed in the examples below. (Dashed lines designate network connections and solid lines designate circuit connections.)

Example 43 LAC-LNS Reference Model



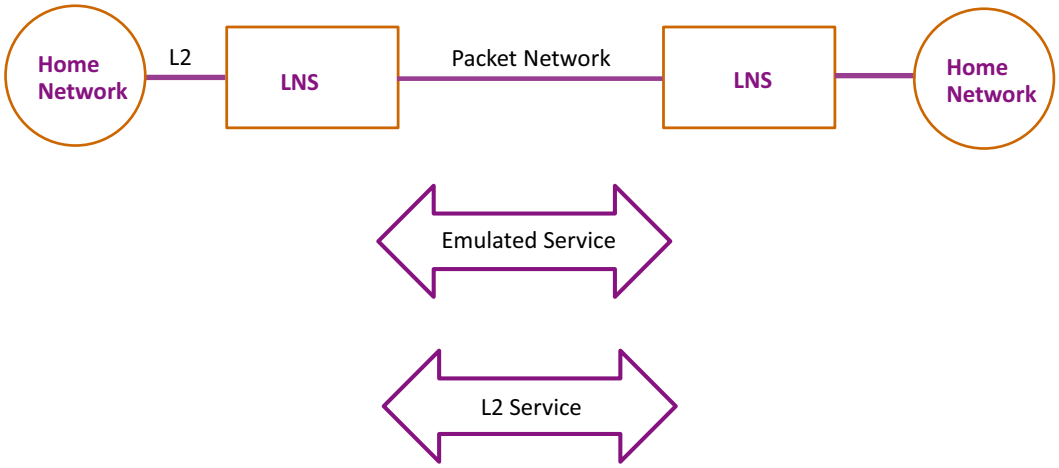
In the LAC-LNS, the LAC receives traffic from an L2 circuit on one side, which it then forwards via L2TP across an IP or other packet-based network. On the other side, an LNS logically terminates the L2 circuit locally and routes traffic (at Layer 3) to the home network. The action of session establishment is driven by the LAC (as an incoming call) or the LNS (as an outgoing call).

Example 44 LAC-LAC Reference Model



In the LAC-LAC model, both LCCEs are LACs. Each LAC forwards circuit traffic from the remote system to the peer LAC using L2TP, and vice versa. In its simplest form, an LAC acts as a simple cross-connect between a circuit to a remote system and an L2TP session. This model typically involves symmetric establishment; that is, either side of the connection may initiate a session at any time (or simultaneously, in which a tie-breaking mechanism is utilised).

Example 45 LNS-LNS Reference Model



The LAC-LNS model has two LNS as the LCCEs. A user-level, traffic-generated, or signalled event typically drives session establishment from one side of the tunnel.

Note: *In L2TPv2, user-driven tunnelling of this type is often referred to as "voluntary tunnelling". Further, an LNS acting as part of a software package on a host is sometimes referred to as an "LAC Client".*

L2TP Protocol Overview

L2TP utilises two types of messages, control messages and data messages (sometimes referred to as "control packets" and "data packets", respectively). Control messages are used in the establishment, maintenance, and clearing of control connections and sessions. These messages utilise a reliable control connection within L2TP to guarantee delivery. Data messages are used to encapsulate the L2 traffic being carried over the L2TP session. Unlike control messages, data messages are not retransmitted when packet loss occurs.

Example 46 L2TPv3 Structure

```
+-------------------+    +-----------------------+
| Tunneled Frame    |    | L2TP Control Message  |
+-------------------+    +-----------------------+
| L2TP Data Header  |    | L2TP Control Header   |
+-------------------+    +-----------------------+
| L2TP Data Channel |    | L2TP Control Channel  |
| (unreliable)      |    | (reliable)            |
+-------------------+----+-----------------------+
| Packet-Switched Network (IP, FR, MPLS, etc.)   |
+------------------------------------------------+
```

The example depicts the relationship of control messages and data messages over the L2TP control and data channels, respectively. Data messages are passed over an unreliable data channel, encapsulated by an L2TP header, and sent over a Packet-Switched Network (PSN) such as IP, UDP, Frame Relay, ATM, MPLS, etc. Control messages are sent over a reliable L2TP control connection, which operates over the same PSN.

The necessary setup for tunnelling a session with L2TP consists of two steps: (1) Establishing the control connection, if required, and (2) establishing a session as triggered by an incoming call or outgoing call. An L2TP session must be established before L2TP can begin to forward session frames. Multiple sessions may be bound to a single control connection, and multiple control connections may exist between the same two LCCEs.

L2TP Control Connection Management

The L2TP Control Connection handles dynamic establishment, tear down, and maintenance of the L2TP sessions and of the control connection itself.

This section describes the typical control connection establishment and tear down exchanges. It is important to note that, in the diagrams that follow, the reliable control message delivery mechanism exists independently of the L2TP state machine. For instance, explicit acknowledgement packets (ZLB ACKs) may be sent after any of the control messages indicated in the exchanges below if an acknowledgement is not piggybacked on a later control message.

LCCEs are identified during control connection establishment either by the Host Name AVP, the Router ID AVP, or a combination of the two. The identity of a peer LCCE is central to selecting the proper configuration parameters (i.e. Hello interval, window size, etc.) for a control connection, determination of how to setup associated sessions within the control connection, password lookup for control connection authentication, and control connection level tie-breaking.

Establishing the control connection involves an exchange of AVPs that identifies the peer and its capabilities. A three-message exchange is used to establish the control connection. The following is a typical message exchange:

Example 47 Control Connection Establishment

```
     LCCE A      LCCE B
     ------      ------
     SCCRQ ->
                 <- SCCRP
     SCCCN ->
```

Control connection tear down may be initiated by either LCCE and is accomplished by sending a single StopCCN control message. As part of the reliable control message delivery mechanism, the recipient of a StopCCN MUST send a ZLB ACK to acknowledge receipt of the message. The recommended time for a full retransmission cycle is at least 31 seconds. The following is an example of a typical control message exchange:

Example 48 Control Connection Tear Down

```
LCCE A      LCCE B
------      ------
StopCCN ->
(Clean up)

            (Wait)
            (Clean up)
```

An implementation may shut down an entire control connection and all sessions associated with the control connection by sending the StopCCN. Thus, it is not necessary to clear each session individually when tearing down the whole control connection.

L2TP Session Management

After successful control connection establishment, individual sessions may be created. Each session corresponds to a single data stream between the two LCCEs. This section describes the typical call establishment and tear down exchanges.

A three-message exchange is used to establish the session. The following is a typical sequence of events:

Example 49 Establish L2TP Session for an Incoming Call

```
LCCE A      LCCE B
------      ------
(Call
 Detected)

ICRQ ->
           <- ICRP
(Call
 Accepted)

ICCN ->
```

A three-message exchange is used to set up the session. The following is a typical sequence of events:

Example 50 Establish L2TP Session for an Outgoing Call

```
LCCE A      LCCE B
------      ------
           <- OCRQ
OCRP ->

(Perform
 Call
 Operation)

OCCN ->

(Call Operation
 Completed
 Successfully)
```

L2TP Session Tear Down

Session tear down may be initiated by either the LAC or LNS and is accomplished by sending a CDN control message. After the last session is cleared, the control connection may be torn down as well (and typically is). The following is an example of a typical control message exchange:

Example 51 L2TP Session Tear Down

```
LCCE A      LCCE B
------      ------
CDN ->
(Clean up)
             (Clean up)
```

L2TP Over Specific Packet-Switched Networks (PSN)

If necessary, L2TP may operate over a variety of Packet Switched Networks (PSN). The L2TP Session Header encapsulation may vary for a given PSN.

There are two modes described for operation over IPv4, L2TP over IP and L2TP over UDP. Waystream's implementation support L2TP over IP.

L2TPv3 over IP utilises the IANA assigned IP protocol ID 115.

Configuring L2TP

Waystream's implementation of L2TP version 3 is designed to allow one or more interfaces in the system to be bridged over an L2TP version 3 tunnel to a remote destination. This effectively allows any Ethernet-based traffic to pass through the network without intermixing with other IP traffic.

Create the L2TP Control Connection

The control connection must be established between the LAC/LNS involved in the communication.

To create the control connection, the remote LAC/LSN IP address must be known.

Example 52 Create L2TP Control Connection

```
ibos# configure terminal
ibos(config)# l2tpv3 192.121.165.34 source-interface loopback0
```

The router periodically sends a hello packet for which it expects an acknowledgement from the peer. If no hello packet is received, the control connection and any related tunnels goes down. The hello packet interval can be controlled by the **l2tpv3 hello-interval** configuration command.

If a packet is sent on the control connection and no acknowledgement is received, the router will attempt to retransmit the packet after an interval that can be configured with the **l2tpv3 retry** configuration command. A maximum **timeout** can also be set that control how long a packet may remain unacknowledged in the transmit queue. If the packet is not successfully acknowledged by the remote LAC/LNS, the control connection is reset and reinitialised (bringing down any tunnels related to the control connection).

Create the L2TP Tunnel

A tunnel interface is created and bound to the interface to be bridged over the tunnel.

There may be multiple tunnels between the same two LAC/LNS routers. Each tunnel must be identified by a virtual circuit id.

Note: *Some implementations are limited in how the virtual circuit id is set. Be sure to set the same virtual circuit id on both endpoints of the tunnel.*

Example 53 Create the L2TP Tunnel

```
ibos# configure terminal
ibos(config)# interface tunnel1
ibos(config-if-tun)# l2tpv3 192.121.165.34 BTUN157
ibos(config-if-tun)# port forward fastethernet1
ibos(config-if-tun)# tunnel mode l2tpv3
ibos(config-if-tun)# no shut
ibos(config-if-tun)# interface fastethernet1
ibos(config-if-fa)# port forward tunnel1
```

In the example, fastethernet1 is bridged over tunnel interface 1 to the remote LAC/LNS 192.121.165.34. The virtual circuit identifier is set to BTUN157, and must be the same on both LAC/LNS involved in the communication.

The traffic sent through and received from the tunnel can be rate-limited with the **port egress-rate** configuration command.

It is also possible to set the IP header don't fragment bit or DSCP values for the tunnel packets.

L2TP Fragmentation

Ethernet up to 100 Mbit/s has a maximum transmission unit (MTU) of 1500 bytes. Normally hosts try to send as large packets as possible, which means that an ethernet connected host may attempt to send 1500 byte packets to a remote destination. Tunnel encapsulation for bridging and VPN purposes add additional headers to the packet to be bridged. The resulting tunnel packet is larger than the original packet.

If a packet to be bridged is 1500 bytes, the resulting encapsulated tunnel packet will be larger than the MTU allowed on a typical ethernet interface. This may require the router to split packets into multiple fragments before each fragment is encapsulated into a tunnel packet. The receiving LAC/LNS must reassemble the packet before it is forwarded.

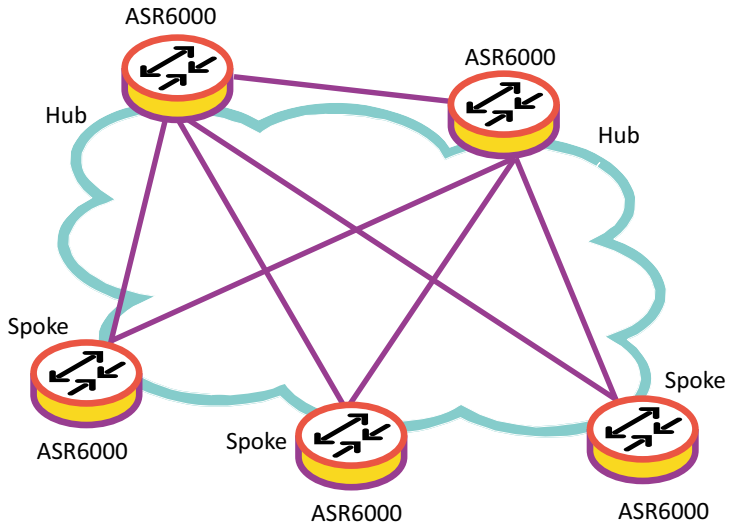
This fragmentation/reassemply can be CPU intense. iBOS include a feature to allow the router to intercept and respond to TCP/IP packets that will require fragmentation of the resulting tunnel packet. The feature can be enabled with the **tunnel ip must-frag** configuration command in tunnel interface context.

For UDP/IP traffic and for other network protocols (IPX, AppleTalk, NetBIOS) there is no similar feature like TCP path MTU discovery. It may be required by the network administrator to configure hosts not to send packets with the maximum MTU in order to maximise the tunnel encapsulation performance of the LAC/LNS routers.

L2TP Redundancy

In tunnels in Layer 2 networks, redundancy is provided using extra hub devices and routing protocols to ensure that the path between two nodes is always up. The redundant hub can be used in a hub-spoke topology, although this more than doubles the number of tunnels required for 2 hubs. The number of tunnels required with multiple hubs will be (N - 1) + 1, where N is the number of hubs used.

Figure 12 Redundancy in hub-spoke Topology



In L2 VPNs, this redundancy ensures that a single link failure in the distribution network can not cause loss of service. In the event of a link failure, RSTP is used to propagate the change to the topology, and this results in a fast switch of the forwarding paths affected by the link failure.

Note: *A point-to-point link between the hubs is also required to minimise hops between sites.*

For efficient use of L2TP redundancy, the command **l2tpv3 passive** is implemented.

Link Aggregation

Link aggregation in accordance with IEEE 802.3ad (including LACP) is implemented in this release.

IEEE 802.3ad is a link aggregation standard whereby several Ethernet ports are aggregated into a single virtual port, which provides greater bandwidth and failure protection. When IEEE 802.3ad support is configured, the router does not need manual configuration of the link aggregation groups because the link aggregation groups within an aggregator interface are created dynamically from the media settings on the ports.

LACP allows a switch to negotiate which link aggregation group to use on an aggregator interface by sending LACP packets to the peer.

When selecting an active link aggregation group (LAG) within an aggregator interface, bandwidth of the links within the LAG is used as a priority. A LAG with a partner that runs LACP will always be preferred over LAGs with no LACP neighbours.

Link aggregation is configured in iBOS with the following commands:

- **port aggregator join**
- **show aggregator**
- **debug aggregator**

Forwarding

Forwarding is the process in which iBOS receives a packet, looks up the packet destination in the hardware or kernel forwarding table, determines any actions to take on the packet in terms of header rewrites, queueing, priority, determining the egress interface and then transmitting the packet out the egress interface.

iBOS support for L2 forwarding is described in:

- "Layer 2 Unicast Support" on page 103
- "UNI Isolated Ports/VLANs" on page 103
- "Layer 2 Multicast Support" on page 104
- "Spanning Tree" on page 106

Layer 2 Unicast Support

The most common type of IP traffic today is unicast, i.e. one-to-one communication between a unique set of IP addresses.

UNI Isolated Ports/VLANs

The purpose of UNI (User Network Interface) isolated ports is to separate client interfaces in the same VLAN on the Layer 2. A UNI port is only able to talk to an NNI (Network Node Interface) port. The port type UNI/NNI configuration is configured for ranges of VLANs, just like the VLAN member command.

When port type UNI is set on an interface, the interface will enable UNI to UNI interface isolation on layer 2. Frames received on one UNI port will not be forwarded to another UNI port even if the two ports are members of the same VLAN.

NNIs on the other hand may communicate freely with UNIs and NNIs.

Note: *The DHCP snooping functionality is affected by these new concepts, since the snooper does filtering of DHCP offers. The forwarding of STP BPDU frames is also affected by the UNI/NNI settings of the interfaces participating in a VLAN.*

The port type UNI configuration is configured for ranges of VLANs, just like the **vlan member** command.

The UNI L2 separation is configured per interface and VLAN using the following iBOS commands:

- **port type uni**
- **port type nni**
- **show interface**

UNI versus No Port Uplink

The command **port type uni** enables UNI port isolation restrictions. If both port uplink and UNI has been configured on an interface the UNI setting will take precedence for the VLANs configured to be UNIs in client manager. The UNI L2 separation is configured per interface and VLAN using the port type uni command

NNI versus Port Uplink

An uplink interface will most likely also be an NNI, but an NNI may very well not be configured as port uplink.

One big difference is that the NNI setting does not imply sending bootp packets for establishing the connection with BECS on the interface. Therfore, the NNI can be facing an external DHCP server without bothering it with bootp requests from the ASR.

If **port uplink** is not configured on the interface, flows are created for the NNI interfaces.

MAC Forced Forwarding

In every broadband network, customer separation is essential. Customers must not be allowed to talk to each other in an uncontrolled manner. In an L2 topology, this can be achieved using a number of techniques. The UNI-NNI feature of iBOS provides Layer2 separation within a single network device (e.g. port 1 cannot talk directly with port 3).

Transit traffic, for instance in a ring of ASRs, is not subject to these UNI-NNI rules and an additional means of Layer2 separation is required. To this end, MAC address Forced Forwarding in accordancewith RFC4562 has been implemented in iBOS.

MACFF ensures Layer2 separation by filtering customer ingress traffic based on a set of known MAC addresses - i.e. the gateway and servers of the service provider.

MACFF is to be used together with the UNI-NNI feature.

Layer 2 Multicast Support

In a triple play broadband network, it is essential that multicast services can be differentiated from each other, and that they work in a robust manner. IP-TV broadcasts must be reliable and only reach those customers that are paying for the service. In addition, the network should not be populated with channels that nobody is watching.

To configure Layer 2 networks for multicasts, IGMP is used to filter channels to specific customers. To do this, the following tasks need to be done:

1. You need to ensure that the client can not source multicast into the network by themselves . This is handled with the **port type uni vlan** command which prevents unsolicited IPv4 multicast injection, for example:

```
interface gigabitethernet1
  vlan member 101
  vlan untagged 101
  port type uni vlan 101
  no shutdown
```

2. Next, you must enable IGMP snooping on the VLAN interface to start the IGMP snooper with the **ip igmp snooping** command so that iBOS is fully configured for IGMP, for example:

```
interface vlan101
  ip igmp snooping
  no shutdown
```

At this point, the L2 multicast configuration provides no restriction for what channels the client can view. To do this, you will also need to apply some ACLs.

Access Control of Multicast Groups

Using an access-list, the command **port ip igmp vlan access-group** allows you to control which multicast groups that can be joined on a particular interface. An example configuration for the gigabitethernet1 interface would be:

```
interface gigabitethernet1
  vlan member 101
  vlan untagged 101
  port type uni vlan 101
  port ip igmp vlan 101 access-group tv_basic
  no shutdown
  !end

access-list tv_basic
  permit ip any 238.1.2.3/32
  !end
```

The access list “tv_basic” only allows access to the 238.1.2.3 group.

Note: *The command “ip igmp access-group <access-group>” command is also available and can be used for access control if “port ip igmp... ”command is not used.*

To implement IGMP immediate leave for access list “tv_basic”, you must extend the configuration with the following:

```
interface gigabitethernet1
  vlan member 101
  vlan untagged 101
  port type uni vlan 101
  port ip igmp vlan 101 access-group tv_basic
  port ip igmp vlan 101 immediate-leave group-list tv_basic
  no shutdown
  !end
```

Note: *If the iBOS devices are connected in a chain where traffic transits multiple devices, all devices in the chain should have “ip igmp snooping” configured to avoid flooding of multicast traffic.*

Spanning Tree

The Spanning Tree Protocol (STP) is a link management protocol based on the IEEE 802.1D standard for bridge protocols. STP is used to manage the information exchange between Layer2 network devices (bridges) to build a tree structure from a mesh of network devices (for example, switches).

The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them. Spanning tree also allows a network design to include redundant links that provide automatic backup paths if an active link fails, eliminating the risk of bridge loops or the need to manually enable/disable of these backup links.

Rapid Spanning Tree (RSTP) is based on IEEE 802.1w and was introduced in 1998 to improve and replace spanning-tree. The main difference is shorter convergence times for topology changes but the basic algorithm is the same as STP.

Bridge Roles

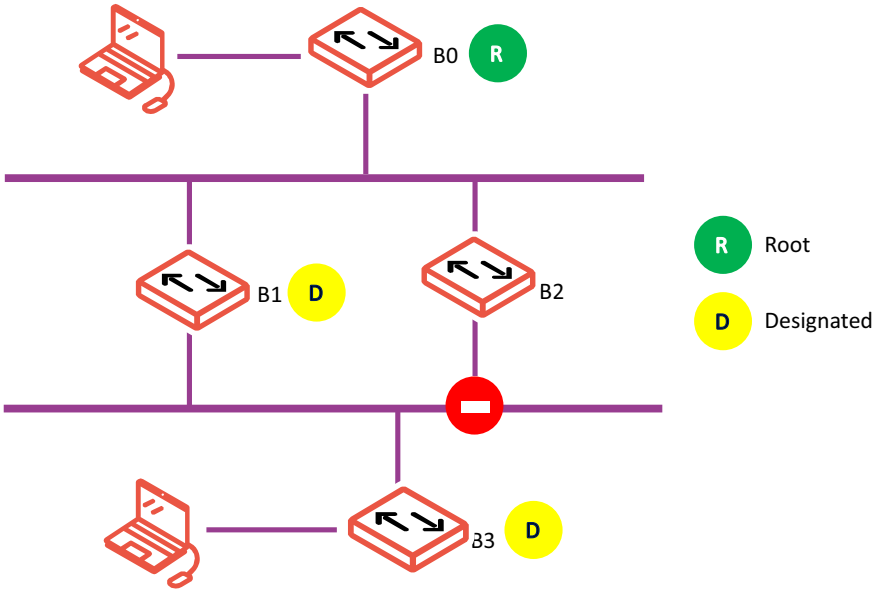
The bridges in a spanning tree are classified in two roles:

- Root Bridge
- Designated Bridge

The **Root Bridge** is the bridge that has been elected to become root, based on the election mechanism described below. The root bridge has the role of sending the BPDUs for the whole tree, making the STP use less BPDUs than it would if all bridges sent BPDUs on their links.

On a LAN segment, a bridge is elected to become the **Designated Bridge** for the segment. The election of designated bridge is based mainly on the lowest path cost. If there are paths with equal cost, the bridge ID is the tie breaker.

Figure 13 Bridge Roles



Election Mechanism

The STP operation first starts by electing the root bridge. The election process is done as follows:

- All bridges send a configuration message called configuration BPDU or CBPDU. The configuration message contains:

Figure 14 STP Message

- All bridges initially assume that they are root and cost is set to 0.
- The bridge will continue to receive BPDUs from all its ports and it will then store the best message on each port.
- A decision is made on which port contains the best BPDU including the bridges own BPDU. the decision is based on the following:
  - Lowest root ID (priority + MAC address)
  - If root ID is equal then the one with the lowest cost. The cost here is the sum of the cost to reach root received from the BPDU on that port, and the cost on the port to reach the announcing bridge.
  - If root ID and cost are equal then the decision will be based on the transmitters ID
  - If all the above conditions are equal (only occurs if there is a loop in the network), the lowest port ID will be the tie breaker.
- Once the best BPDU is chosen, the bridge starts to transmit the BPDU on all ports.
- If the received BPDU on a port is better than the one chosen to be transmitted the bridge will not send a BPDU on that port, this port is called the root-port.
- If the received BPDU on a port is worse than own BPDU, the port becomes designated for that bridge.

STP Ports and States

The ports in a spanning tree are set to states corresponding to their function in the tree.

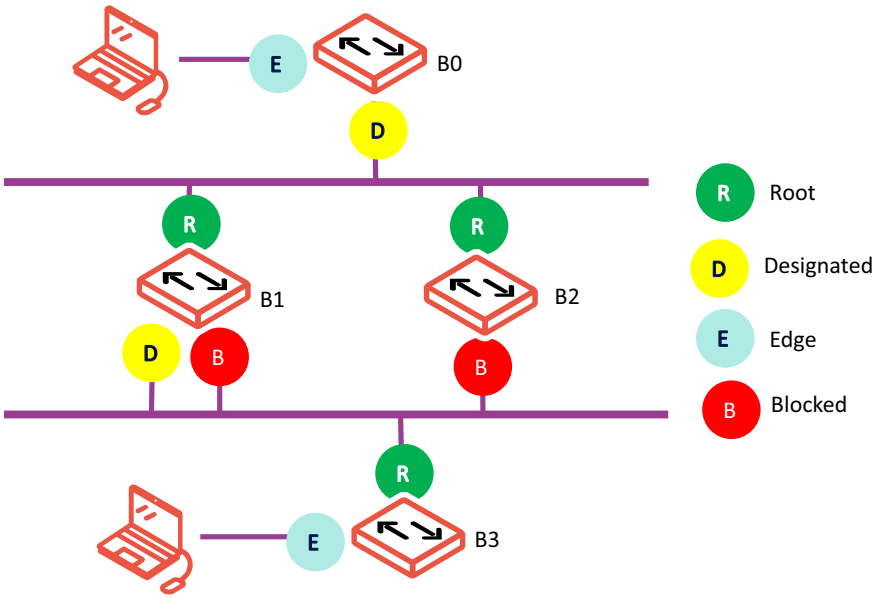
The states that a port might go through are the following:

- Listening: This is the first state once a port is activated, during this state the port will receive and process BPDUs, but it will not learn and build the forwarding table. This state is a sort of delay to allow the tree to stabilize.
- Learning: This state occurs after the listening state, during this state the bridge will learn about the tree by populating the bridge table, based on the configuration messages received.
- Forwarding: This state occurs after the listening state, once the algorithm decides that the port should be a forwarding port. During the forwarding state the packets are forwarded from and to the port.
- Blocking: This state occurs after the listening state, once the algorithm decides that the port should be a blocking port. During the blocking state all traffic is blocked from and to the port except the BPDUs, they are allowed in case of a topology change.

The port modes are the following:

- Root port: is the port closes to the root bridge. The state here is set to Forwarding meaning that traffic is forwarded from and to bridge.
- Designated port: is the port where the bridge acts as a designated bridge for the bridges connected to that port. The state on the Designated port is set to Forwarding.
- Blocked port: is the port that is not part of the tree, the port becomes blocked once it receives a better BPDU than its own but not better than the root BPDU. In case two or more ports on a bridge are connected to the same segment the port with the higher cost or port-ID is set to blocking. Blocked meaning that it will still listen to BPDUs but will not forward traffic from both directions and will not learn MAC addresses of stations.

Figure 15 Port Roles



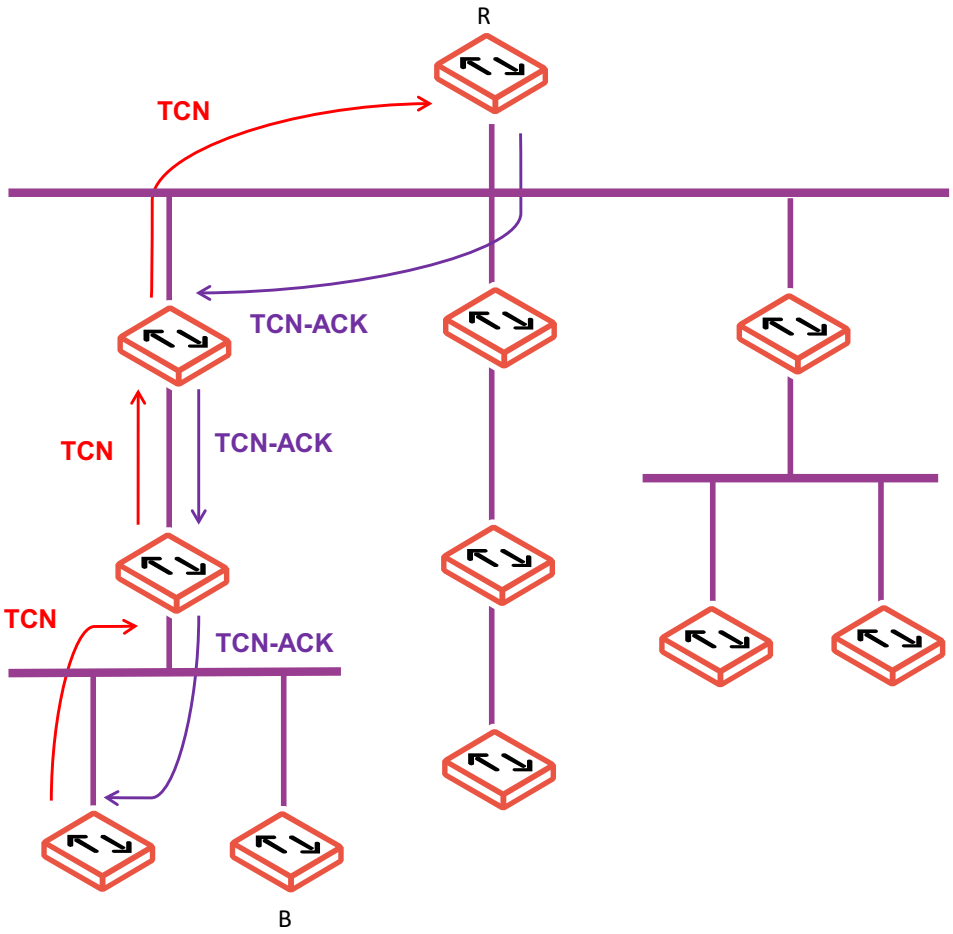
STP Topology Changes

The network mesh is changed often due to either a network device failure or introducing/removing a network device. These changes are handled by the spanning tree algorithm with the help of BPDU messages and timers.

Topology changes might occur in the network where STP enabled switched resides, the Spanning tree algorithm deals with such changes by sending topology change messages, Topology Change Notification (TCN) and Topology Change Acknowledgments (TCN-ACK).

The TCN is used to notify other switches about a change in the topology, for example when a link fails or is added in the network. The TCN is sent out by the switch that detects the change towards the root bridge. The TCN BPDU is sent until the designated bridge on the segment sends a TCN-ACK in reply (which is a CBPDU with the topology change acknowledgment bit set). The designated bridge then sends a TCN of its own on its root port.

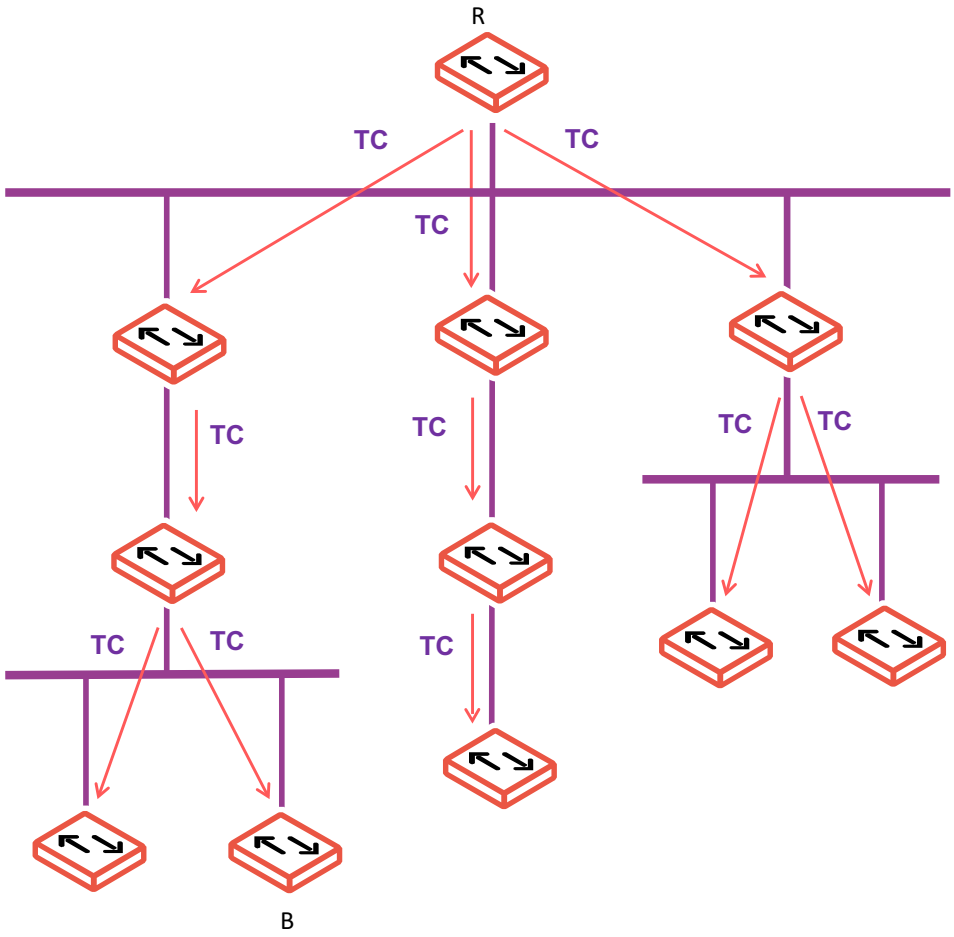
Figure 16 TCN Messages in the Spanning Tree Network



Bridge B notifies a topology change by sending a TCN on its root port. The TCN is acknowledged with TCN-ACK and the TCN is forwarded up to the root bridge R.

When the TCN has propagated to the root bridge, it will set a topology change bit set in its CBPDU. This TC-CBPDU is relayed by all switches in the network, thus informing all switches in the network that a change has occurred and that the current forwarding table entries should be flushed and recalculated.

Figure 17 Topology Change Bit Relayed in the Spanning Tree Network



The root bridge **R** sets the TC bit in the bpdu, This bpdu is then relayed to the whole network.

The TC-CBPDU is set by the root bridge for a period of max_age + forward delay seconds, which is default (20-15) = 35 seconds.

Whenever a host disappears or appears on the network (including when a connected host is powered up), a topology change occurs. In large networks, there can be a lot of topology changes (particularly if the topology is flat). This situation has been improved with rapid spanning tree where a port can go immediately to forwarding if it is known to connect to a host (also called the edge port).

RSTP

Enhancements were made in RSTP to make it converge faster than STP. These enhancements resulted in changes to port states and roles.

RSTP Port States

RSTP has eliminated states Disable, Blocking and Listening, and has replaced them with one common state, **Discarding**.

The states in RSTP are

- **Discarding**; the discarding state is a combination of spanning-trees Disabled, Blocking, and Listening states. The port is not part of the spanning-tree topology and no MAC learning occurs.
- **Learning**; the switch does not forward frames, it is not yet certain if this port goes to another branch in the tree. It does learn MAC addresses seen on the port and thereby builds the forwarding table. The Forward delay timer determines the time it takes for the port to go from discarding to forwarding state.
- **Forwarding**; the port forwards frames. It also continuously learns about MAC-addresses for the forwarding table. This port switches to this state from Listening state after a duration of “Forward delay timer”.

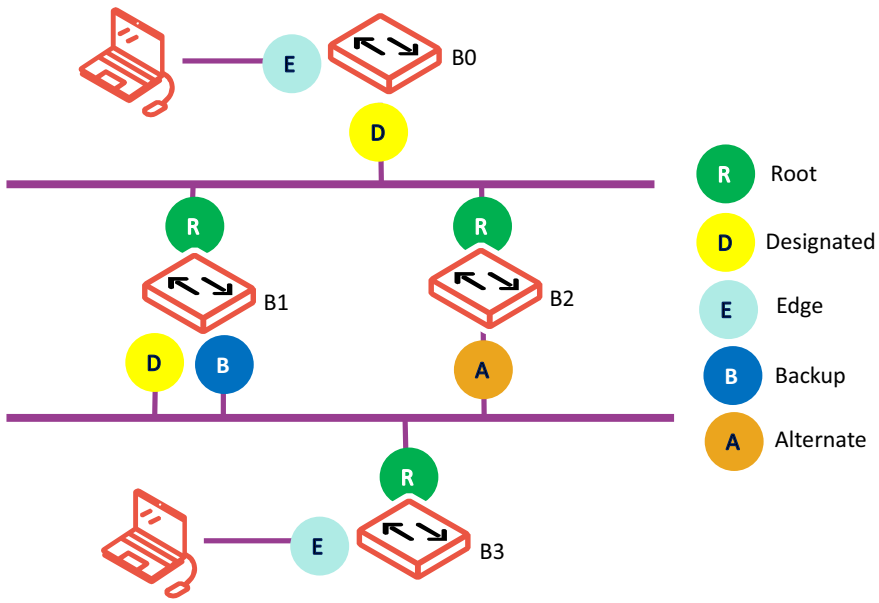
RSTP Port Roles

More importantly, RSTP introduces the concept of port roles. In STP, a port can be either a **root port** (lowest cost in the switch towards the root bridge) or a **designated port** (lowest cost on the segment towards the root bridge).

RSTP also introduces the **alternate** and **backup** port roles. Similar to the STP blocked port, RSTP backup port is a possible designated port candidate, but the port has a higher cost than the selected designated port or i higher port-ID and is therefore set to backup state to avoid loops. An alternate port is similar to the backup port except that it can not reside on the same bridge. A port is set to alternate if it receives a better BPDU than its own but worse than the root BPDU.

RSTP also introduced a new port role, the **edge** port. An edge port is immediately put into forwarding state. Therefore such ports should only be configured when it is certain that they are not connected to another branch of the tree. Typically ports that connect to end hosts can be set as edge ports.

Figure 18 RSTP Port Roles



RSTP Enhancements

RSTP introduces new configuration and calculation parameters. The most important enhancements compared to STP are:

- Edge ports
- Failover time
- Convergence

Edge Ports

RSTP also introduced a new port role, the edge port. An edge port is immediately put into forwarding state. Therefore, edge ports should only be configured when it is certain that they are not connected to another branch of the tree. Typically, ports that connect to end hosts are set as edge ports.

Failover Time

The failover time has been dramatically reduced with RSTP. With STP, it can take minutes for the network to converge into a larger topology. With RSTP, the converance times are much shorter, This is partially due to root switch failure detection in 3 hello times (default 6 seconds), and partially due to a mechanism where a switch can suggest to the designated port it has a better path towards the root bridge and thereby trigger a rapid transition to forwarding state.

Instead of waiting for the network to converge before putting ports into forwarding state, RSTP can confirm through a feedback mechanism if a port immediately can move into forwarding state thereby reducing failover time.

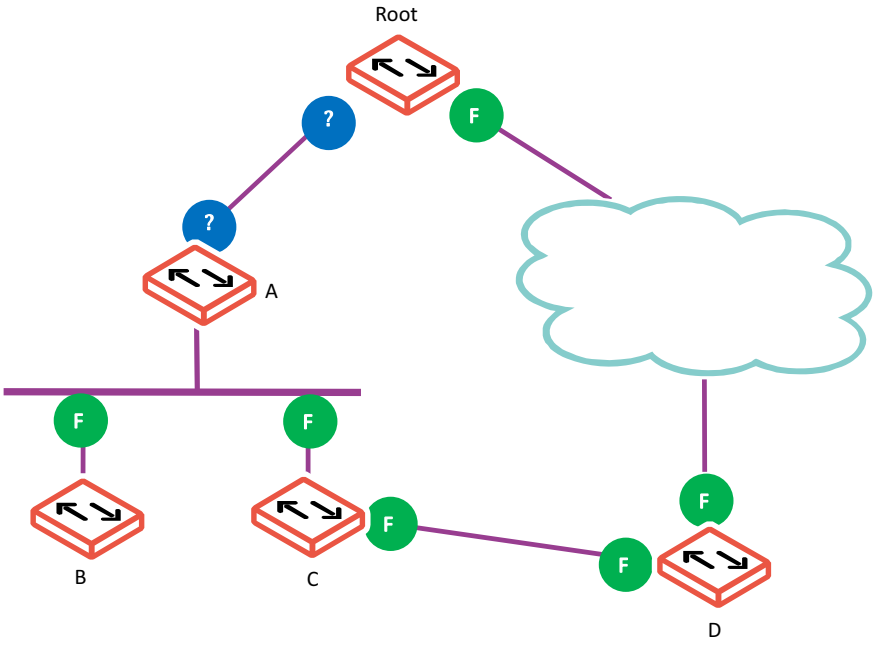
Convergance

As previously stated, one important advantage of RSTP over STP is the way it converges new links in the bridged topology. The following two examples demonstrate convergance in spanning tree and rapid spanning tree topologies, respectively.

Example 54 STP Convergance

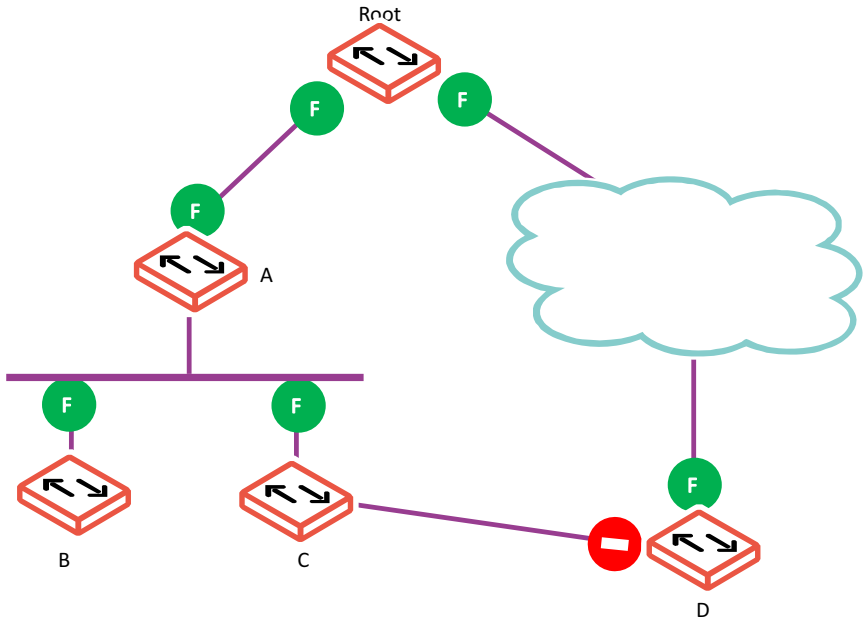
- Switch A is already connected to the root bridge through C and D. When a link is added between the A switch and the root bridge in this topology, the ports go into listening state.

Figure 19 Adding New Links with STP



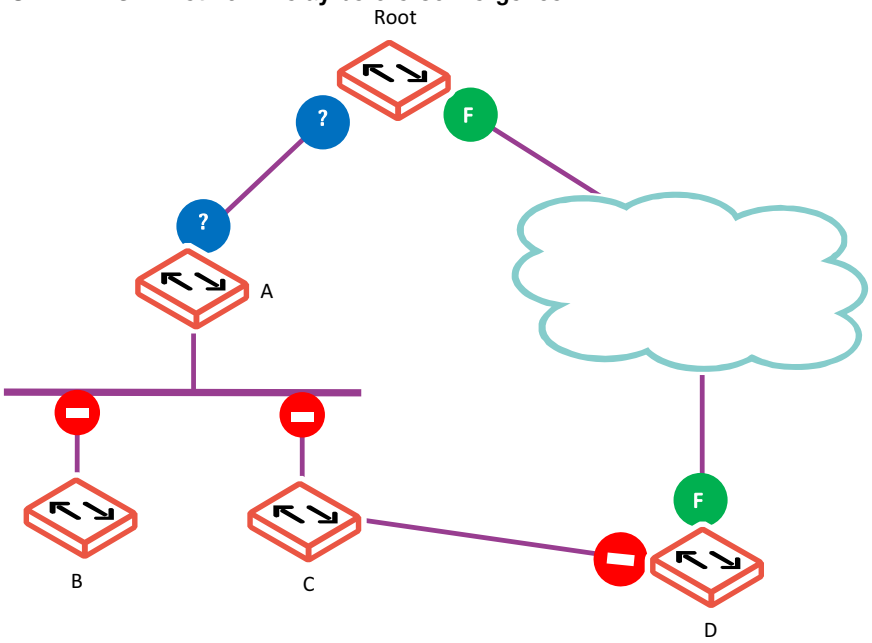
- When A sees the BPDU from the root switch, it will send its own BPDUs towards B and C. C in turn will do the same thing towards D.
- D will now have two possible paths towards the root bridge. D will therefore block the port towards C to avoid the loop.

Figure 20 STP Converges Network



Unfortunately, the timers involved in transition between various states means that it will take up to 30 seconds before the link between A and root bridge goes into forwarding state, but the connection from C to D is already blocked so no traffic can go through there. The result is that the addition of the new link cuts off bridges A, B and C from the network until the link to the root bridge enters the forwarding state.

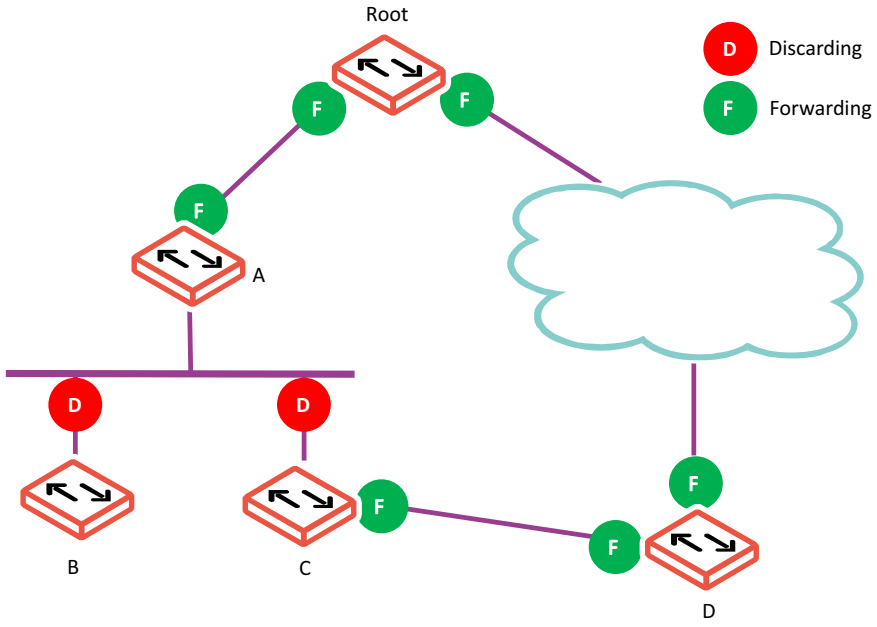
Figure 21 STP Network Delay before Convergence



Example 55 RSTP Convergance Example

With rapid spanning tree, when the link between A and the root bridge is connected, the ports enter discarding state, but when the root bridge BPDU reaches A on the new link, it will immediately put the ports to B and C into discarding state and authorize the root bridge to set its port towards A into forwarding state. A will also put its own port towards the root bridge into forwarding state. The topology now looks as follows;

Figure 22 Adding a Link to RSTP

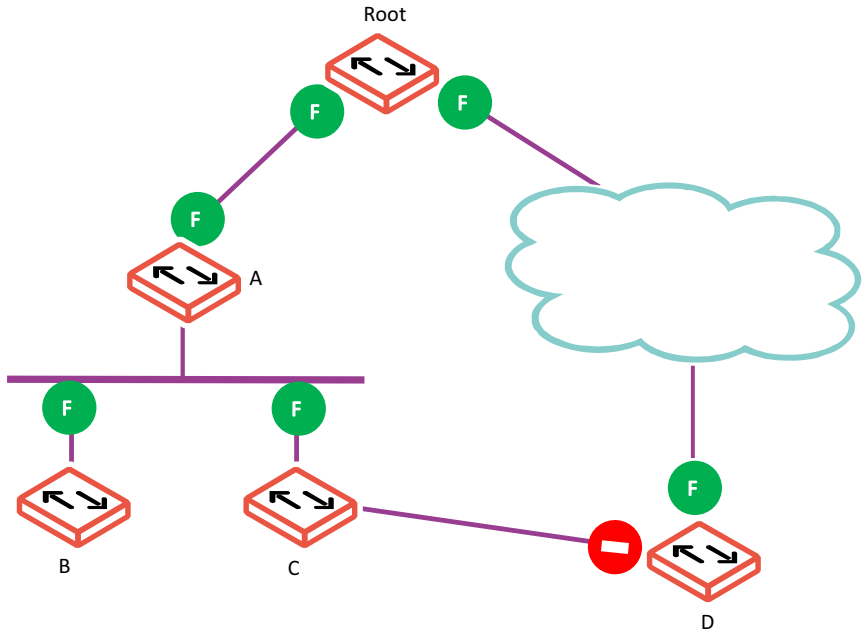


There is no loop in the network, but B is cut of from the network at this point.

However, this mechanism continues as BPDUs from A is sent to B and C. So in the negotiation between A and B, the link will rapidly be put into forwarding state. Since B only has edge ports it has no ports to put into discarding state and B can therefore immediately authorize A to put its port towards B into forwarding state.

For C, it has a link towards D, but since it has a higher cost towards the root bridge than the new BPDUs from A indicate, that port is also put into discarding state as C then orders A to put its port towards C into forwarding state.

Figure 23 Network Converges



Now at this point, D has a better path towards the root bridge over its current root port and therefore the changes to the topology stops here, but the resulting topology is the same as in the spanning-tree example, only that convergance has occurred much quicker through the acknowledge mechanism in RSTP where a switch can authorize its designated switch on the root port to immediately go into forwarding state.

Note: *This mechanism only works on point-to-point connections and benefits the most where edge port configuration can be made to exclude ports from consideration during this type of topology change.*

Multiple Spanning Tree (MSTP)

STP and RSTP in bridged networks is replaced with MSTP as defined in IEEE 802.1S. As well as supporting the features in STP/RTP, MSTP has significant advantages over STP and RSTP:

- MSTP introduces the REGION concept. A switched network is divided into multiple regions, and each region contains multiple spanning trees that are independent of one another. MSTP uses a common and internal spanning tree instance (CIST) to exchange information between regions, and this eliminates loops in the network
- MSTP introduces the INSTANCE concept. In MSTP, instances are independent spanning trees where each spanning tree corresponds to a group of VLANs. By mapping multiple VLANs to a spanning tree instance, transmission overheads and network resources are reduced. Load balancing can also achieved using multiple instances
- MSTP is fully backwards compatible with STP and RSTP.
- MSTP is mainly used to manage VLANs in Layer 2 networks.

Configuring Spanning Tree

The following commands are used to configure the Spanning Tree protocol features:

- **cisco-interoperability**
- **instance priority**
- **instance vlan**
- **max hops**
- **region**
- **revision**
- **spanning-tree enable**
- **spanning-tree hello-time**
- **spanning-tree instance**
- **spanning-tree transmit-holdcount**
- **spanning-tree edgeport**
- **spanning-tree force-version**
- **spanning-tree link-type**
- **spanning-tree root-guard**

- **show spanning-tree mst config**
- **show spanning-tree mst detail**
- **show spanning-tree mst instance interface**

Refer *iBOS Command Reference* for more details on configuration commands for MSTP.

Spanning Tree Configuration Examples

Spanning Tree is disabled by default in iBOS and must be enabled globally and on each interface that will use the protocol. Typical configuration examples are listed below.

Example 56 Enable MSTP

```
ibos(config)# interface gigabitethernet22
ibos(config-if-gi)# spanning-tree enable
ibos(config-if-gi)# exit
ibos(config)# spanning-tree mst
ibos(config-stp)# end
```

To verify the configuration, use the **show spanning-tree brief** command:

```
ibos# show spanning-tree brief
Default Bridge up - Spanning Tree Enabled  detected

MST0
MST0 with 1 ports
Root Id 0008ae860860 (Prio 32768), Bridge Id 0008ae860860 (Prio
32768)
1 topology changes, last change 4m21s ago

Port name        Port    Id         Role      State      Cost  Type
Recv Send
-----------------------------------------------------------------
--------------
gi22              22  8016   Designated Forwarding         0  p2p
none mstp

ibos#
```

If you need to enforce an earlier version of spanning tree, for example RSTP, to comply with devices that lack MSTP support, use the **spanning-tree force-version** command in the interface configuration.

Example 57 Enable RSTP

```
ibos(config)# interface gigabitethernet2
ibos(config-if-gi)# vlan member 11
ibos(config-if-gi)# vlan untagged 11
ibos(config-if-gi)# spanning-tree enable
ibos(config-if-gi)# spanning-tree edgeport
ibos(config-if-gi)# spanning-tree force-version rstp
ibos(config-if-gi)# spanning-tree edgeport bpdu-guard enable
ibos(config-if-gi)# no shutdown
```

PPP over Ethernet (PPPoE)

PPPoE (Point-to-Point Protocol over Ethernet) is a specification for connecting multiple computer users on an Ethernet local area network to a remote site using a common customer premises equipment. PPPoE can be used to make it possible for an office of end users share a common Digital Subscriber Line (DSL), cable modem, or wireless connection to the Internet.

PPPoE combines the Point-to-Point Protocol (PPP), commonly used in dialup connections, with the Ethernet protocol, to support multiple users in a local area network. The PPP protocol information is encapsulated within an Ethernet frame.

PPPoE has become one of the standard technologies used by Service Providers (SP) to connect DSL customers to the Internet by distributing PPPoE clients to end-users.

The iBOS support for PPPoE is described in:

- "PPPoE Intermediate Agent" on page 120
- "PPPoE Server" on page 121

PPPoE Intermediate Agent

In accordance with DSL Forum TR101, it is possible to tag PPPoE Discovery packets at the service provider access switch with end-user information. The PPPoE Discovery packet is used to add subscriber specific information which is then interpreted by the PPPoE Intermediate Agent (PPPoE-IA) before forwarding to the VLAN clients. The PPPoE-IA achieves a lower provisioning overhead and simplifies coordination between OSS systems that manage the access switch and the service provider content server. The PPPoE-IA helps the service provider server to distinguish between end users connected over Ethernet to an access switch.

The PPPoE Intermediate Agent (PPPoE-IA) is placed between the end user and the service provider server to make it possible to identify specific end users connected over Ethernet to an access switch. On the access switch, the PPPoE-IA can mark end user connections by appropriately tagging Ethernet frames from different end users. (The tag contains specific information about which end user is connected to the switch and VLAN.)

The PPPoE-IA acts as a firewall between the clients and the service provider server by intercepting all PPPoE Active Discovery (PAD) messages according to the port and VLAN ids.

In iBOS, the PPPoE-IA functionality is performed by the script **pppoe-ia** and is configured per port. To configure tagging with the **pppoe-ia** script, the following commands are used:

- **port pppoe intermediate-agent vlan**
- **parameter pppoe-ia agent-circuit-id**
- **parameter pppoe-ia agent-remote-id**
- **show pppoe intermediate-agent**
- **script run pppoe-ia show clients**

For more information, refer *iBOS Command Reference*.

PPPoE Server

PPP over Ethernet (PPPoE) provides the ability to connect a network of hosts over a simple bridging access device to a remote Access Concentrator. PPPoE can be used to have an office or building-full of users share a common Digital Subscriber Line (DSL), cable modem, or wireless connection to the Internet. PPPoE combines the Point-to-Point Protocol (PPP), commonly used in dialup connections, with the Ethernet protocol, which supports multiple users in a local area network. The PPP protocol information is encapsulated within an Ethernet frame. Each user will maintain a separate PPPoE session with the Remote Access concentrator and this provides a way to keep track of which user traffic should go to and which user should be billed.

The PPPoE server configuration is described in the following:

- "Configure PPPoE Access Concentrator" on page 122
- "PPPoE Authentication" on page 124
- "Configuring PPPoE Address Assignment" on page 126
- "LCP Negotiation" on page 129
- "LCP Echo Requests" on page 130
- "Collect PPPoE Server Statistics" on page 131
- "PPPoE Data Forwarding" on page 132
- "PPPoE Session Parameters" on page 133
- "Terminate PPPoE Sessions" on page 134
- "PPPoE Redundancy" on page 134

Configure PPPoE Access Concentrator

Many ISPs route their PPPoE connections through access concentrators. These devices allow service providers to grant access to large numbers of users without needing to create a dedicated connection for each one. The advantage is that the ISP can add more users without changing the network architecture.

For PPPoE servers, iBOS supports handling of PPPoE Active Discovery messages from the client in the following way:

1. PPPoE client sends broadcast PADI with host-unique-key
2. PPPoE server responds with PADO by filling it's service name, AC-Name, AC-Cookie and clients host-unique-key
3. PPPoE client sends PADR with the selected AC-cookie, his host-unique-key and service offered by Server
4. Server sends PADS with the session-id along with parameters sent in PADO.
5. Server and client form a PPPoE session.
6. Server and client negotiate the LCP parameters (MRU, authentication protocol, Magic number etc).
7. AAA server authenticates the client using negotiated authentication protocol (iBOS currently supports RADIUS protocol).
8. Server and client negotiate IP address in the IPCP phase, The PPPoE server assigns the client IP address and also provides the DNS server details.

PPPoE Server configuration is supported on the following interface types:

- Physical (port)
- VLAN (vidX)
- Remote-Vid/ Vid-in-Vid
- Aggregator (agX)

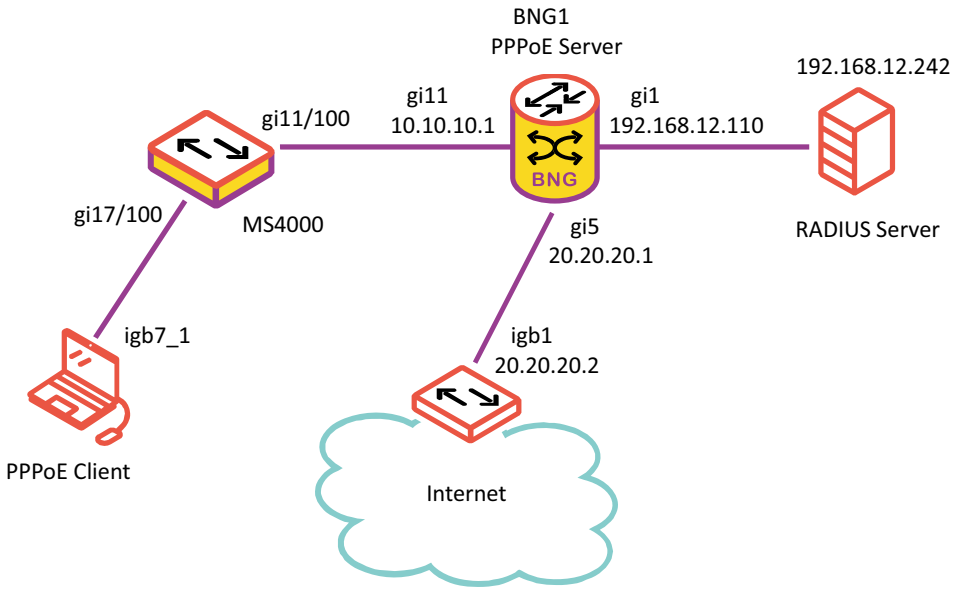
To configure the PPPoE Access Concentrator in the following example, the following commands are used:

- **pppoe tag ac-name**
- **pppoe tag service-name**

For more details on these commands, refer *iBOS Command Reference.*

In the following example, BNG1 is the PPPoE server and is configured with AC-name and service name tags. When the PPPoE client is running, it initiates a PADI message with a service and BNG1 responds with PADO message if the service requested by the client is available. The client and BNG1 perform the protocol exchanges (as already described) and the PPPoE Server issues an IP address to the client.

Figure 24 PPPoE Server Example



To configure the PPPoE Server:

```
BNG1(config)# interface gigabitethernet11
BNG1(config-int-gi)# ip address 10.10.10.1/24
BNG1(config-int-gi)# pppoe tag ac-name waystream
BNG1(config-int-gi)# pppoe tag service-name Internet
BNG1(config-int-gi)# pppoe ipv4-pool 10.10.10.2 10.10.11.2
255.255.255.0
BNG1(config-int-gi)# pppoe ppp-dns primary 2.2.2.2 secondary
3.3.3.3
```

To verify the session, use the command **show pppoe-sessions**:

```
BNG1# show pppoe sessions interface gigabitethernet11 detailed
Session id: 12227
 Username: autotest1
 Interface name: gigabitethernet11
 Remote MAC address: 0000.0000.0100
 Remote IP address: 10.10.10.2
 Session state: up   LCP state: up   IPCP state: up
 Radius details:
  Interface: gigabitethernet11
  On-line: Yes
  Created On: Sat Jun 24 15:53:14 2017
  Uptime: 3m5s
  Session Timeout: 5m
  Account Session Id: gigabitethernet11:37594289
  Status: Access-Accepted

 Configuration details: (i- Interface, p- Profile, d- Default)
  PPPoE Tags:
    Service-Name: Internet(i)
    Ac-name: test(i)
  DNS Primary: 8.8.8.8(i)        DNS Secondary: 2.2.2.2(i)
 Keepalive time: 10(d)         Maxmiss: 3(d)     Idle-time: 120(d)
  PPP MTU: 1492(d)

------------------------------------------------------------------
```

```
--------------
Number of sessions: 1

BNG1#show pppoe sessions
Interface          Session-id State Uptime   Remote-mac    Remote-
ip
-------------------------------------------------------------------
--------------
gigabitethernet11     747        UP    41s       0000.0000.0100
10.10.10.2
```

PPPoE Authentication

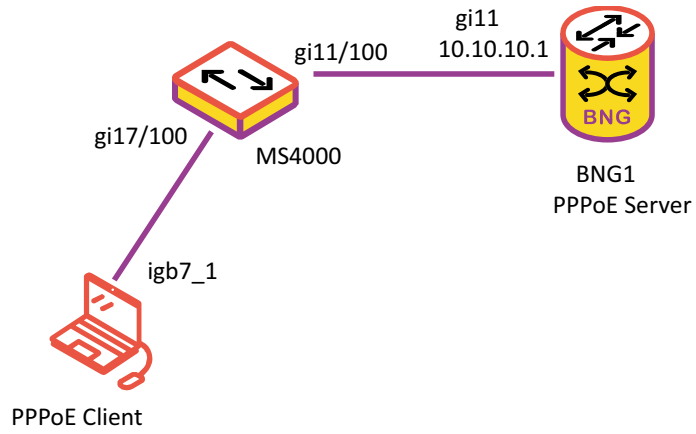
Network administrators need to be able to configure PPP Authentication protocols (e.g. PAP, CHAP).

By default, a PPP authentication server supports both CHAP and PAP. Based on the server and client negotiation, one of these two protocols is chosen for the client authentication. Based on authentication result (Access-Accept or Access-Reject), iBOS sends PAP ACK or PAP NAK message to the client.

iBOS supports both CHAP and PAP protocols and the client negotiates to one of these protocols. The default method for authentication is RADIUS.

- If the authentication protocol is PAP, iBOS performs the handshake negotiation when the link (LCP) is established. The PPPoE client sends the username and password to iBOS which forwards the access request to the RADIUS server. The RADIUS server validates the username and password and, if username and password are validated, the servers returns the “access accept” message with the service templates. If the validation fails, RADIUS sends the “access reject” message.
- If the authentication protocol is CHAP, iBOS sends the “challenge” message to the client after LCP, and the client sends the challenge-response to IBOS which forwards the challenge and challenge response messages to the RADIUS server for validation. If the validation is successful, the RADIUS server sends an “access accept” message. If the validation fails, RADIUS sends the “access reject” message.

Figure 25 PPPoE Server Example



When the client is successfully authenticated, the following takes place:

1. the RADIUS server sends Access-Accept with the service templates configured for the authorized user
2. BNG1 waits until the IPCP phase resolves the IP address for the PPPoE client
3. BNG1 installs the services and starts the accounting-interim updates to the RADIUS server (updates to input and output packets rates are sent to RADIUS according to configured accounting-interim update interval)

PPPoE server configuration commands are configured in the interface context or in the PPPoE profile context and include:

- **parameter pppoe-server service service-selection**
- **pppoe ppp-authentication pap | chap | any**

For PPPoE service-selection mode, the following commands are used:

- **service-selection pppoe-server-sel**
- **parameter pppoe-server auth server-ip <ip address> server-port <port number>**
- **parameter pppoe-server aaa-source-interface <interface-name>**
- **parameter pppoe-server acct server-ip <ip address> server-port <port number>**

Configure PPPoE server settings for BNG1:

```
BNG1(config)# interface te2/4/vid602
BNG1(config-if-vid)#  pppoe ppp-authentication pap
BNG1(config-if-vid)#  no shutdown

BNG1(config)# service-selection pppoes-ss
BNG1(config)# parameter pppoe-server aaa-source-interface ethernet0
BNG1(config)# parameter pppoe-server auth server-ip 10.193.0.231
server-port 1812 secret waystream
BNG1(config)# parameter pppoe-server acct server-ip 10.193.0.231
server-port 1813 secret waystream
```

To show check the PPPoE session details, use **show pppoe sessions interface detailed:**

```
BNG1#show pppoe sessions interface   gigabitethernet11 detailed
Session id: 13834
 Username: autotest1
 Interface name: gigabitethernet11
 Remote MAC address: a036.9f9a.fa0e
 Remote IP address: 10.10.10.2
 Session state: up   LCP state: up   IPCP state: up
 Radius details:
  Interface: gigabitethernet11
  On-line: Yes
  Created On: Wed Jun 28 18:18:54 2017
  Uptime: 15s
  Session Timeout: 5m
  Account Session Id: te0/3/vid100:16901928
  Status: Access-Accepted

 Configuration details: (i- Interface, p- Profile, d- Default n-
Negotiated)
  PPPoE Tags:
    Service-Name: Internet(i)
    Ac-name: test(i)
  DNS Primary: 8.8.8.8(i)        DNS Secondary: 2.2.2.2(i)
  Keepalive time: 10(d)         Maxmiss: 3(d)     Idle-time: 120(d)
  PPP MTU: 1492(d)

-------------------------------------------------------------------
--------------
Number of sessions: 1
```

Configuring PPPoE Address Assignment

iBOS supports handling of IP Control Protocol (IPCP) negotiation in the network layer, IP address, and DNS server details are negotiated.

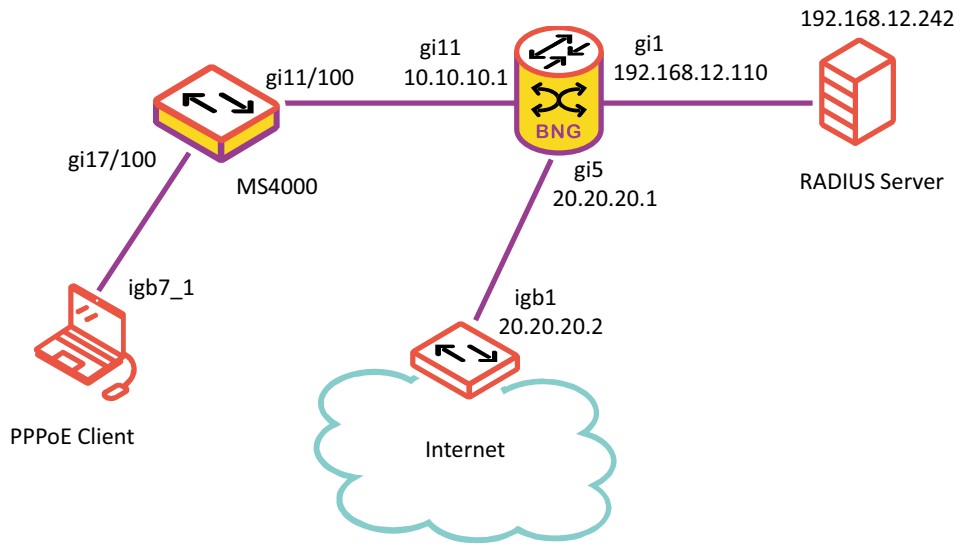
Configure PPPoE Client IP Address

During IPCP phase, client and server negotiate the IP address:

1. The client sends an IPCP request with specific IP address or sends 0.0.0.0. If the request is 0.0.0.0, the server will assign the IP Address available in pool.
2. If the specific IP address is available in server pool, the PPPoE server sends the configuration ACK. If the IP address is not available, the server sends the configuration nak with an available IP address from the pool.
3. The client sends the config request with the IP address supplied by the PPPoE server.
4. The PPPoE server sends the ack so that the client gets the IP address and sets the interface state to UP.

In the following example, the PPPoE client sends the IPCP config request with specific IP address.

Figure 26 PPPoE IPCP Example



To configure IPCP, use the **pppoe ipv4-pool** command either in interface context or pppoe profile context.

Configure BNG1:

```
BNG1(config)# interface gigabitethernet11
BNG1(config-gi11)# ip address 10.10.10.1/24
BNG1(config-gi11)# pppoe tag ac-name waystream
BNG1(config-gi11)# pppoe tag service-name Internet
BNG1(config-gi11)# pppoe ipv4-pool 10.10.10.2 10.10.11.2
255.255.255.0
```

To verify the session details, the command **show pppoe sessions** is used.

To verify the IP address issued to client:

```
BNG1# show pppoe sessions
Interface          Session-id State Uptime    Remote-mac    Remote-
ip
------------------------------------------------------------------------
--------------
gigabitethernet11     29075      UP    1m35s     a036.9f9a.fa0e
10.10.10.2

BNG1# show pppoe sessions interface gigabitethernet11
Interface          Session-id State Uptime    Remote-mac    Remote-
ip
------------------------------------------------------------------------
--------------
gigabitethernet11     29075      UP    3m30s     a036.9f9a.fa0e
10.10.10.2

BNG1# show pppoe sessions interface gigabitethernet11 detailed
Session id: 11007
 Username: autotest1
```

```
 Interface name: gigabitethernet11
 Remote MAC address: a036.9f9a.fa0e
 Remote IP address: 10.10.10.2
 Session state: up   LCP state: up   IPCP state: up
 Radius details:
  Interface: gigabitethernet11
  On-line: Yes
  Created On: Wed Jun 28 18:29:04 2017
  Uptime: 2m53s
  Session Timeout: 5m
  Account Session Id: gigabitethernet11:03101590
  Status: Access-Accepted

 Configuration details: (i- Interface, p- Profile, d- Default n-
Negotiated)
  PPPoE Tags:
    Service-Name: Internet(i)
    Ac-name: test(i)
  DNS Primary: 8.8.8.8(i)        DNS Secondary: 2.2.2.2(i)
  Keepalive time: 10(d)        Maxmiss: 3(d)     Idle-time: 120(d)
  PPP MTU: 1492(d)

------------------------------------------------------------------
--------------
Number of sessions: 1
```

Configure DNS Server for PPPoE Clients

After discovery phase, during IPCP negotiation along with IP address, the IP addresses for the primary DNS and/or secondary DNS server are negotiated between the server and the client.

The client can send an IPCP request to query the primary and/or secondary DNS server. The server sends the primary and/or secondary DNS servers based on the current server configuration.

Note: *If the client sends a request for IP 0.0.0.0 or a specific DNS server IP that is not configured on the server, the PPPoE server sends Nak with the configured DNS server IP for primary and/or secondary DNS.*

During IPCP phase PPPoE client sends config request with particular primary and/or secondary DNS servers. If the requested primary and/or secondary DNS servers exists in BNG1, then BNG1 sends config-ack with requested primary and/or secondary DNS servers. In case if client request for specific DNS server IP or IP 0.0.0.0, which is not configured at BNG1, BNG1 sends NAK with configured DNS server IP for primary and/or secondary DNS

To configure DNS for clients on PPPoE Server in “PPPoE IPCP Example” on page 127, use the command **pppoe ppp-dns** in the inteface context or the pppoe profile context. For example:

```
BNG1(config)# interface gigabitethernet11
BNG1(config-gi11)# pppoe ppp-dns primary 192.168.10.126 secondary
8.8.8.8
```

To verify the configuration for the interface:

```
BNG1# show running-config context interface gigabitethernet11
 pppoe ppp-dns primary 192.168.10.126 secondary 8.8.8.8
```

To show check for re-configured DNS servers, use the command **show pppoe sessions interface detailed:**

To verify that Primary and Secondary DNs servers issues to client:

```
BNG1# show pppoe sessions interface gigabitethernet11 detailed
Session Id: 14149
Username:
Remote MAC address: a036.9fa1.1b6b
Remote IP address: 10.10.10.149
 Profile name:
 AC name: waystream
 Max-payload: 0
 Authentication Protocol:
 Authentication Mode: Local
 Keep alive Time: 20
 Maxmiss: 3
 PPP MTU: 1400
 PPP MRU: 1300
 Primary DNS Server: 192.168.10.126
 Secondary DNS Server: 8.8.8.8
 LCP State: Up
 IPCP State: Up
 Session State: Up
 Radius Details:
  Interface:
  Remote MAC address:
  On-line:
  Created On:
  Expired In:
  Account Session Id:
  Status:
  Authentication Protocol:
  Session Timeout:
  Service:
  Input Rate:
  Output Rate:

------------------------------------------------------
Number of sessions: 1
```

LCP Negotiation

Once the PPPoE discovery phase is over, the PPPoE session starts with LCP negotiation for Link and PPP parameters during the Link Establishment Phase. During this phase, the MTU, authentication protocol (PAP or CHAP), and Magic-number are negotiated between iBOS and the PPPoE client.

- The MRU for the session is the lowest MRU value.
- The authentication protocol is the one supported by both iBOS and PPPoE client.
- An unique Magic-number is negotiated in both the directions.

For the example “PPPoE Server Example” on page 125, the following occurs:

1. The PPPoE client has MRU of 512 and BNG1 MTU is 1492 so the negotiated MTU will be 512.
2. BNG1 supports both CHAP and PAP, so CHAP is first sent for negotiation in LCP

config request to client. CHAP is not supported by client, so the client sends LCP config nak. BNG1 then sends the LCP config request with PAP as authentication protocol and this time client sends LCP config ACK.

3. BNG1 and the PPPoE client both send LCP config requests in both directions with their Magic-number, so when PPPoE client receives config request from BNG1, the client checks the Magic-number to make sure it is not same as the number sent in the client config request. BNG1 performs the same check.

To see the client negotiated MRU with iBOS MTU:

```
BNG1#show pppoe sessions interface gigabitethernet11 detailed
Session id: 5711
 Username: autotest1
 Interface name: gigabitethernet11
 Remote MAC address: 0000.0000.0100
 Remote IP address: 10.10.10.2
 Session state: up   LCP state: up   IPCP state: up
 Radius details:
  Interface: gigabitethernet11
  On-line: Yes
  Created On: Sat Jun 24 16:47:59 2017
  Uptime: 2m16s
  Session Timeout: 5m
  Account Session Id: gigabitethernet11:93476389
  Status: Access-Accepted

 Configuration details: (i- Interface, p- Profile, d- Default)
  PPPoE Tags:
    Service-Name: Internet(i)
    Ac-name: test(i)
  DNS Primary: 8.8.8.8(i)        DNS Secondary: 2.2.2.2(i)
  Keepalive time: 10(d)         Maxmiss: 3(d)     Idle-time: 120(d)
  PPP MTU: 512(d)
```

LCP Echo Requests

To determine the state of the PPPoE session, iBOS can be configured to occasionally send LCP Echo-Request packets to the client.

When the PPPoE client has formed a session with BNG1 and acquired an IP address, the following command configures the keep-alive-time to 10 seconds, the max-miss as 3 seconds, and the idle-time as 10 seconds:

```
BNG1# pppoe keep-alive-time 10 max-miss 3 idle-time 10
```

To view the state of the PPPoE session:

```
BNG1# show pppoe sessions
Interface          Session-id State Uptime   Remote-mac    Remote-
ip
-------------------------------------------------------------------
--------------
gigabitethernet11     29075      UP    1m35s     a036.9f9a.fa0e
10.10.10.2
```

The PPPoE Server checks at keep-alive-time 10sec for no data or LCP echo-replies has received within the last idle-time of 10sec then sends the LCP echo request packet else again checks at keep-alive-time 10 secs for no data or LCP echo-replies has received within the last idle-time of 10sec.

If max-miss equals 3, this means that if consecutive LCP echo replies are not received for 3 consecutive LCP echo requests, the server terminates the session. To determine the session is no longer active takes 30 seconds or, if data exists, the last idle-time 10 seconds data is added and it takes 40 seconds to terminate the session.

After 30 seconds, the session is removed if LCP echo-replies are not received.

```
BNG1# show pppoe sessions
BNG1#
```

Collect PPPoE Server Statistics

iBOS provides detailed PPPoE server statistics for Discovery and session messages, PPPOE session details, PPPoE interface status, etc.

PPPoE and PPP Tx counters are incremented whenever a packet is transmitted from the server and Rx counters are incremented whenever a packet is received by the server

In the“PPPoE Server Example” on page 123, BNG1 acts as the PPPoE server. During the Discovery phase a client sends PADI and the server responds with PADO. The client then replies with PADR and the server responds with PADS.

During this PPPoE session phase, the server and client exchanges: LCP/IPCP config request, LCP/IPCP config Ack, LCP/IPCP config reject, LCP/IPCP config Nak, LCP/IPCP Protocol reject, LCP echo request and LCP echo replies and the drop count increments when the particular packet is dropped due to unavailability of resources or the admin restricted the number of sessions

To verify both detailed PPPoE and PPP server statistics, use the commands:

- **show pppoe counters interface detailed**
- **show pppoe sessions**

```
BNG1# show pppoe counters interface gigabitethernet11 detailed
Packets      Sent         Received     Dropped
-------------------------------------------------------------------
-------------
PADI         0            71           0
PADO         71           0            0
PADR         0            71           0
PADS         71           0            0
PADT         70           70           0
-------------------------------------------------------------------
-------------
TOTAL        212          212          0

PPP Counters::
Packets      LCP-Tx  LCP-Rx  IPCP-Tx  IPCP-Rx
-------------------------------------------------------------------
-------------
CONF_REQ     142     142     71       143
CONF_ACK     71      142     71       71
CONF_NAK     0       0       1        0
CONF_REJ     71      0       71       0
TERM_REQ     0       0       0        0
TERM_ACK     0       0       0        0
CODE_REJ     0       0       0        0
```

```
PROTO_REJ    71      0       0        0
ECHO_REQ     140     0       0        0
ECHO_REPLY   0       140     0        0
DISCARD_REQ  0       0       0        0
------------------------------------------------------------------
-------------
TOTAL        495     424     214      214


Auth Counters::
Packets          Sent         Received
------------------------------------------------------------------
-------------
PAP_REQ          0            71
PAP_ACK          71           0
PAP_NAK          0            0
CHAP_CHALLENGE   0            0
CHAP_RESPONSE    0            0
CHAP_SUCCESS     0            0
CHAP_FAILURE     0            0
------------------------------------------------------------------
-------------
TOTAL            71           71
```

```
BNG1#show pppoe sessions
Interface          Session-id State Uptime   Remote-mac    Remote-
ip
------------------------------------------------------------------
-------------
gigabitethernet11     5900       UP    51s       0000.0000.0100
10.10.10.2
to client

BNG1#show pppoe sessions interface gigabitethernet11
Interface          Session-id State Uptime   Remote-mac    Remote-
ip
------------------------------------------------------------------
-------------
gigabitethernet11     5900       UP    2m42s     0000.0000.0100
10.10.10.2
```

PPPoE Data Forwarding

iBOS forwards traffic to/from subscribers over the PPPoE session.

Once the PPPoE session is formed and the client acquires the IP address from the PPPoE Server, all traffic that comes from the client is PPPoE encapsulated. When the server receives the traffic, the PPPoE server decapsulates the PPPoE and PPP header, and sends the IP packet to IP module for further processing.. For external traffic destined to the PPPoE client, the PPPoE server encapsulates all the traffic with a PPPoE header before sending the data to the client.

Note: *Decapsulation and encapuslation of the PPPoE header is identical for all types of interfaces; physical, remote-VID, VID, VID-in-VID, VLAN and Aggregator.*

In “PPPoE Server Example” on page 125, the PPPoE client gets the IP address 10.10.10.2 from the PPPoE server IPv4 pool, then the client sends PPPoE encapsulated traffic to Internet 20.20.20.2. When BNG1 receives this traffic, the BNG decapsulates the PPPoE Header and sends it to the Internet next hop. Similarly, when Internet 20.20.20.2 sends native IP traffic to PPPoE client 10.10.10.2, BNG1 encapsulates the received traffic with PPPoE header and sends it to the PPPoE client.

PPPoE Session Parameters

The PPPoE session parameters are:

- MTU
- Authentication Type
- IP-Address Pool
- Remote-Peer address assignment
- Maximum PPPoE sessions
- Maximum PPPoE sessions per Remote-MAC

The parameters for PPPoE sessions can be configured either in profile or interface contexts using the following commands:

Table 17 PPPoE Configuration Commands

Context	Commands
PPPoE profile	tag ac-name <ac-name> tag service-name <servicename> ppp-mtu <mtu-value> ppp-authentication <any/pap/chap> max-sessions profile <value> max-sessions snpa <mac-address> <value>
Interface	pppoe tag ac-name <ac-name> pppoe tag service-name <servicename> pppoe ppp-mtu <mtu-value> pppoe ppp-authentication <any/pap/chap> pppoe ipv4-pool <start-ip> <end-ip> <IPv4 mask> pppoe remote-peer-ip <host-ip> <IPv4 mask> pppoe max-sessions interface <value> pppoe max-sessions snpa <mac-address> <value>

For more details on the PPPoE commands, refer *iBOS Command Reference*

Terminate PPPoE Sessions

A PPP peer should use the PPP protocol itself to bring down a PPPoE session, but the PADT MAY be used when PPP can not be used.

After a session is established, the client or the access concentrator can send a PPPoE Active Discovery Termination (PADT) packet anytime to terminate the session. The PADT packet contains the destination address of the peer and the session ID of the session to be terminated. After this packet is sent, the session is closed to PPPoE traffic.

The BNG sends PADT when the session has to be terminated.

To clear the established session, use the command **clear pppoe session session-id**.

In the example “PPPoE Server Example” on page 125, the PPPoE client establishes an session with BNG1 (PPPoE server). When the established session is cleared, BNG1 sends PADT to the client and terminates the session:

```
BNG1#show pppoe sessions
Interface          Session-id State Uptime   Remote-mac    Remote-
ip
-------------------------------------------------------------------
--------------
gigabitethernet11     52745      UP    2m2s      0000.0000.0100
10.10.10.2
BNG1# clear pppoe session session-id 52745
BNG1# show pppoe sessions
```

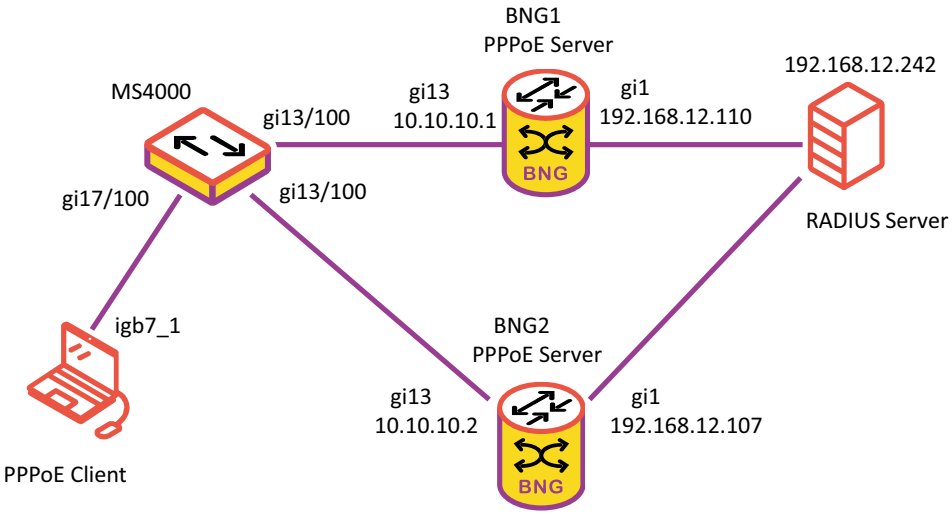
PPPoE Redundancy

Two BNGs are maintained for redundancy in a LAN when the Service Provider wants to maintain PPPoE sessions even when the primary BNG is down. This redundancy can be achieved by delaying the PADO packet response on the secondary server after the PADI is received. The clients form sessions with the BNG that first responds to the PADI packets which will be the primary BNG.

If the primary BNG fails, the secondary BNG continues to responds to clients with an PADO after the configured PADO delay and the clients form sessions with secondary BNG instead.

In the following example, BNG1 is the primary PPPoE server with no PADO delay and BNG2 is the secondary server with a PADO delay of 1 second.

Figure 27 PPPoE Redundancy Example



If a PPPoE client initiates a session by sending PADI with any service, the BNGs are configured with the client requested service and both servers will receive the PADI, but the BNG1 responds immediately while BNG2 delays the PADO packet 1 second. All clients establish the sessions with the BNG1 because it responds with PADO first for their PADI packets.

To configure the PADO delay, use the command **pppoe pado-delay**

To configure BNG2 as secondary server:

```
BNG2(config)# interface gigabitethernet13
BNG2(config-int-gi)#  pppoe pado-delay 1000
```

Layer 3 Specific Configuration

iBOS supports Layer 3 configuration that provides IP routing all the way to the access edge.

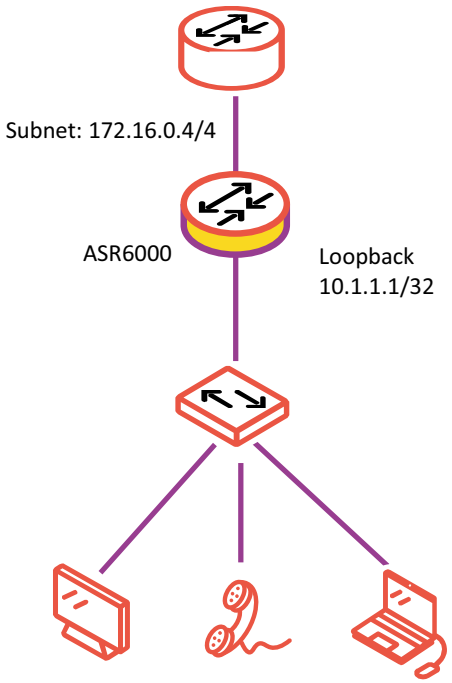
Layer 3 specific configuration is described in the following sections:

- “Layer 3 Standard System” on page 138
- “Layer 3 Customer Interfaces for Residential Services” on page 141
- “Layer 3 Services” on page 142

Layer 3 Standard System

In a standard Layer 3 system configuration, the iBOS device is configured to use routing protocols to learn the routes of the network. Preferably, a loopback interface is used for network management of the iBOS device.

Figure 28 Layer 3 Access Edge System



The config-register net-boot controls if the iBOS device should attempt BOOTP/DHCP on its uplink interfaces to obtain an initial IP and system configuration. Some iBOS devices, such as the ASR switches, have the net-boot default enabled. If the upstream router is configured to relay the requests to a BOOTP server, the IP address for the uplink interface and for a default configuration file are included in the BOOTP response.

In the example, there is a static configuration for the standard system which includes the uplink interfaces. The following is configured:

- The router hostname only has local significance to identify the system you are working on. A local user account 'test' with password 'test' is set up, and the enable mode is also protected with the password 'test'. Remote authentication using RADIUS is possible using the **port authentication 802.1X** command, refer to the iBOS Command Reference for details.
- The OSPF routing protocol is set up to make the router a member of an NSSA area with area-id 10. The loopback and uplink subnets are included.
- In addition, a route-map is configured to redistribute the routes of connected clients. When clients are assigned IP addresses from a DHCP server, their addresses can be automatically redistributed to other routers. This is a unique Waystream feature that allows an IPv4 subnet to be used across multiple routers within the same OSFP area.

- SNMP is set up to allow community public to read all supported MIBs using SNMP version 2.
- Logging is set up so that the router records log information to the local log and to a remote telnet connected session. If debugging is turned, the log also includes debugging output. These log messages can also be sent to a central server using syslog.
- The system is also set up to provide date-time formatting of time stamps in the log. Use the **clock set** command to set the local system time for correct time stamping.

Example 58 Standard Router Configuration in Layer 3 Network

```
hostname TEST-ASR
username test password test
enable secret test

interface loopback0
 ip address 10.1.1.1 255.255.255.255
 no shutdown
 gigabitethernet1
 description Uplink interface
 ip address 172.16.0.2 255.255.255.252
 no shutdown

router ospf
 router-id 10.1.1.1
 redistribute kernel metric 10 route-map ospf_redist_kernel
 redistribute connected metric 10 route-map ospf_redist_connected
 redistribute static metric 10 route-map ospf_redist_static
 area 10 nssa
 network 172.16.0.0/30 area 10
 network 10.1.1.1/32 area 10

!
! Clients that are discovered through DHCP snooping are created
automatically in the
! internal routing table of the ASR. They have a special flag, the
client flag, that
! allows the route-map to identify them and automatically distribute
them without
! having to create an access-list that matches their assigned IP
address.
!
route-map ospf_redist_connected 10 permit
 match client

route-map ospf_redist_kernel 10 permit
 match client

route-map ospf_redist_static 10 permit
 match client


! Define an access view that includes the entire MIB
snmp view all include .1

snmp-group version 2c
```

```
 community public read-write view all

!
logging console informational
logging monitor debugging
logging buffered informational
logging trap debugging
logging facility local7
logging accounting facility local7

service timestamp debug datetime msec
service timestamp log datetime msec

ip name-server 10.64.1.253
ip domain-name lab.waystream.com

!
! Your upstream router should provide a default route by OSFP.
! If not, you can use the command below by removing the '!'
! ip route 0.0.0.0/0 172.16.0.1 254
```

Layer 3 Customer Interfaces for Residential Services

Downlink interfaces are normally intended for connection to end-users, or customers.

A downlink interface has several commands related to security to prevent end-users sending packets from illegal hosts/addresses.

iBOS has a special, virtual interface, called **downstream0**, which has two main purposes:

1. To store the default-gateway address of subnet(s) used for clients. This is part of the functionality that allows the subnet to span across multiple routers and allows the router to perform proxy-ARP for this address on all interfaces where clients from the configured subnet is connected.
2. To handle IGMP Querier and IGMP snooping functionality for the physical interfaces. Downstream0 acts as the IGMP Querier interface on all physical interfaces that have an active multicast member

The customer interface is configured with an access-list that create a special handling of DHCP packets, which is required for the DHCP snooping functionality.

The **ip strict-clients** configuration ensures that traffic received on the port is only accepted from IP addresses that have been assigned by DHCP to a client on the port. This is an anti-spoofing protection mechanism.

The ARP proxy ensures that the router responds to ARP queries for addresses it knows about. This allows the router to respond with the client default-gateway address configured on downstream0 and also on behalf of other addresses in the subnet of the client to allow communication between local and remote ports with clients in the same subnet.

Example 59 Customer Interface Configuration in Layer 3 Networks

```
interface fastethernet1
 description Customer interface
 ip access-group cust-in in
 ip strict-clients
 arp proxy
 no shutdown

access-list cust-in
 seq 10 permit udp any 68 any 67 shortcut
 seq 20 permit udp any 67 any 67 shortcut
 seq 30 permit ip any any
```

Layer 3 Services

Service Definition

A service is defined by a set of policy commands. Policy commands define rate-limiting of traffic, access-lists to filter traffic and other policy related features that generates the service.

A policy-map command ties a set of policies with a common name to a classifier that identifies the host(s) for which the policy should apply. In a Layer 3 access edge deployment, the IP address of the client(s) to be part of the service is typically used in the policy-map but there are also other alternatives for mapping a policy.

For a complete list of policy and policy-map commands, refer *iBOS Command Reference*.

Service Templates

The configuration required to activate a particular service is very similar between services. In many cases the only difference between two Internet services is the speed of the service or the IP address range for which the service should apply.

A feature in iBOS allows the definition of a Service Template. The template is the blue print for the configuration to apply when activating a service. Some parts of the configuration, such as the bandwidth of the service, can be left out of the Service Template and referred to by a variable name.

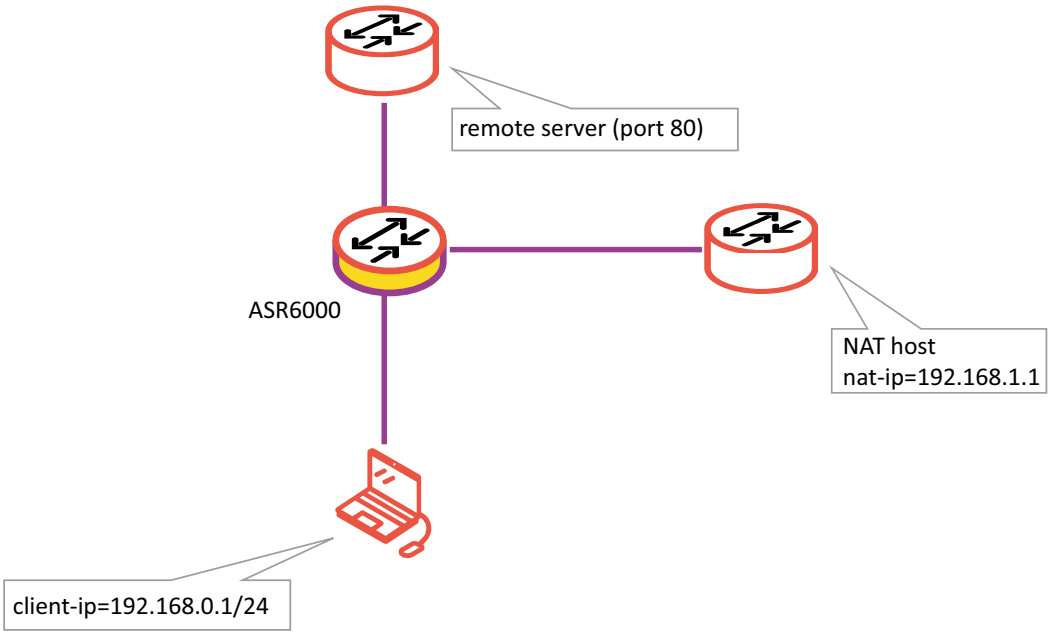
When a client device connects to the network and requests an IP address by DHCP, the DHCP packet can be used to identify the type of client that connects, and thereby the service to apply. A service template can have client-conditions defined, that are rules that if matching the DHCP packet will cause the template to be applied as a service for the client.

The result of using Service Templates is that the router by itself, or with cooperation of a central server, will activate the configuration needed to enforce the service for the client without any need of manual or automated configuration of the router.

The service template consists of three components:

- *Interface Configuration:* defines which services (service templates) that are selectable for clients on the physical interface, and provides custom settings for this particular end-user/service on this interface
- *Service-selection Configuration:* defines the client-conditions or method by which the router will evaluate if the client's DHCP packet matches the rules for the service to apply to the particular client
- *Service-definition Configuration:* defines which configuration commands to activate in order to enforce the service for clients that match the service-selection criteria.

Figure 29 Service Template in Layer 3 Networks



The service selection in this example matches any client and relays the DHCP packet to a DHCP server, setting option-82 information and the giaddr to the address configured on downstream0.

The DHCP relay and service template functionality in iBOS is implemented using a customizable script in the Python programming language. iBOS 6.1 contains several default scripts, including one named dhcp for IPv4. There is also a script for IPv6. The interface fastethernet1 where the client is connected is set to use the dhcp script with the service-definition and service-selection of the selected service. The sequence number determines the order by which service-selection criteria is evaluated against the client DHCP packet. The first matching service-selection will determine the service-definition that is used. If there are multiple services on the port, these have to be listed in the sequence of evaluating, preferably starting the the service that has the most specific matching criteria. In this example, the single service has a match criteria of any MAC address, which will make every client on the interface receive this example service.

Example 60 Service Template in Layer 3 Networks

```
interface downstream0
 no shutdown

interface fastethernet1
 ip dhcp relay script
 parameter dhcp service seq 10 service-definition service_Internet
 service-selection service_Internet


service-selection service_Internet
 parameter dhcp match snpa any relay
 parameter dhcp relay-to dhcp-server 172.16.100.1 giaddr
```

```
172.16.1.254 option-82


service-definition service_Internet
 parameter dhcp policy-conditioner-egress shape 10000 green mark 0
 parameter dhcp policy-conditioner-ingress shape 10000 green mark 0
```

IPv4

The Internet Protocol version 4 (IPv4) is described in the following sections:

- “IPv4 Addressing” on page 146
- “IPv4 Routing” on page 154
- “OSPF v2” on page 154
- “PIM” on page 160
- “IGMP” on page 170
- “IPv4 Forwarding” on page 176

IPv4 Addressing

IPv4 addressing is described in:

- "iBOS BOOTP Client" on page 146
- "DHCP" on page 146
- "Configuring BOOTP/DHCP Relay" on page 148
- "DHCP and PXE Relay" on page 149
- "ARP" on page 150

iBOS BOOTP Client

iBOS contains a BOOTP client that is activated during system startup if the netboot configuration register is enabled. The BOOTP client is active on all system uplink interfaces - even if configuration is saved to the system flash.

Which interfaces that are default uplink interfaces depend on the hardware platform. For example, on the ASR6000, tengigabitethernet interfaces are default uplink interfaces.

An interface can be configured/deconfigured as uplink interface with the **port uplink** configuration command. If a non-default interface is used as uplink interface, the **port uplink** configuration must be saved to the system flash to allow the BOOTP client to operate on the interface after the next restart of the system.

The BOOTP client can be completely shut down with the **ip bootp-client disable** configuration command. If used, the client will no longer monitor uplink interfaces for loss of address configuration.

If the BOOTP response received originates from a BECS version 3 system, the BOOTP client will set the IP address received on the interface but will continue to operate on all other uplink interfaces in the system even if configuration received from BECS sets an IP address on that interface. The continued operation protects against some types of human errors in configuration templates.

The client continues to operate in the background monitoring all uplink interfaces for changes in IP address configuration. If the IP address received with BOOTP is removed (deconfigured) the BOOTP client will resume operation to attempt to recover the lost address.

DHCP

DHCP is built on a client-server model, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. This protocol is commonly used to assign IP addresses to residential broadband customers, but is also used in other scenarios.

There are three mechanisms for IP address allocation in DHCP.

- *Automatic allocation* – A permanent IP address is assigned to the client by DHCP.
- *Dynamic allocation* – An IP address with a limited lease time is assigned to the client.
- *Manual allocation* – Allows a network administrator to assign a specific IP address to the client.

DHCP messages are based on the format of BOOTP messages. This makes existing BOOTP clients inter operate with DHCP servers. Using BOOTP relay agents eliminates the need for a DHCP server on each physical network segment.

The BOOTP protocol is described in RFC 951. The DHCP protocol is described in RFC 2131

DHCP Operation

Example 61 DHCP Request

```
Host A             DHCP Server
DHCPDISCOVER ---> Broadcast
                   Unicast   <--- DHCPOFFER
DHCPREQUEST  ---> Broadcast
                   Unicast   <--- DHCPACK
```

The example above outlines the protocol operation. The client host broadcasts a DHCPDISCOVER message on the local area network, which is received by all present hosts. A connected DHCP server responds to the broadcast with an address and configuration offer that the client then turns into a request. The request is finally acknowledged by the DHCP server.

DHCP Relay Agent

If a server and a client are not on the same physical subnet, the DHCP messages need to be relayed. Before forwarding the packet as a unicast packet to the DHCP server, relay agents alter some of the information in the DHCP request from the client. If a relay agent receives a packet with 0.0.0.0 contained in the gateway IP address field (within the header), then it is the closest relay agent to the client.

iBOS can operate as a DHCP relay-agent on routers. This makes the system capture all packets destined for UDP port 67 and 68 (the BOOTP/DHCP ports) and relay them to a central server.

DHCP Relay Options

When a DHCP relay agent intercepts a DHCP request, it adds the DHCP relay agent option to the packet. Information in this option can be used by the DHCP server to distinguish between clients. The DHCP relay agent option has two sub options.

- Agent Circuit ID sub-option
- Agent Remote ID sub-option

Both these sub-options can be used by iBOS . The **ip relay information** configuration command enables iBOS to add the sub-options to the relayed DHCP packet.

In the first option, Circuit ID, iBOS will enter the port id of the interface receiving the DHCP message.

In the second option, Remote ID, the MAC address of the host requesting an address will be entered.

This information allows the DHCP server to maintain a record of which port on which iBOS system was assigned to a specific IP address at any given time.

Configuring BOOTP/DHCP Relay

To configure iBOS for BOOTP/DHCP relay, the relay agent must first be set up.

The interface **ip bootp** and **ip dhcp** configuration commands control the relay of BOOTP and DHCP requests. The IP address of the DHCP server to which received BOOTP/DHCP packets should be forwarded must be supplied. It is possible to specify different settings for BOOTP and DHCP packets. When relay of BOOTP/DHCP packets are enabled, iBOS will automatically create static ARP entries for hosts assigned addresses with BOOTP/DHCP. This allows dynamic learning of ARP entries to be disabled on client interfaces.

The **ip dhcp relay information replace** configuration command should also be used to add necessary information to the relayed packets so that the DHCP server can determine the appropriate configuration to deploy in the system for each client.

The **ip dhcp relay source-interface** configuration command should also be used in order to always relay packets from the same address in the system. This further helps the DHCP server to identify the appropriate configuration to deploy.

The **ip dhcp relay server-id replace** configuration command must also be present. This makes iBOS provide its own local interface address as DHCP server address to the client. All DHCP communication with the DHCP server will then be directed to the iBOS system by the client allowing iBOS to control lease times. The **show clients** command can display the current DHCP leases handled through iBOS.

Example 62 DHCP Relay

```
interface loopback1
  ip address 192.168.5.1 255.255.255.255

interface gigabitethernet1
  ip address 172.16.9.1 255.255.255.0
  ip bootp relay 10.52.1.12

interface fastethernet1
  ip dhcp csar

ip dhcp relay information replace
ip dhcp relay source-interface loopback1
```

BOOTP/DHCP Relay for BECS

If the system is controlled by BECS, DHCP/BOOTP packets on client interfaces are normally relayed to the BECS system. In BECS controlled networks, the **csar** keyword inserts the IP address of the BECS system when configured. This forces iBOS to relay BOOTP/DHCP packets to the same address as the current CSAR session since that address is usually the same as the one used for DHCP/BOOTP.

The **csar** keyword should not be used on uplink interfaces, as they are normally handled by a different server than clients in a BECS controlled network. Instead, the configuration deployed by BECS to the server can include the `auto.bootp_server` dynamic parameter statement in the configuration community to set the relay destination for uplink interfaces.

CSAR connection is established immediately during startup and has automatic notification to the relay agent.

The source-interface should be the same interface that received the element.ip configuration from BECS; typically one of the loopback interfaces.

DHCP and PXE Relay

PXE (Pre-Boot Execution Environment) servers allow workstations to boot from a server in a network so that network administrator does not have to physically visit the specific workstation to upgrade the OS installation. To integrate a PXE server, or an external DHCP server, the following two commands in iBOS are used:

- **ip dhcp relay server-id replace**
- **ip dhcp relay source-interface**

These commands are configured in the interface context and in the client context, respectively.

Example 63 Configuration Syntax for PXE Server on Interface Context

```
ip dhcp { csar | <ip> }
    {
      { source-address <x.x.x.x> | source-interface <iface } |
      giaddr <x.x.x.x> |
      server-id-replace |
      drop-ip-assignment | *
    }
```

Example 64 Configuration Syntax for PXE Server on Client Context

```
 ip dhcp$ <ip>
    {
      { source-address <x.x.x.x> | source-interface <iface } |
      giaddr <x.x.x.x> |
      server-id-replace |
      drop-ip-assignment | *
    }
```

The following should be noted:

- giaddr is set to the same as the source address unless specified explicitly
- if **ip dhcp relay server-id replace** is configured in the root context, this will override any configuration of the interface relay command
- **drop-ip-assignment** is just a flag telling clientmgr that this server is not allowed to give out ip addresses, and if it tries to the packet will be dropped. Typically, this must be configured for the PXE-server and is basically a safety net in case someone wrongly configures the PXE server to be a DHCP server as well. This command prevents the clientmgr from setting an ARP entry when the client receives an IP from a DHCP server.

Example 65 Configuration of PXE Server

```
int fa2
  ip dhcp relay <dhcp-ip> source-interface fa2
  ip dhcp relay <pxe-ip> source-interface fa2 drop-ip-assignment
  no ip dhcp gateway auto
```

The **no ip dhcp gateway auto** is necessary to stop some of the clientmgr behaviour that is configured by default. The gateway statement will prevent clientmgr from setting an ip statement on downstream0 and the drop-ip statement prevents the server handing out ip-addresses.

The interface then needs to be configured for the server uplink:

Example 66 Configuration of PXE Server Uplink

```
int fa9
  ip dhcp csar source-interface loopback0 server-id-replace
 ip dhcp relay <pxe-ip> source-address <x.x.x.x> drop-ip-assignment
```

The following line can also be added to the service definition:

```
ip dhcp relay <pxe-ip> source-address <x.x.x.x> drop-ip-assignment
```

ARP

Ethernet is a Layer 2 protocol. Every host connected to an Ethernet network has an Ethernet address, also known as a MAC or hardware address.

A host connecting to the Internet must also have an IP address. IP works on layer3 of the OSI model.

This layering means that there needs to be a connection between the Layer 2 MAC-address and the Layer 3 IP address - if hosts on the same network are to exchange traffic on Layer 3. For IPv4, that connection is the Address Resolution Protocol (ARP). It is used by hosts to translate between Layer 2 Ethernet MAC-addresses and Layer 3 IP addresses on the local IP subnet (local meaning the subnet the host is attached to).

In most cases, the IP address of the destination is known beforehand, but the MAC-address is not. ARP enables the host to find the MAC-address of the destination.

ARP was specified in RFC826 and is an Internet Standard.

Understanding ARP

Whenever a host (A) on a network needs to obtain the hardware address of another host (B) on the network, for which the layer3 IP address is known, the host generates an Ethernet broadcast to all hosts on the network requesting the hardware address of the sought host.

When the sought host (B) receives the broadcast, it recognises the specified layer3 address as its own and subsequently answers with a directed ARP reply to host A. By looking at the source MAC-address of the ARP reply, host A can determine the correct hardware address of host B, and thus the address has been resolved.

Proxy ARP

During the early days of Internet Protocol, not all hosts understood the concept of routing. At that time, a feature of ARP known as proxy ARP was introduced in many routers.

Proxy ARP means that a host or router on a network acts as a proxy for a remote host, and answers ARP requests for that host's IP address. By tricking the local host into believing that the MAC address of the remote host is the router, the local host will direct the traffic meant for the remote host to the MAC address of the router. When the packet arrives at the router, it will forward the packet according to its routing tables.

This feature allows the local host to know nothing about the concept of routing; it will be tricked into believing that virtually all IP addresses have the same MAC-address (that of the router), which is okay as the router normally serves as default gateway for the host.

RFC925 and RFC1027 discuss the concept of proxy ARP.

Controlling ARP Traffic in iBOS

ARP is a fundamental feature of Ethernet-based IP networks.

Static ARP Entries

Normally every host or router learns about end systems through ARP. The information received is used to populate the ARP table, which is later used to resolve the layer2 address when a packet is directed to the layer3 address of that end system.

The network administrator can disable the automatic learning of ARP information by using the **no arp dynamic** interface configuration command. Any ARP message received will be processed, and if the requested layer3 address is the router itself, then the router will answer. However, the router will not populate its ARP table based on this information.

Without correct ARP entries, communications for clients connected on an interface will not work. If dynamic learning of ARP entries has been disabled, they must be configured statically instead. This can be done with the **arp entry** or **interface arp entry** configuration commands.

If it is crucial that only specific clients are allowed to send and receive traffic on an interface, disabling dynamic ARP and configuring specific ARP entry statements for those clients in combination with **policy-map** and **ip strict-clients** configuration commands creates a solution. When the BOOTP/DHCP relay agent is activated with the **ip bootp relay** or **ip dhcp relay** configuration commands, interface ARP entries for hosts assigned addresses with DHCP are automatically created.

Sharing Subnets

In a broadband network, client hosts are connected on several different system interfaces, sometimes even spread over multiple systems. In order to preserve IPv4 address space, it is often a requirement that all clients can be assigned IP addresses from the same IP subnet. In a traditional router, in order for clients to share the same subnet they must be connected to the same broadcast domain. With iBOS this is not required.

The proxy ARP feature in iBOS allows the system to respond to any ARP request for any address that is directly connected on the system interfaces or found in the system routing table. There are two exceptions to this rule. Proxy ARP in iBOS will not respond if the only route matching is the system default route or if the requested destination is located in the same broadcast domain (typically the same vlan or interface) as that from where the request was received.

The iBOS solution allow clients to be connected on different physical and logical interfaces. These interfaces are separate broadcast domains, preventing any layer2 traffic from passing between them. When addresses are assigned to clients in different broadcast domains with DHCP, the DHCP relay agent in iBOS intercepts DHCP packets and creates static ARP entries. The default gateway address of the subnet assigned to clients is configured on the iBOS downstream0 system interface (which thereby in fact is a different broadcast domain). By enabling **arp proxy** on the client interfaces, iBOS will respond to any ARP request for the default gateway address and for any other address found in the system ARP table. This include client addresses assigned on different interfaces.

The requesting client will receive the MAC address of the iBOS system as the hardware address for the requested destination. The client thereby directs all traffic to the destination through the iBOS system allowing it to route the traffic to the correct destination based on the system ARP and routing tables. If the destination client is connected on a different interface on the same iBOS system, it will receive traffic from the source with the iBOS system hardware address as source. Return traffic will yet again go through the iBOS system. Each of the source and destination clients will believe that the iBOS system is the other client.

When **arp proxy** is enabled, iBOS will answer any ARP request for any destination it has in its ARP table or in its routing table. This allows clients on the same and on multiple iBOS systems to be assigned addresses from the same IP subnet as long as a route to the assigned client is created and spread with a routing protocol between the iBOS systems. iBOS contains features to achieve automatic route redistribution of client assigned addresses through the **client** keyword used with interface **arp entry** and **arp entry** and through the **route-map** and **redistribute** routing protocol commands.

ARP Configuration Example

In the following configuration example, iBOS is set up to use the ARP related commands. Two clients are connected on the interfaces fastethernet1 and fastethernet2. They are both assigned addresses from the same IP subnet (192.168.2.0/24) and a static arp entry binds their respective IP address to their MAC address.

The downstream0 interface has the default gateway address used by both clients. It has been assigned to the downstream0 interface automatically by the DHCP relay agent. Dynamic ARP learning is disabled on the client interfaces and proxy ARP is enabled.

The arp entries are flagged as clients and thereby possible to redistribute with OSPF. In the configuration, two host routes for the clients are generated and sent by the system to all OSPF neighbours. This allows other iBOS systems to have the two clients in their routing table and thereby also answer any ARP requests for the 192.168.2.2 and 192.168.2.3 addresses sent by any of their local clients.

Example 67 ARP Configuration Example

```
interface fastethernet1
 arp proxy
 no arp dynamic
 interface arp entry 192.168.2.2 0000.a593.1100 client

interface fastethernet2
 arp proxy
 no arp dynamic
 interface arp entry 192.168.2.3 0000.c759.330e client

interface downstream0
 !volatile ip address 192.168.2.1/24

router ospf
 redistribute kernel route-map clientmatch

route-map clientmatch 1 permit
 match client
```

IPv4 Routing

IPv4 routing is described in the following sections:

- “OSPF v2” on page 154
- “PIM” on page 160
- “IGMP” on page 170

OSPF v2

OSPF is a link-state routing protocol. What this means, is that each router maintains a database with information about the state of all links in the network. As the links change state, the topology of the network changes and the routers recalculate the routing table based on the database.

OSPF is one of the most popular and widely deployed Interior Gateway Protocols (IGP) in large Internet networks.

Understanding OSPF

The following is based on information in section 1.1 of RFC 2328.

The OSPF protocol routes IP packets based on the destination IP address in the packet header. It is a dynamic protocol that quickly detects topological changes in the Autonomous System (AS) and calculates new loop-free routes after convergence. The convergence is short and involves a minimum of routing traffic.

Link State

In a link-state routing protocol, each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each participating router has an identical database. Each individual piece of this database is a particular router's local state (e.g., the router's usable interfaces and reachable neighbours).

The router distributes its local state throughout the Autonomous System through flooding.

Each router contributes its own piece of information about its own interfaces. As this information reaches all other routers, they are able to form a complete picture of the network.

Shortest Path Tree Algorithm

The algorithm used to calculate the routing table is known as the shortest path tree algorithm (or Dijkstra algorithm).

All routers run the exact same algorithm, in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as root. This shortest-path tree gives the route to each destination in the Autonomous System. Externally derived routing information appears as leaves on the tree.

When several equal-cost routes to a destination exist, traffic is distributed equally among them. The cost of a route is described by a single dimensionless metric.

Network Hierarchy

OSPF allows sets of networks to be grouped together. Such a grouping is called an area. The topology of an area is hidden from the rest of the Autonomous System. Hiding this information enables a significant reduction in routing traffic. Also, routing within the area is determined only by the area's own topology, lending the area protection from bad routing data.

Subnets and Authentication

OSPF enables the flexible configuration of IP subnets. Each route distributed by OSPF has a destination and a mask. Two different subnets of the same IP network number may have different sizes (i.e., different masks). This is commonly referred to as variable length subnets. A packet is routed to the best (i.e., longest or most specific) match. A host route is considered to be a subnet with a subnet mask where all bits are set to one (hex 0xffffffff).

All OSPF protocol exchanges are authenticated. This means that only trusted routers can participate in the Autonomous System's routing. A variety of authentication schemes can be used; in fact, separate authentication schemes can be configured for each IP subnet.

Protocol Implementation

The OSPF protocol consists of a number of packets with varying purposes.

Packet Types

The OSPF protocol sends packets to the multicast address 224.0.0.5 (AllSPFRouters) and 224.0.0.6 (AllDRrouters). There are five types of OSPF packets.

Table 18 OSPF Packet Types

Type	Description
1	Hello packet
2	Database description packet
3	Link-state request packet
4	Link-state update packet
5	Link-state acknowledgement packet

Hello Packet

The hello packet is sent to the AllSPFrouters multicast address and used to detect neighbours and form adjacencies. It is also used to elect the designated router (DR) on a broadcast medium such as Ethernet.

Since some of the protocol features can be negotiated, and this is done by hello packets.

Database Description Packet

The Database description packet is used by the OSPF speaking routers to exchange link state database information. During the exchange, one router is master and the other is slave. The master transmits a DBD packet, which the slave may respond to.

Link-state Request Packet

If a router detects that it is missing parts of the database during the database exchange sequence, it can send a Link-State request packet to request the missing part of the database.

This packet may also be sent if the Link state advertisement contained in the database is older than that of the neighbouring router .

Link-state Update Packet

The Link-State Update packet is the response to a link-state request packet. It contains the link-state advertisement (LSA) for one or several networks. Each LSA must be acknowledged by the receiving router.

Link-state Acknowledgement Packet

The Link-State Acknowledgement packet is sent in response to an LSA to acknowledge reception. This makes the exchange process reliable. This packet is sent either to AllSPFRouters or to AllDRrouters, depending on the media. It can also be sent as a unicast message.

Link-state Advertisement

The Link-State Advertisements are the routing information messages flooded throughout the area - and sometimes the network. There are several types of LSAs, containing different information.

Table 19 OSPF LSA Types

LSA Type	Description
1	Router link state
2	Network link state
3	Summary link state type 3
4	Summary link state type 4
5	External link state

The different LSA types serve the following purposes.

Router Link State

Each router generates a router link-state LSA that depicts to the area the router's interfaces and their status. A router LSA is flooded throughout a single area only.

Network Link State

Network LSAs are generated for broadcast and NBMA networks by the designated router. The LSA contains the list of routers connected to the network. The network LSA is flooded throughout a single area only.

Summary Link State

The summary LSA is generated by area border routers (ABR) and contains network summaries for destinations outside the area, but within the AS.

A summary in this case is not the same as an aggregation of the area's subnets. Instead, it implies that the area topology represented by the summary is hidden from routers outside the area. It merely contains a list of destinations reachable within the area, not a list of aggregated subnets found in the area.

External Link State

The external LSA contain routing information about destinations outside the AS. Default routes for the AS can also be AS-external-LSAs.

There are two types of external LSAs:

Type 1

A type 1 external LSA includes the cost of all internal links between the router and the exit point of the AS on its network links.

Type 2

A type 2 external LSA contains only the cost from the local system to the destination.

The difference between type 1 and type 2 allows some destinations to disregard the AS exit point, ignoring the internal network cost of reaching the destination.

Areas

OSPF areas introduce a network hierarchy. There are two levels. OSPF area 0 is the backbone area to which all other areas must be connected.

The area concept means that the exact nature of the topology between a group of routers need not be known to the entire network. Instead, the mere fact that some destinations are reachable in an area of the network is usually enough information for routers in other parts of the network to direct traffic toward that area. Once the packet reaches the area border, the exact topology must be known in order to route the packet to the correct destination.

This concept limits the size and complexity of the link state database and routing calculations. All traffic between areas must pass through the backbone area, area 0. Routers can belong to the backbone area without being directly physically connected to each other. This is achieved by using the virtual links feature, which allows routers to form a virtual connection and exchange backbone routing information. Virtual links have some limitations. Other routers in the area, through which the virtual link passes, must have an understanding of the network topology – they cannot pass through stub, nssa or tsa areas.

Normal Area

A normal area is generally only referred to as an OSPF area. There are no limitations to the kind of LSAs permitted, which can make a normal area vulnerable to flapping external links since redistributed external routing information must then be updated. A virtual link can only pass through a normal area.

Stub Area

A stub area disallows the external-LSAs, making the stub area more tolerant to flapping external routing. The area border routers may inject a default route into the area, making any local router select the closest area border router for all traffic outside the AS. Information about other areas is received through the normal summary LSAs.

Because not all routing information is available to routers within the stub area, they cannot connect to other routers outside the Autonomous System (AS). For the same reason, routers within a stub area cannot be the endpoint or transit of virtual-links.

Totally Stubby Area

A totally stubby area is created when a route filter on all area border routers prevents normal summary LSAs from being announced into the area.

This allows the routers of the totally stubby area to only know of the area's internal routing. If one or multiple area border routers announce a default-route into the area, the area internal routers will still be able to forward traffic out of the area.

The area border routers announce normal summary LSAs for the area networks to the OSPF backbone. One drawback with totally stubby areas is that the area's internal router will select its closest area border router (which announces a default-route) for area external traffic. Because of this, the actual packet path through the network may not take the optimal path.

As with stub areas, routers within the area cannot connect to AS external routers, nor provide transit for virtual-links.

Totally stubby areas are not defined by any RFC. They are a vendor specific extension of the OSPF protocol. Waystream's implementation is compatible with similar implementations by other router vendors. All routers within the totally stubby area are configured as belonging to a normal stub area. To convert the area to a totally stubby area, the area border routers must have a route filter configured. iBOS may operate as such an area border router since it is capable of such route filtering.

Not so Stubby Area (NSSA)

The NSSA area works the same way as a stub area, although some external routes may be redistributed into the area.

Virtual Links

The backbone area (area 0.0.0.0) cannot be disconnected, or some areas in the network will become unreachable. Virtual links offer a method of maintaining connectivity of the backbone area. A virtual link is configured through a non-backbone area (transit area), to make routers reachable through such a part of the backbone area.

The two end-points of a virtual link are area border routers, and both routers must be configured for the virtual link.

Virtual links are used to establish adjacencies between routers in the backbone area. However, they are not used for actual forwarding (i.e. packets forwarded to the adjacent router are not encapsulated). Instead, packets between backbone routers connected with a virtual link will pass through the transit area. Therefore it is not possible to configure a virtual link through a stub area - or any other area where routers lack complete routing of the network.

Configuring OSPF

To activate OSPF on an interface, configure a **network** statement for the IP subnet(s) used on that interface in the **router ospf** context. The area to which the interface shall belong must also be specified.

For each interface it is possible to configure a set of OSPF parameters, including the interval between hello packets, the cost of routes learnt from other OSPF routers on the interface, the priority to become designated router etc.

Any router participating in the OSPF protocol must have a unique router-id which identifies the router in the network. By default, the first active interface found in the system with a configured IP address is selected for router-id.

The router's router-id can also be configured. It is strongly recommended that a manually configured router-id matches an existing system interface. Having two routers with the same router-id in the network will cause problems. Choosing the address of an existing interface reduces the risk of router-id conflicts in the network.

Waystream's implementation of OSPF is based on RFC2328. Some improvements were made to the protocol so that RFC compared to the implementation described in RFC1583. In order to be compatible with routers that have the RFC1583 implementation, the rfc1583compatibility command can be used. The no rfc1583compatibility configuration command can be used if there are no RFC1583-only routers in your network. However, if any router only supports the RFC1583 implementation, leave the rfc1583compatibility flag on.

Area Configuration

To define the type of OSPF area or to configure virtual-links use the **area** configuration command. The required argument is the area-id of the area, or the **backbone** keyword to create the backbone area.

If the area is created as **stub**, optional argument **no-summary** should be used if this system operates as an area border router that should prevent any area external routing information (including summary LSAs) from being announced into the area. This effectively creates a totally stubby area.

OSPF Configuration Example

In the following configuration example, the two gigabitethernet interfaces are configured into area 73. The loopback interface is also configured to be announced in routing updates by this system and is also used as router-id for the system.

In addition, any static ip routes are redistributed into the OSPF protocol as external routes.

Example 68 OSPF Configuration

```
interface loopback0
 ip address 10.0.1.4 255.255.255.255
 no shutdown

interface gigabitethernet1
 ip address 10.71.96.10 255.255.255.252
 no shutdown
 ip ospf hello-interval 1
 ip ospf dead-interval 4

interface gigabitethernet2
 ip address 10.71.96.13 255.255.255.252
 no shutdown
 ip ospf hello-interval 1
 ip ospf dead-interval 4

router ospf
 router-id 10.0.1.4
 redistribute static
 network 10.71.96.8/30 area 0.0.0.73
 network 10.71.96.12/30 area 0.0.0.73
 network 10.0.1.4/32 area 0.0.0.73
```

PIM

The PIM protocol is used to exchange multicast group and rendezvous point (RP) information between routers in a multicast domain.

PIM relies on unicast routing protocols such as OSPF for routing information needed to calculate the best path to the source and thereby build the distribution tree.

PIM version 1 is specified in RFC2362 as an experimental RFC. Version 1 of the protocol has some shortcomings and therefore PIM version 2 is recommended.

Waystream supports PIM Sparse-Mode version 2 as specified in draft-ietf-pim-sm-v2-new-05 and draft-ietf-pim-sm-v2-new-03.

This description of PIM, based on text from the PIMv2 drafts, uses a number of terms to refer to the roles of routers participating in PIM-SM. The following terms have special significance for PIM-SM:

- “Definitions” on page 161
- “PIM Routing” on page 161
- “Finding the RP” on page 164
- “Bootstrap Router Overview” on page 165
- “PIM-SM Overview” on page 165
- “Overview of Bootstrap and RP Discovery” on page 167
- “Overview of Bootstrap and RP Discovery” on page 167
- “Configuring PIM” on page 168

Definitions

The following sections describes a set of common terms in PIM terminology.

Rendezvous Point (RP)

An RP is a router that has been configured for use as the root of the non-source-specific distribution tree for a multicast group. Join messages from receivers for a group are sent to the RP, and data from senders is sent to the RP so that receivers can discover who the senders are and start receiving traffic destined for the group.

Designated Router (DR)

A shared-media LAN like Ethernet may have multiple PIM-SM routers connected to it. If the LAN has directly connected hosts, then only one of these routers, the DR, will act on behalf of those hosts with respect to the PIM-SM protocol. A single DR is elected per interface (LAN or otherwise), using a simple election process.

Multicast Routing Information Base (MRIB)

This is the multicast topology table, which is typically derived from the unicast routing table or routing protocols that carry multicast-specific topology information, such as MBGP. In PIM-SM, the MRIB is used to decide where to send Join/Prune messages. A secondary function of the MRIB is to provide routing metrics for destination addresses. These metrics are used when sending and processing Assert messages.

RPF Neighbour

RPF stands for “Reverse Path Forwarding”. A router’s RPF Neighbour, with respect to an address, is the neighbour that the MRIB indicates should be used to forward packets to that address. In the case of a PIM-SM multicast group, the RPF neighbour is the router that a Join message for that group would be directed to, in the absence of a modifying Assert state.

Upstream

Upstream flows towards the root of the tree. The root of the tree may either be the source or the RP, depending on the context.

Downstream

Downstream flows away from the root of the tree.

PIM Routing

PIM relies on an underlying topology-gathering protocol to populate the MRIB with routes. The routes in this table may be taken directly from the unicast routing table, or they may be different and provided by a separate routing protocol.

Regardless of how it is created, the primary role of the MRIB in the PIM protocol is to provide the nexthop router along a multicast-capable path to each destination subnet. The MRIB is used to determine the nexthop neighbour to which any PIM Join/Prune message is sent. Data flows along the reverse path of the Join messages. Thus, in contrast to the unicast RIB that specifies the nexthop that a data packet would take to

get to a subnet, the MRIB gives reverse-path information, and indicates the path that a multicast data packet would take from its origin subnet to the router that has the MRIB.

Like all multicast routing protocols that implement the service model from RFC 1112, PIM-SM must be able to route data packets from sources to receivers without either the sources or receivers knowing a-priori of the existence of the others. This is essentially done in three phases, although since senders and receivers may come and go at any time, all three phases may occur simultaneously. The three phases are described in:

- "Phase One: RP Tree" on page 162
- "Phase Two: Register Stop" on page 163
- "Phase Three: Shortest Path Tree" on page 163

Phase One: RP Tree

In phase one, a multicast receiver expresses its interest in receiving traffic destined for a multicast group. Typically it does this by using IGMP, but other mechanisms might also serve this purpose.

One of the receiver's local routers is elected as the Designated Router (DR) for that subnet. On receiving the receiver's expression of interest, the DR then sends a PIM Join message towards the RP for that multicast group.

Joining and Pruning

This Join message is known as a (*,G) Join, because it joins group G for all sources to group G. The (*,G) Join travels hop-by-hop towards the RP for the group, and in each router it passes through, a multicast tree state for group G is instantiated.

Eventually the (*,G) Join either reaches the RP or reaches a router that already has (*,G) Join state for that group. When many receivers join the group, their Join messages converge on the RP and form a distribution tree for group G that is rooted at the RP. This is known as the RP Tree (RPT), and is also known as the shared tree, because it is shared by all sources sending to that group.

Join messages are re-sent periodically, as long as the receiver remains in the group. When all receivers on a leaf-network leave the group, the DR will send a PIM (*,G) Prune message towards the RP for that multicast group. However, if for any reason the Prune message is not sent, the state will eventually time out.

Source Sending Data

When a multicast data sender starts sending data destined for a multicast group, the sender's local router (DR) takes those data packets, unicast-encapsulates them, and sends them directly to the RP.

The RP receives these encapsulated data packets, decapsulates them, and forwards them on to the shared tree. The packets then follow the (*,G) multicast tree state in the routers on the RP Tree, are replicated wherever the RP Tree branches, and eventually reach all the receivers for that multicast group.

The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are known as PIM Register packets.

At the end of phase one, multicast traffic is flowing encapsulated to the RP, and then natively over the RP tree to the multicast receivers.

Phase Two: Register Stop

Register-encapsulation of data packets is inefficient for two reasons:

- Encapsulation and decapsulation may be relatively expensive operations for a router to perform, depending on whether or not the router has the appropriate hardware for these tasks.
- Travelling all the way to the RP and then back down the shared tree may entail travelling a relatively long distance to reach receivers that are close to the sender. For some applications, this increased latency is undesirable.

Although Register-encapsulation may continue indefinitely, for the reasons above the RP will normally choose to switch to native forwarding.

To do this, the RP will normally initiate an (S,G) source-specific Join towards S when it receives a register-encapsulated data packet from source S on group G. This Join message travels hop-by-hop to S, instantiating (S,G) multicast tree state in the routers along the path. (S,G) multicast tree state is only used to forward packets for group G if those packets come from source S.

Eventually, the Join message reaches S's subnet or a router that already has (S,G) multicast tree state, and then packets from S start to flow along the (S,G) tree state towards the RP. These data packets may also reach routers with (*,G) state along the path towards the RP - if so, they can short-cut onto the RP tree at that point.

Register Stop Message

While the RP is in the process of joining the source-specific tree for S, encapsulated data packets will continue to the RP. When packets from S also start arriving natively at the RP, the RP will then be receiving two copies of each of these packets. At this point, the RP starts discarding the encapsulated copy of these packets and sends a Register Stop message back to S's DR to prevent the DR from encapsulating the packets unnecessarily.

At the end of phase 2, traffic will be flowing natively from S along a source-specific tree to the RP, and from there along the shared tree to the receivers. Where the two trees intersect, traffic may transfer from the source-specific tree to the RP tree, thus avoid taking a long detour via the RP.

It should be noted that a sender may start sending before or after a receiver joins the group, and thus phase two may take place before the shared tree to the receiver is built.

Phase Three: Shortest Path Tree

Although having the RP rejoin the source removes the encapsulation overhead, it does not completely optimise the forwarding paths. For many receivers the route via the RP may involve a significant detour when compared with the shortest path from the source to the receiver.

To obtain lower latencies, a router on the receiver's LAN, typically the DR, has the option to initiate a transfer from the shared tree to a source-specific shortest-path tree (SPT). To do this, it issues an (S,G) Join towards S. This instantiates state in the routers along the path to S.

Eventually, this join either reaches S's subnet or a router that already has (S,G) state. When this happens, data packets from S start to flow along the (S,G) state until they reach the receiver.

Multicast Traffic from Two Directions

At this point the receiver (or a router upstream of the receiver) will be receiving two copies of the data - one from the SPT and one from the RPT. When the first traffic starts to arrive from the SPT, the DR or upstream router starts dropping the packets for G from S that arrive via the RP tree.

In addition, it sends a (S,G) Prune message towards the RP. This is known as a (S,G,rpt) Prune. The Prune message travels hop-by-hop, instantiating state along the path towards the RP, indicating that traffic from S for G should NOT be forwarded in this direction. The prune is propagated until it reaches the RP or a router that still needs the traffic from S for other receivers.

By now, the receiver will be receiving traffic from S along the shortest-path tree between the receiver and S. The RP is also receiving the traffic from S, but this traffic is no longer reaching the receiver along the RP tree. As far as the receiver is concerned, this is the final distribution tree.

Finding the RP

The rendezvous point (RP) is crucial to the PIM protocol. Any router participating in the PIM protocol must know the IP address of the RP.

PIM offers two methods for finding out about the RP.

Static RP

A router can have static configuration for one or multiple RPs. This is a configuration statement that simply tells the router to which RP register messages and joins should be sent for a particular range of multicast groups.

Bootstrap Router

The bootstrap router (BSR) is an improvement of PIM where one or more routers in the network act as a bootstrap router, regularly flooding information throughout the network about available RPs and their group ranges.

Any router announcing its candidacy as RP must announce it to the bootstrap routers in the network using candidate-RP messages.

Bootstrap Router Overview

The following information is from the IETF draft <draft-ieft-pim-sm-bsr-02.txt>.

For correct operation, every PIM Sparse-mode router within a PIM domain must be able to map a particular global-scope multicast group address to the same RP. If this is not the case then black holes may appear where some receivers in the domain cannot receive some groups. A domain in this context is a contiguous set of routers that all implement PIM and are configured to operate within a common boundary defined by PIM Multicast Border Routers (PMBRs). PMBRs connect each PIM domain to the rest of the Internet.

A PIM domain may also broken up into multiple administrative scope regions - these are regions where a border has been configured so that a range of multicast groups will not be forwarded across that border. For more information on Administratively Scoped IP Multicast, see RFC 2365. The modified criteria for admin-scoped regions are that the region is convex with respect to forwarding based on the MRIB, and that all PIM routers within the same scope region map a particular scoped group to the same RP within that region.

The PIM-SM specification does not mandate the use of a single mechanism to provide routers with the information needed to perform the group-to-RP mapping.

PIM-SM Overview

Every PIM-SM multicast group needs to be associated with the IP address of a Rendezvous-Point (RP). When a new multicast sender starts sending, its local Designated Router (DR) will encapsulate these data packets in a PIM Register message and send them to the RP for that multicast group. When a new multicast receiver joins, its local DR will send a PIM Join message to the RP for that multicast group. When any PIM router sends a (*,G) Join message, it needs to know which is the nexthop router towards the RP for G to send the message to. Also, when a PIM router is forwarding data packets using (*,G) state, it needs to know which is the correct incoming interface for packets destined for G, since it needs to reject any packets that arrive on other interfaces. Thus it is important for all the PIM routers in a domain to be able to map each multicast group to the correct RP address.

There are a number of ways that group-to-RP mapping can be done. The simplest solution is for all the routers in the domain to be configured with the same information. However, static configuration generally doesn't scale well, nor does it dynamically adapt to route around router or link failures. The mechanism specified here is known as the PIM BootStrap Router mechanism, or BSR for short, and it provides a dynamic, adaptive mechanism for distributing group-to-RP mapping information rapidly throughout a domain.

PIM Rules for Mapping

Irrespective of how it obtains the mapping information, a PIM-SM router will have a mapping table that contains multiple entries, each of which has the following form:

- Multicast group range, expressed as an address and prefix length.
- RP Priority.
- IP address of RP.

In general, these mapping entries may overlap in arbitrary ways; a particular multicast group may be covered by multiple mapping entries. When this is the case, the router chooses only one of the entries by applying a deterministic algorithm (specified in the PIM-SM protocol specification), so that all routers in the domain choose the same entry and hence apply the same group-to-RP mapping.

PIM-SM BSR Mechanism

The BSR mechanism provides a way in which viable group-to-RP mappings can be created and distributed to all the PIM-SM routers in a domain. It is adaptive in that if an RP becomes unreachable, this will be detected, and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

The general idea behind the BSR-mechanism is that some of the PIM routers within a PIM domain are configured to be potential RPs for the domain. These are known as candidate-RPs (C-RPs). A subset of the C-RPs will eventually be used as the actual RPs for the domain. In addition, some of the PIM routers in the domain are configured as candidate bootstrap routers (C-BSRs). One of these C-BSRs will be elected to serve as the bootstrap router (BSR) for the domain, and all the PIM routers in the domain will learn the result of this election through Bootstrap messages. The C-RPs will then report their candidacy to the elected BSR, which will choose a subset of the C-RPs to form the actual set of RPs to be used. This RP-Set will then be distributed to all the routers in the domain through Bootstrap messages.

Administratively Scoped Regions

The mechanism is made slightly complicated by the presence of administratively-scoped multicast regions within the PIM-SM domain. An admin-scope region is a convex connected set of PIM routers, surrounded by an admin-scope boundary. The boundary specifies a range of multicast addresses that will not be forwarded into -or out of - the scoped region.

This complicates BSR, because a PIM router within the scoped region shall not use an RP outside the scoped region (or vice-versa). This is done by electing a BSR for every admin-scope region within a PIM domain, as well as a global BSR for the entire PIM domain.

C-RPs typically register multiple times; once to each BSR in every admin scope zone the C-RP is in.

Within each scope zone, the BSR for that zone acts similarly to the global BSR for the entire domain.

BSR Mechanism Phases

There are four basic phases in the BSR mechanism (although in practise all phases may occur simultaneously):

- *BSR Election:* Each Candidate-BSR originates bootstrap messages (BSMs). Every BSM contains a BSR priority field. Routers within the domain flood the BSMs throughout the domain. If a C-BSR hears about C-BSR with a higher-priority than itself, it then suppresses the sending of further BSMs for a period of time. The single remaining C-BSR becomes the elected BSR, and its BSMs inform all the other routers in the domain that it is the elected BSR.

- *C-RP Advertisement:* Each Candidate-RP within a domain sends periodic Candidate-RP-Advertisement (C-RP-Adv) messages to the elected BSR. In this way, the BSR learns about potential RPs that are currently up and reachable.
- *C-RP-Set Formation:* In order to form the RP-Set, the BSR selects a subset of the C-RPs that it has heard C-RP-Adv messages from. In general it should do this in such a way that the RP-Set is neither too large to inform all the routers in the domain about, nor so small that load is overly concentrated on some RPs. It should also attempt to produce an RP-Set that does not change frequently.
- *RP-Set Flooding:* In future bootstrap messages, the BSR includes the RP-Set information. Since bootstrap messages are flooded rapidly through the domain, this ensures that the RP-Set rapidly reaches all the routers in the domain. BSMs are originated periodically to ensure consistency after failure restoration.

Overview of Bootstrap and RP Discovery

A small set of routers from a domain are configured as candidate bootstrap routers (C-BSRs) and, through a simple election mechanism, a single BSR is selected for that domain.

A set of routers within a domain are also configured as candidate RPs (C-RPs); typically these will be the same routers that are also configured as C-BSRs.

Candidate RPs periodically unicast Candidate-RP-Advertisement messages (C-RP-Advs) to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, as well as an optional list of group addresses and mask length fields, indicating the group prefix(es) for which the candidacy is advertised. The BSR then includes a set of these Candidate-RPs (the RP-Set), along with the corresponding group prefixes in Bootstrap messages it periodically originates. Bootstrap messages are distributed hop-by-hop throughout the domain.

All the PIM routers in the domain receive and store Bootstrap messages originated by the BSR. When a DR receives an indication of local membership (typically from IGMP or MLD) or a data packet from a directly connected host of a group for which it has no forwarding state, the DR uses a hash function to map the group address to one of the C-RPs from the RP-Set whose group-prefix includes that group. If the local host joined the group, the DR then sends a Join message in the direction of that RP. If the local host sent a packet to the group, the DR Register-encapsulates and unicasts the data packet to the RP.

A Bootstrap message indicates if the RPs included in the message are still alive. If an RP is included in the message, then it is tagged as ‘up’ by the routers; RPs not included in the message are removed from the list of RPs over which the hash algorithm acts. Each router continues to use the contents of the most recently received Bootstrap message until it accepts a new Bootstrap message.

If a PIM domain becomes partitioned, each area separated from the old BSR will elect its own BSR, which will then distribute an RP-Set containing RPs that are reachable within that partition. When the partition heals, another election will occur automatically and only one of the BSRs will continue sending out Bootstrap messages. As is expected, some disruption in packet delivery may occur at the time of a partition or healing.

Administratively Scoped Multicast and BSR

Administratively Scoped IP Multicast, as defined in RFC 2365, allows a network provider to configure scope boundaries at multicast routers.

Such a scope boundary consists of a range of multicast addresses (expressed as an address and mask) that the router will not forward across the boundary. For correct operation, such a scope zone border must be complete and convex. This means that there must be no path from inside the scoped zone to outside of it, unless it passes through a configured scope border router. The multicast capable path between any arbitrary pair of multicast routers in the scope zone must also remain in the zone.

In order to function correctly with admin scoping, there must be a BSR and at least one C-RP within every admin scope region. Admin scope zone boundaries must be configured at the Zone Border Routers (ZBRs), since they need to filter PIM Join messages that might inadvertently cross the border due to error conditions. In addition, at least one C-BSR within the admin scope zone must be configured to be a C-BSR for the admin scope zone's address range.

A separate BSR election will then take place (using bootstrap messages) for every admin scope range (as well as one for the global range). Admin scope ranges are identified in the bootstrap message because the group range is marked (using the “Admin Scope” bit, previously a “Reserved” bit) to indicate that this is an administrative scope range, and not just a range that a particular set of RPs are configured to handle.

These admin scope bootstrap message packets are flooded in the normal way, but will not be forwarded by another ZBR across the boundary of that scope zone (see Section 3.3 for the specifics of this). It is not required that C-RPs within the scope zone be configured to know about the scope zone, since they can learn of its existence from bootstrap messages.

Unless configured otherwise, C-RPs discover the existence of the admin scope zone and its group range by receiving a bootstrap message from the scope zone's elected BSR. This BSR contains the scope zone's group-range and is marked using the "Admin Scope" bit. A C-RP stores each elected BSR's address and the admin scope range contained in its bootstrap message. It separately unicasts Candidate-RP-Advertisement messages to the appropriate BSR for every admin scope range within which it is willing to serve as an RP.

All PIM routers within a PIM bootstrap domain where admin scope ranges are in use must be capable of both receiving bootstrap messages and of storing the winning BSR and RP-set for all admin scope zones that apply. Thus PIM routers that only implement RFC 2362 (which only allows one BSR per domain) cannot be used in PIM domains where admin scope zones are configured.

Configuring PIM

PIM configuration is split into interface configuration and root context configuration.

Interface Configuration

The current implementation supports only PIM sparse-mode. To activate an interface in the PIM domain, configure **ip pim sparse-mode** in interface configuration context.

There are a number of parameter variables available in PIM. Under interface

configuration, parameters for priority to become designated router on the network, interval between PIM hellos among others, can be configured.

Root Context

Configuration statements for RP and other sparse mode related settings are available in the root context.

To configure static RPs for this system, use the **ip pim rp-address** configuration command.

If the network has a bootstrap router, information about available RPs will reach the router via other routers closer to the BSR. In this case, there is no need for static RP configuration.

If this router will act as bootstrap router, use the **ip pim bsr-candidate** configuration command to set the interface used as source for bootstrap messages.

If this router will act as a candidate-RP for a group range, use the **ip pim rp-candidate** configuration command to set interface used as RP.

Shortest Path Switch

An important part of PIM is to decide when to switch to the shortest path tree. In iBOS, this can be done when the first packet is seen over the shortest path to the source. To activate this functionality use the **ip pim spt-threshold** configuration command.

PIM Configuration Example

In the following example, the router has been set up as candidate-RP for the group 227.1.5.12 and as a potential BSR in the network with priority 100. The system is configured to switch immediately to the shortest path tree.

Example 69 PIM Configuration

```
interface vlan10
 ip address 192.168.5.1 255.255.255.0
 ip pim sparse-mode

interface vlan11
 ip address 10.1.15.1 255.255.255.252
 ip pim sparse-mode

interface vlan12
 ip address 10.1.15.5 255.255.255.252
 ip pim sparse-mode

interface loopback0
 ip address 10.5.1.1 255.255.255.255
 ip pim sparse-mode

ip pim spt-threshold
ip pim rp-candidate loopback0
ip pim bsr-candidate loopback0
```

IGMP

The purpose of the IGMP protocol is to enable hosts to report their interest in one or multiple multicast groups to their neighbouring systems, so that any listening router can establish a distribution tree for that group.

IGMP is specified in RFC791 (September 1981), and is part of Internet Standard 5, The Internet Protocol. The protocol has been updated through RFC1112. Version 2 of the protocol was defined RFC2236. Version 3 is defined in RFC3376.

Waystream supports IGMP version 2 and version 3.

Multicast routers use IGMP to learn which groups have members on each of their attached networks.

A multicast router keeps a list of multicast group memberships for each attached network, and a timer for each membership. “Multicast group memberships” means the presence of at least one member of a multicast group on a given attached network, not a list of all the members.

Querier

With respect to each of its attached networks, a multicast router may assume one of two roles: Querier or Non-Querier. There is normally only one Querier per physical network. All multicast routers start up as a Querier on each attached network, but if a multicast router hears a Query message from a router with a lower IP address, it must become a Non-Querier on that network. If a Non-Querier has not heard from the Querier for some time, it may assume the role of Querier on the network.

Routers periodically send a General Query on each attached network for which the router is the Querier, in order to solicit membership information. On start-up, a router should send a number of General Queries spaced closely together in order to quickly and reliably determine membership information.

General Query

A General Query is addressed to the all-systems multicast group (224.0.0.1), and has a Group Address field of 0. The Max Response Time protocol field dictates how long any multicast receiving host has to report back before the router expires the group on the network.

When a host receives a General Query, it sets delay timers for each group (excluding the all-systems group) of which it is a member, on the interface from which it received the query. Each timer is set to a different random value, using the highest clock granularity available on the host and selected from the range (0, Max Response Time) with Max Response Time as specified in the Query packet.

If a host is a member of a group on a given interface, and the host receives a Group-Specific Query for that Group, the host will set a random timer (ranging from 0 to Max Response Time) for the queried group, as long as the query was received on the same interface. If a timer for the group is already running, it is reset to the random value only if the requested Max Response Time is less than the remaining value of the running timer.

Membership Report

When a group's timer expires, the host multicasts a Version 2 Membership Report to the group, with an IP TTL of 1. If the host receives another host's Report (version 1 or 2) while it has a timer running, it stops its timer for the specified group and does not send a Report, thus avoiding duplication of Reports.

When a router receives a Report, it adds the reported group to the list of multicast group memberships on the network on which it received the Report, and sets the timer for the membership to Group Membership Interval. Repeated Reports refresh the timer. If no Reports are received for a particular group before this timer has expired, the router assumes that the group has no local members and that it need not forward remotely-originated multicasts for that group onto the attached network.

Joining Groups

When a host joins a multicast group, it should immediately transmit an unsolicited Version 2 Membership Report for that group, in case it is the first member of that group on the network. In case the initial Membership Report is lost or damaged, it is recommended that it be repeated once or twice after a short delay. (A simple way to accomplish this is to send the initial Version 2 Membership Report and then act as if a Group-Specific Query was received for that group, and set a timer accordingly).

Leaving Groups

If a host was the last to reply to a Query with a Membership Report for a multicast group, then when the host leaves that group it should send a Leave Group message to the all-routers multicast group (224.0.0.2).

If it was not the last host to reply to a Query, it need not send anything, as there must be another member on the subnet. This is an optimisation that reduces traffic. A host without sufficient storage to remember whether or not it was the last host to reply, may always send a Leave Group message when it leaves a group.

Leave Group messages are addressed to the all-routers group, because other group members do not need to know that a host has left the group. However, it does no harm to address the message to the group.

Routers should accept a Leave Group message addressed to the group being left in order to accommodate implementations of an earlier version of the IGMP standard.

Router Response to Leave Group

When a Querier receives a Leave Group message for a group that has group members on the reception interface, the Querier sends a set of Group-Specific Queries every Last Member Query Interval to the group being left. These Group-Specific Queries have their Max Response time set to Last Member Query Interval. If no Reports are received after the last query response time expires, the routers assume that the group has no local members, as above.

Any Querier to Non-Querier transition is ignored during this time; the same router keeps sending the Group-Specific Queries.

Non-Queriers must ignore Leave Group messages, and Queriers should ignore Leave

Group messages for which there are no group members on the reception interface.

When a Non-Querier receives a Group-Specific Query message, it sets its group membership timer to Last Member Query Count multiplied by the Max Response Time specified in the message if its existing group membership timer is greater than that value.

Changes to Version 1

An important difference between IGMP version 1 and IGMP version 2 is the leave group message. In IGMP version 1, there was no explicit method of making the router aware that there were no more listeners to a multicast group on an interface. Instead, the router had to wait a specified amount of time after seeing the last Membership Report for the group before the group could be removed.

In a network with multiple multicast groups, where hosts frequently changed groups (such as television streams over multicast and channel zapping), the interface to the end host would be forwarding multiple multicast groups - even if no listeners existed.

Another change from version 1 is IGMP snooping. IGMP snooping refers to the technique of intercepting IGMP messages to determine where listening hosts are connected, and to which groups they listen. This technique is commonly implemented in layer2 switches in order to forward multicast frames on only those ports where there are downstream listeners for each particular group.

Switches without this feature must forward multicast frames on all ports associated with the Vlan, which means that hosts who are not listening may be overwhelmed with multicast traffic.

A Waystream router may have multiple ports assigned to the same vlan. Internally the different ports behave much like the layer2 switch ports described above. IGMP snooping can therefore be activated to make iBOS intercept IGMP messages from hosts, thereby creating an optimal forwarding table for multicast groups. This feature is activated using the **ip igmp snooping** interface configuration command on the vlan interface.

When enabled, the IGMP snooping feature supports IGMP version 2 and version 3. Any version 1 IGMP packets are dropped since version 2 is the minimum requirement for efficient multicast forwarding.

IGMPv3 Message Format

IGMPv3 updates the IGMPv2 protocol and is specified in RFC 3376. The main addition is the ability to join and leave multicast groups with one or more sources specified. Sources can be explicitly included, or implicitly included using exclude lists.

To avoid possible denial of service attacks, the rate of igmp messages sent to IBOS, the number of groups allowed to be joined per interface, and the number of sources in the source list per group are limited. The rate limit is 25 packets per second, number of groups joined per interface is limited to 64, and at most 16 sources can be specified per group.

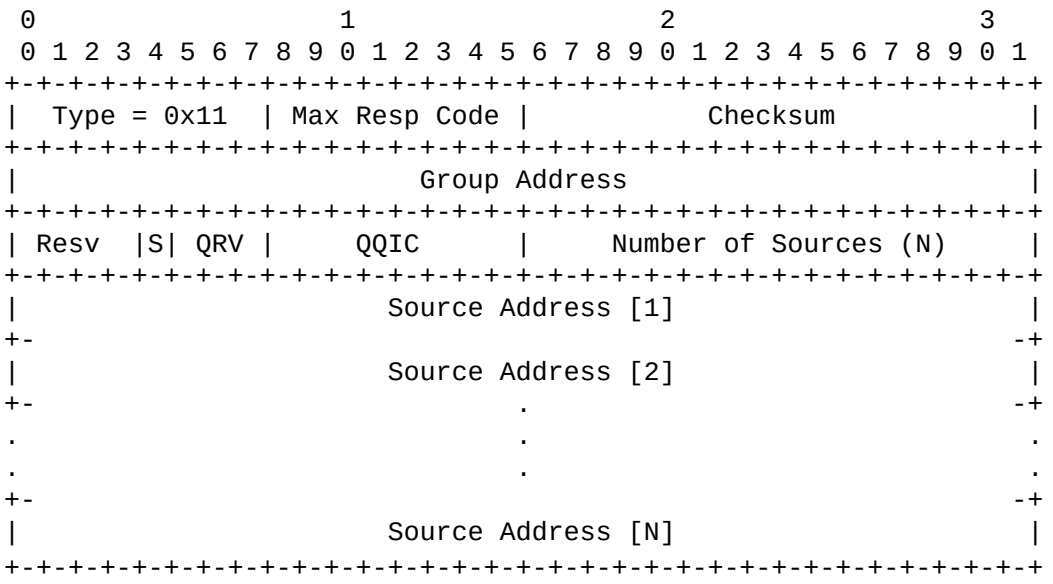
The IGMP Queries are extended with group and source specific queries to determine when a (S,G) multicast forwarding state can be pruned.

Whenever an IGMPv2 report is seen, IGMPv3 routers operate in an IGMPv2 compatibility mode for a LAN segment.

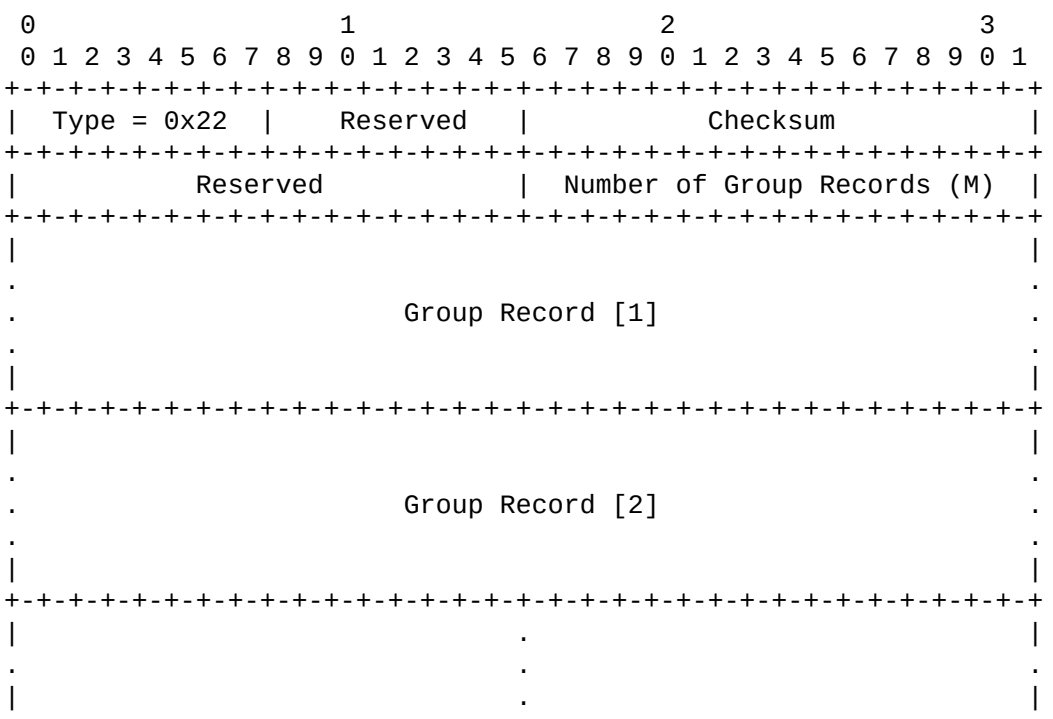
When an IGMPv3 client sees a IGMPv2 query, it switches to IGMPv2 mode to maintain compatibility with v2 queries. After the querier switches from v2 to v3 mode, a certain timeout must pass before the clients also switch to v3 mode.

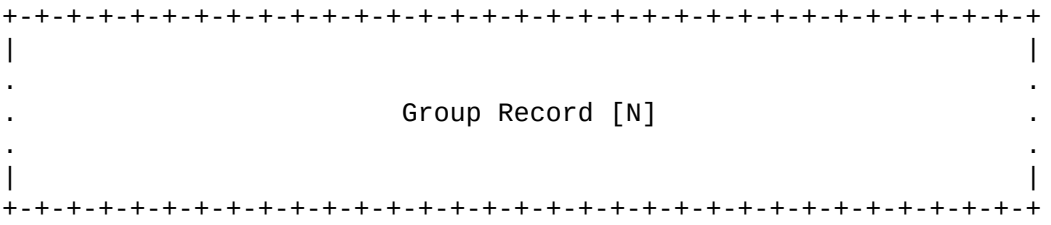
Packet Formats for IGMPv3 Messages

Example 70 IGMPv3 Message Packet Format - Query

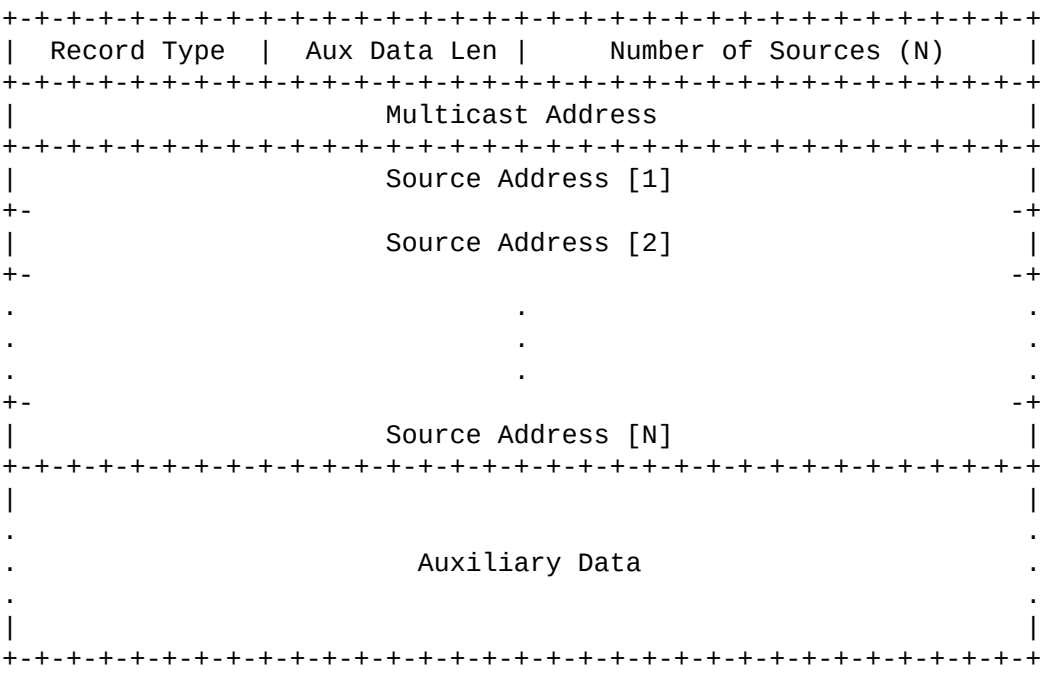


Example 71 Membership Report Message Format





where each Group Record has the following format:



IGMPv3 Snooping

VLAN interfaces configured with **ip igmp snooping** as well as downstream mapped client interfaces are snooped. The snooper supports IGMPv3.

SSM Mapping and IGMP Reports

The SSM range (multicast group addresses 232.0.0.0/8) maps specific sources for multicast groups within the range used for IGMPv2 joined clients. IGMPv3 clients can specify sources that differ from the configured SSM mapping using explicit include on (S,G).

Extended Output

The output from **show ip igmp snooper groups** shows the IGMP version and the source filter lists that are active for each snooped interface.

Example 72 Output for IGMPv3 Snooper

```
ibos# show ip igmp snooper groups
Snooped groups on all interfaces
 V  Group          Snooped Interface                   Age  MisRprt
                        Reporter/Source
 ------------------------------------------------------------------
----------
 3  232.22.22.2     fastethernet7                    18h30m43s      0
                    10.54.7.2                       16h20m19s      0
                    include 8.8.8.8                 16h20m19s      0
                    include 10.54.8.2               16h20m19s      0
 3  232.22.22.2     fastethernet6                    18h30m51s      0
                    10.54.6.2                        1h30m30s      0
                    include 1.1.1.1                  1h30m30s      0
                    include 10.54.8.23               1h30m30s      0
```

IPv4 Forwarding

iBOS support for IPv4 forwarding is described in:

- "GRE Tunnels" on page 176
- "Virtual Router Redundancy Protocol (VRRP)" on page 177
- "IP MTU Size" on page 178
- "DNAT Forwarding" on page 179

GRE Tunnels

Generic Routing Encapsulation (GRE) tunnels are a simple form of tunnels in which the IP packet is encapsulated into another IP packet before transmitted. A tunnel interface is created at both endpoints of the tunnel. Another local interface IP address is used as the tunnel source in the encapsulating IP packet header when transmitting packets. The same address must be used as destination in the encapsulating IP packet header when receiving packets. The tunnel interface is configured with the source and destination IP address to be used in the encapsulating IP packet header.

The GRE tunnel itself is configured with an IP address as any normal layer3 interface. The tunnel behaves as a point-to-point link and normal routing applies to traffic to be sent through the tunnel.

Any intermediate device between the source and destination systems will route based on the encapsulating IP header. Only when the tunnel packet has reached its destination the encapsulating header is removed. The tunnel endpoint router then performs normal forwarding operation on the decapsulated IP packet.

Since the tunnel packet will be larger than the encapsulated packet MTU issues and fragmentation may be required. See the section About fragmentation above.

To create a GRE tunnel, use the **tunnel mode** configuration command in interface tunnel context to set the tunnel mode to GRE.

Then use the **gre** command in tunnel interface context to set up the destination IP address of the other end of the tunnel to which all packets routed into the tunnel should be forwarded. Also specify the **source-interface** in the local system from which to pick the IP address to use as source address in the encapsulating IP packet header. The primary address of the interface will be selected.

The IP address of the tunnel can then be configured.

Example 73 Creating a GRE Tunnel

```
# Configuration on ASR1
asr1# configure terminal
asr1(config)# interface loopback1
asr1(config-if-lo)# ip address 192.121.165.1 255.255.255.255
asr1(config-if-lo)# ip primary 192.121.165.1
asr1(config-if-lo)# no shutdown
asr1(config-if-lo)# exit
asr1(config)# interface tunnel1
asr1(config-if-tun)# tunnel mode gre
```

```
asr1(config-if-tun)# gre 192.121.166.1 source-interface loopback1
asr1(config-if-tun)# ip address 192.168.1.1 255.255.255.252

# Configuration on ASR2

asr2# configure terminal
asr2(config)# interface loopback1
asr2(config-if-lo)# ip address 192.121.166.1 255.255.255.255
asr2(config-if-lo)# ip primary 192.121.166.1
asr2(config-if-lo)# no shutdown
asr2(config-if-lo)# exit
asr2(config)# interface tunnel1
asr2(config-if-tun)# tunnel mode gre
asr2(config-if-tun)# gre 192.121.165.1 source-interface loopback1
asr2(config-if-tun)# ip address 192.168.1.2 255.255.255.252
```

In the example above, a GRE tunnel is created between the two routers asr1 and asr2. The tunnel uses the 192.168.1.0/30 subnet. A loopback interface is used on each router as the local endpoint of the tunnel and the IP address of that interface is configured as the tunnel destination on the other router.

Routing protocols must be set up to carry routing information for the tunnel endpoints (loopback interfaces) so that each router and all intermediate routers have a valid forwarding entry for the tunnel endpoint addresses.

Virtual Router Redundancy Protocol (VRRP)

VRRP is an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP configured routers on a LAN. The advantage gained from using VRRP is that it provides a high-availability default path without having to configure dynamic routing, or router discovery protocols, on every end-host.

A virtual router must use 00-00-5E-00-01-XX as its Media Access Control (MAC) address. The last byte of the address (XX) is the Virtual Router IDentifier (VRID), which is different for each virtual router in the network. This address is used by only one physical router at a time, and it will reply with this MAC address when an ARP request is sent for the virtual router IP address. Physical routers within the virtual router must communicate within themselves using packets with multicast IP address 224.0.0.18 and IP protocol number 112.

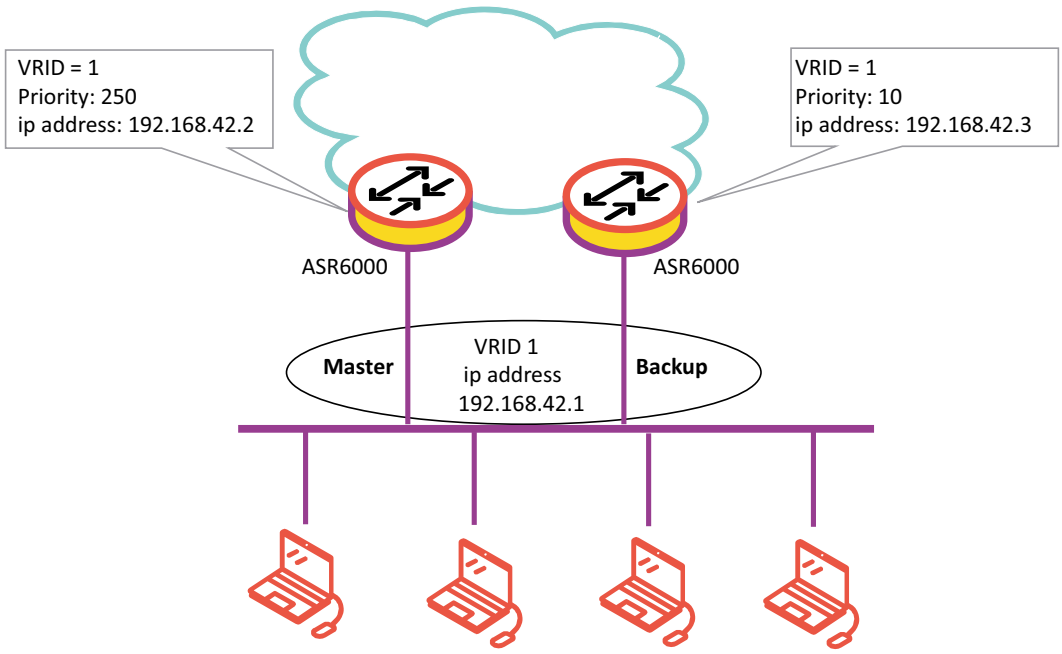
Routers have a priority of between 1-255 and the router with the highest priority will become the master. This is useful when a master router is to be withdrawn, as its priority can be lowered so that a backup router pre-empts the master router status, instead of waiting for the hold time to expire. This use of priority to reassign the master status reduces the black hole period.

The VRRP router controlling the IP address(es) associated with a virtual router is called the ***Master***, and forwards packets sent via these IP addresses. An election process is used to provide dynamic failover for forwarding when the current master router becomes unavailable. Any of the virtual router IP addresses on the LAN can be used as the default first hop router by end-hosts.

A failure to receive a multicast packet from the master router for a period longer than three times the configured advertisement timer causes the backup routers to assume that the master router is dead. The virtual router then transitions into an unsteady state

and an election process is initiated to select the next master router from the backup routers. This is done using multicast packets. The backup router with the highest priority becomes the master router by raising its priority above that of the current master. It will then take responsibility for routing packets sent to the virtual gateway MAC address. In cases where backup routers all have the same priority, the backup router with the highest IP address becomes the master router.

Figure 30 VRRP Example



VRRP is implemented in compliance with RFC 3768.

Note: *When* **ip vrrp accept** *is enabled, VRRP servers can also accept ICMP ping requests.*

Commands used to configure VRRP are:

- **ip vrrp address**
- **ip vrrp priority**
- **ip vrrp advertise-interval**
- **ip vrrp prohibit-preemption**
- **ip vrrp shutdown**
- **show ip vrrp**

IP MTU Size

Support for IP packets larger than 1500 bytes can be enabled. One practical use for larger IP MTU is to avoid fragmenting tunnelled full-size ethernet frames in L2TPv3 tunnels. The maximum IP MTU of 1600 bytes is implemented.

1600 bytes IP MTU is supported on MPC480. Extending this MTU size allows Layer2 VPNs to be used over L2TPv3 tunnels with no fragmentation.

Note: *Jumbo frames is supported by ASR6000 and MS4000 and the maxiumum MTU is 9198 bytes*

The maximum accepted ethernet frame size for bridged traffic received on gigabitethernet interfaces has been increased from 1532 to 1638 bytes. Larger ethernet frames will be dropped. iBOS will need to fragment routed IP packets larger than configured as the IP MTU on the egress interface. For carrying ethernet frames unfragmented in an L2TPv3 tunnel the IP MTU must be set high enough to allow for 1546 bytes tagged ethernet on the egress interface carrying the L2TPv3 tunnel data. The L2TPv3 tunnel adds 20 bytes IPv4 header and 12 bytes L2TPv3 header to the received ethernet frame (1514 + 20 + 12 = 1546).

Note: *The IP MTU settings must be the same on both routers sharing a link to get OSPF neighbourship up.*

The command is used to configure MTU size:

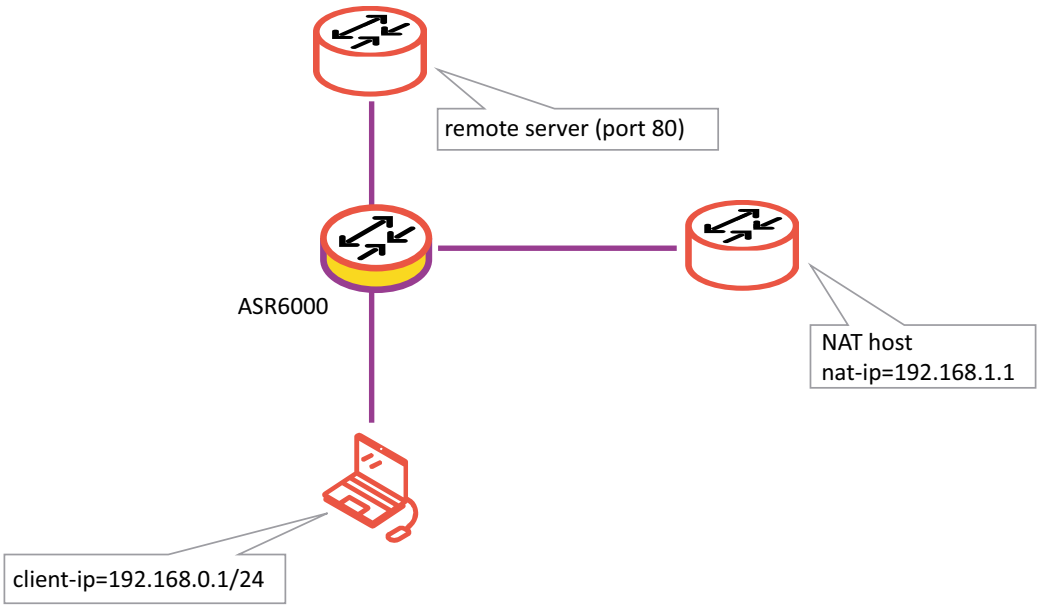
- **ip mtu**

DNAT Forwarding

The destination network address translation (DNAT) feature allows an internal host to handle requests to external IP addresses. This feature can be used on captive portals to redirect users that lack services to an information page on a web server.

Destination NAT uses a cache to identify and rewrite return traffic packets which allows bidirectional communication to be triggered by the client IP.

Figure 31 Destination NAT



Commands used to configure destination NAT are:

- **policy ip destination-nat**
- **clear ip destination-nat sessions**
- **ip destination-nat session-timeout**
- **show ip destination-nat sessions**
- **debug ip destination-nat sessions**

IPv6

The IPv6 protocol replaces IPv4. There are a lot of similarities but there are also fundamental differences.

This chapter aims to provide a short technical description on the fundamental parts of IPv6 implementation in iBOS. The informations presented here is focused on what is important for Waystream products deployed in Ethernet based broadband networks.

The following is described:

- "IPv6 Overview" on page 182
- "IPv6 Addressing" on page 183
- "IPv6 Routing" on page 204
- "IPv6 Forwarding" on page 205

For full details on IPv6, refer to the RFC directly.

IPv6 Overview

IPv6 addresses are 128 bits long (IPv4 used 32 bits). The IPv6 subnet is generally /64, which means that half of the 128 bits are the host part of the address.

There are also two types of address groups in IPv6. Link local addresses, which are intended to be used only between hosts in the same broadcast domain, and global addresses which are needed to communicate on the Internet.

Link local IPv6 addresses start with FE80::/10. Link local communication is used for a lot of signalling between routers and hosts, for example the Neighbour Discovery protocol (replaces ARP from IPv4). When IPv6 is enabled on a host interface, the host constructs a link-local address from its MAC address and the standardized FE80::/10 prefix. The address is then verified to be unique using Duplicate Address Detection.

Note: *Another method for generating the link local address is to use a randomized lower 64 bits for the host part of the address. This method is specified in RFC 4941.*

Global addresses are unique and are required by each host that communicates on the Internet. Every host has one link-local address on every network interface, and may have one (or more) global addresses on every network interface. The host part (interface id) is the same, regardless of the subnet. A host has the same host-part IPv6 address in link-local scope as well as in global scope. The first 64 bits, the network part of the address, must also be set. IPv6 is designed to have built-in support for hosts to figure out the network part of the address as well. This is called Stateless Address Auto-Configuration (SLAAC).

Note: *With IPv6, there are normally multiple addresses on each interface using different prefixes. This is a main difference to IPv4, where each interface normally has only a single IPv4 address and belongs to a single subnet.*

SLAAC

SLAAC is part of IPv6 protocol operation, every host and router must support it. When a router is configured with an IPv6 address on an interface, it then sends out the network part of the configured address in a Router Advertisement (RA) message so that any host connected on that interface receives information about which IPv6 subnets that are present. In addition the RA is used by the host to create a list of default routers for any communication to the Internet. The “default route” in IPv6 points at the link-local IPv6 address of one or more routers.

With prefix information from the RA, the host can proceed to create one address from each announced IPv6 prefix using the same interface id as used with the link-local address. In this way, the host receives a valid and working IPv6 address from every subnet configured on the interface of the router.

Stateful Address Assignment

A router can indicate in its RA message that configuration of addresses for a particular IPv6 prefix is stateful rather than stateless. This means that the host must use DHCPv6 to obtain its address.

For stateful addresses, the host will perform a DHCP SOLICIT, and a DHCPv6 server will then respond with a suggested address that the host will use.

IPv6 Addressing

IPv6 addresses are 128-bits long (compared to 32 bits for IPv4). But IPv6 is not simply an extended address size compared to IPv4. There are significant changes to the address architecture. For more details on the IPv6 addressing architecture, see RFC4291.

There are three types of IPv6 addresses.

- *Unicast:* identifies a single interface (multiple physical interfaces can share a single unicast address as long as the multiple physical interfaces are presented as a single interface to the IP layer - this can be useful for load-sharing).
- *Anycast:* identifies a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to the closest (according to routing protocols) interface.
- *Multicast:* identifies a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.

Note: *IPv6 has no broadcast address, nor does it make any use of broadcast. This is an important difference to IPv4.*

All interfaces are required to have at least one link-local (FE80::) unicast address. The interface can have multiple IPv6 addresses of any type or scope.

One or more prefixes is associated with a link.

IPv6 addressing is further described in:

- "Representation of IPv6 Addresses" on page 184
- "Address Type Identification" on page 184
- "Unicast Addresses" on page 185
- "Anycast Addresses" on page 186
- "Multicast Addresses" on page 187
- "Required Addresses" on page 189
- "DHCPv6" on page 190
- "Neighbour Discovery" on page 197
- "Stateless Address Auto Configuration" on page 199

Representation of IPv6 Addresses

IPv6 addresses are represented as x:x:x:x:x:x:x:x where x consists of 1-4 hexadecimal digits. For example;

- ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
- 2001:DB8:0:0:8:800:200C:417A

Leading zeros can be omitted in any individual field, but there must be at least one numeral in every field except when multiple consecutive fields contains the value zero. In that case, in order to make writing easier, such fields can be compressed and expressed by "::" which means one or more groups of 16 bits of zeros.

Note: *"::" can only be used once in an address.*

The following table shows different representations of IPv6 addresses:

Table 20 Representation of IPv6 Addresses

Address	Compressed Version	Description
2001:DB8:0:0:8:800:200C:417A	2001:DB8::8:800:200C:417A	unicast address
FF01:0:0:0:0:0:0:101	FF01::101	multicast address
0:0:0:0:0:0:0:1	::1	loopback address
0:0:0:0:0:0:0:0	::	unspecified address

A node with address 2001:0DB8:0:CD30:123:4567:89AB:CDEF that is connected to a link with the prefix 2001:0DB8:0:CD30::/60 it is possible to express both the prefix and node address with the combination 2001:0DB8:0:CD30:123:4567:89AB:CDEF/60.

Address Type Identification

The type of IPv6 address (unicast, anycast, multicast) can be identified by the high order bits as follows.

Table 21 Types of IPv6 Addresses

Address Type	Binary Prefix	IPv6 Notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Global unicast	(everything else)	

Anycast addresses are taken from the global unicast type. An anycast address can not be distinguished from a normal unicast address by syntax or form.

Unicast Addresses

There are several types of unicast addresses in IPv6.

- Global unicast
- IPv6 with embedded IPv4 addresses
- Site-local unicast (now deprecated)
- Link-local unicast

A node may have little knowledge of the internal structure of the IPv6 address. At a minimum a node may consider that the unicast address has no internal structure.

Example 74 Unicast Address

```
|                           128 bits                            |
+---------------------------------------------------------------+
|                          node address                         |
+---------------------------------------------------------------+
```

More sophisticated nodes (such as routers) may have better understanding of prefixes and boundaries.

Interface identifiers in IPv6 are used to identify interfaces on a link. For all unicast addresses, except those that start with binary value 000, interface identifiers are required to be 64 bits long and to be constructed in modified EUI-64 format. This effectively requires all prefixes on links that connect nodes to be /64 size.

The Unspecified Address

The address 0:0:0:0:0:0:0:0 (::) is called the unspecified address. It must never be assigned to any node and indicates absence of an address. It must never be used as destination address and a router must never forward a packet with :: as source address

The Loopback Address

The address 0:0:0:0:0:0:0:1 (::1) is called the loopback address. It must not be assigned to a physical interface nor used as source address in any packet that are send outside of a single node. If used as destination address, it must also not ever be sent outside of a single node.

IPv4-Compatible IPv6 Address

An IPv6 address with embedded IPv4 address has a starting binary value 000 (exception to the 64-bit prefix rule). This is the IPv4-Compatible IPv6 address that was defined to assist in the IPv6 transition.

Example 75 IPv4-Compatible IPv6 Address

```
|                80 bits                | 16 |      32 bits        |
+---------------------------------------+----+---------------------+
|0000..............................0000|0000|    IPv4 address     |
+---------------------------------------+----+---------------------+
```

These are now deprecated (RFC4291) because of the current transition mechanisms no longer use these addresses. New or updated implementations are not required to support this address type.

IPv4-Mapped IPv6 Address

An IPv6 address with embedded IPv4 address is another type of IPv6 address that holds an embedded IPv4 address is defined.This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses.

Example 76 IPv4-Mapped IPv6 Address

```
|               80 bits               | 16 |      32 bits        |
+-------------------------------------+----+---------------------+
|0000.............................0000|FFFF|    IPv4 address     |
+-------------------------------------+----+---------------------+
```

More information on these can be found in RFC4038.

Link-local Addresses

Link-Local addresses are used on a single link.

Example 77 Link-local Address

```
|   10     |
|  bits    |         54 bits         |          64 bits           |
+----------+-------------------------+----------------------------+
|1111111010|           0             |       interface ID         |
+----------+-------------------------+----------------------------
```

Link-local addresses begin with FE80, and are never forwarded by routers outside of their link.

Site-local IPv6 Unicast Addresses (Deprecated)

Site local addresses (FEC0::/10) were originally intended to be used for addressing within a “site” without the need for global prefix. However, this use was deprecated in RFC3879 due to multiple issues with defining “site”, complexity in router implementations etc.

Example 78 Site-local IPv6 Unicast Address

```
|   10     |
|  bits    |         54 bits         |         64 bits            |
+----------+-------------------------+----------------------------+
|1111111011|        subnet ID        |       interface ID         |
+----------+-------------------------+----------------------------+
```

Anycast Addresses

Anycast addresses are taken from unicast addresses. An anycast address is an address used by multiple interfaces in the network. Traffic from a node to an anycast address is taking the shortest path to the closest interface with the anycast address configured.

Thus, anycast addresses are syntactically indistinguishable from unicast addresses.

One expected use of anycast addresses is to identify the set of routers belonging to an organization providing Internet service. Such addresses could be used as intermediate

addresses in an IPv6 Routing header, to cause a packet to be delivered via a particular service provider or sequence of service providers.

There is one predefined anycast address, the Subnet-Router

Example 79 Anycast Address

```
|                         n bits                  |   128-n bits   |
+-------------------------------------------------+----------------+
|                    subnet prefix                | 00000000000000 |
+-------------------------------------------------+----------------+
```

Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet. All routers are required to support the Subnet-Router anycast addresses for the subnets to which they have interfaces.

The Subnet-Router anycast address is intended to be used for applications where a node needs to communicate with any one of the set of routers.

Multicast Addresses

An IPv6 multicast address is an identifier for a group of interfaces. An interface may belong to any number of multicast addresses.

Example 80 Site-Local IPv6 Unicast Address

```
|   8    |  4 |  4 |                   112 bits                    |
+------ -+----+----+-----------------------------------------------+
|11111111|0RPT|scop|                   group ID                    |
+--------+----+----+-----------------------------------------------+
```

Multicast addresses always begin with FF followed by 4 bits defining flags, followed by four bits defining scope.

The high-order flag bit is reserved for not and must always be 0.

T-flag is used to indicate if the address is permanently assigned (“well-known”) when set to 0 or non-permanently-assigned (“transient” or “dynamic”) when set to 1. A permanently assigned address is independent of the scope.

P-flag is defined in RFC3306.

R-flag is defined in RFC3956.

GroupID identifies the multicast group within the given scope. See also RFC3306.

The scope field defines the multicast scope. See table below

Table 22 Multicast Scopes

Scope	Definition
0	reserved
1	Interface-Local scope (spans only a single interface on a node, only useful for loopback transmission of multicast)
2	Link-Local scope (spans same topology as link-local addresses)
3	reserved

Table 22 Multicast Scopes

Scope	Definition
4	Admin-Local scope (must be administratively configured, it cannot be automatically derived from physical connectivity or other non-multicast-related configuration)
5	Site-Local scope (intended to span a single site)
6	(unassigned)
7	(unassigned)
8	Organization-Local scope (intended to span multiple sites belonging to the same organisation)
9	(unassigned)
A	(unassigned)
B	(unassigned)
C	(unassigned)
D	(unassigned)
E	Global scope
F	reserved

Permanently-assigned Multicast Address

A permanently assigned multicast address (T-flag) is an address that serves the same purpose regardless of scope.

- FF01:0:0:0:0:0:0:101 means all NTP servers on the same interface (i.e., the same node) as the sender.
- FF02:0:0:0:0:0:0:101 means all NTP servers on the same link as the sender.
- FF05:0:0:0:0:0:0:101 means all NTP servers in the same site as the sender.
- FF0E:0:0:0:0:0:0:101 means all NTP servers in the Internet.

Non-permanently-assigned multicast addresses are meaningful only within a given scope. FF15:0:0:0:0:0:0:101 is a site-local multicast group, but its purpose may differ between different sites.

Pre-defined Multicast Addresses

The following table lists well-known multicast addresses used for a particular purpose.

Table 23 Pre-defined Multicast Addresses

Description	Multicast address
Reserved Multicast Addresses	FF00:0:0:0:0:0:0:0 FF01:0:0:0:0:0:0:0 FF02:0:0:0:0:0:0:0 FF03:0:0:0:0:0:0:0 FF04:0:0:0:0:0:0:0 FF05:0:0:0:0:0:0:0 FF06:0:0:0:0:0:0:0 FF07:0:0:0:0:0:0:0 FF08:0:0:0:0:0:0:0 FF09:0:0:0:0:0:0:0 FF0A:0:0:0:0:0:0:0 FF0B:0:0:0:0:0:0:0 FF0C:0:0:0:0:0:0:0 FF0D:0:0:0:0:0:0:0 FF0E:0:0:0:0:0:0:0 FF0F:0:0:0:0:0:0:0 These addresses are reserved and shall never be assigned to any multicast group.
All nodes addresses	FF01:0:0:0:0:0:0:1 FF02:0:0:0:0:0:0:1 These addresses identify the group of all nodes within interface-local and link-local scope.
All Routers Addresses	FF01:0:0:0:0:0:0:2 FF02:0:0:0:0:0:0:2 FF05:0:0:0:0:0:0:2 These addresses identify the group of all routers within interface-local, link-local and site-local scope.
Solicited-Node Address	FF02:0:0:0:0:1:FFXX:XXXX Solicited-node address is computed from a nodes unicast and anycast address by taking the low-order 24 bits of that unicast/anycast address and appending those bits to the FF02:0:0:0:0:1:FF00::/104 prefix. This gives a multicast address in the range FF02:0:0:0:0:1:FF00:0000-FF02:0:0:0:0:1:FFFF:FFFF For example, the Solicited-Node multicast address corresponding to the IPv6 address 4037::01:800:200E:8C6C is FF02::1:FF0E:8C6C

Required Addresses

A node must recognize a number of IPv6 addresses as identifying itself;

- The link-local address for each Interface (FE80::...)
- Any configured unicast or anycast address
- The loopback address (::1)
- The All-nodes multicast address (FF02::1)
- The Solicited-node multicast address for each of its unicast and anycat addresses (FF02:0:0:0:0:1:FF....)
- Multicast addresses of all other groups to which the node belongs.

A router must, in addition to the above, also recognize the following addresses as identifying itself;

- The Subnet-Router anycast address for all interfaces for which it is configured to act as a router.
- All other anycast addresses with which the router has been configured
- The All-Routers multicast address defined (FF02::1 and FF05::1)

DHCPv6

Dynamic Host Configuration Protocol version 6 (DHCPv6) is the stateful address assignment protocol used in IPv6.

DHCPv6 is defined in RFC3315.

The information on this page is derived from this and other standards, and combined with analysis and conclusions. Always refer to the standard document for accurate protocol description.

DHCPv6 Overview

DHCPv6 is a client/server protocol for stateful (address) auto-configuration of IPv6. DHCPv6 is used only for IPv6 address assignment. A client that also requires IPv4 addresses may use DHCPv4 to obtain IPv4 addresses. There is currently no standard to allow DHCPv6 to provide both IPv6 and IPv4 addresses and configuration.

DHCPv6 is used when the M-bit or O-bit is set in the Router Advertisement message issued by a router on a link (see “Neighbour Discovery” on page 197). If the M-bit is set, DHCPv6 will be used to obtain all address information. If only the O-bit is set, DHCPv6 is used for additional parameters (such as DNS server, NTP server) but not for actual IPv6 address assignment of clients.

Clients and servers exchange DHCPv6 messages using UDP. The client uses a link-local address or addresses determined through other mechanisms for transmitting and receiving DHCPv6 messages.

DHCPv6 servers receive messages from clients using a reserved, link-scoped multicast address, (multicast address for all DHCP Relay Agents and Servers is FF02::1:2). A DHCPv6 client transmits all DHCPv6 messages to this reserved multicast address unless explicitly permitted by the server to send unicast messages. Even if communicating with a particular DHCPv6 server, the multicast address can be used as the server is identified by its unique DUID (see below).

A DHCPv6 relay agent on the client's link can relay messages between the client and a server that is not directly connected. The operation of the relay agent is transparent to the client.

The server provides IPv6 configuration parameters and addresses depending on the clients request. It is possible for a DHCPv6 server to assign Temporary addresses (RFC4941) to improve privacy.

Two Message Exchange

If a client does not need IPv6 address assignment by DHCPv6 (O-bit set, M-bit is not set), or if the client is willing to accept address suggestions immediately, the packet exchange with the DHCPv6 server can consist of only two messages.

In the first case, the client issues an Information-Request message to obtain additional configuration information from the server, and the server replies with a Reply message containing the configuration information for the client.

In the second case, if the client also requires an address, it sends a Solicit message that also indicates that the client is willing to accept an immediate Reply message. The server will respond if it is willing to immediately commit address assignment to the client. Each address assigned to the client has associated preferred and valid lifetimes specified by the server. To request an extension of the lifetimes assigned to an address, the client sends a Renew message to the server. The server sends a Reply message to the client with the new lifetimes, allowing the client to continue to use the address without interruption.

Four Message Exchange

The four-message exchange is used by clients to obtain IPv6 addresses and other configuration information (M-bit set). The client sends a Solicit message to find available DHCPv6 servers. A server that can meet the clients requirements responds with an Advertise message. The client may receive multiple responses and therefore choses one of the servers and sends a Request message for which the server responds with a Reply message that contains the confirmed address and configuration.

Renewals are handled as described in the previous section.

DHCPv6 Operation

DHCPv6 Messages

The following table describes the messages used by DHCPv6.

Table 24 DHCPv6 Messages

Client message	Server message	Description
SOLICIT		Sent to locate servers
	ADVERTISE	Response to Solicit, server available for DHCP service.
REQUEST		Request configuration parameters from specific server
CONFIRM		Sent by clients to any available server to determine if addresses it was assigned are still appropriate to the link to which the client is connected.
RENEW		Sent by client to original server to extend lease of IPv6 address.
REBIND		Sent by client to any server (if no response to RENEW) to extend lease of IPv6 address

Table 24 DHCPv6 Messages

Client message	Server message	Description
	REPLY	Server response (to Solicit, Request, Renew or Rebind) with assigned addresses and configuration. Server response (to Information-request) with configuration parameters. Server reponse (to Confirm) to permit or deny address vs. link. Server response (to Release or Decline) to acknowledge recepit.
RELEASE		Sent by client to discontinue use of assigned address(es).
DECLINE		Sent by client if it determines that server's suggested address is already in use
	RECONFIGURE	Sent to trigger Renew/Information-request message from client (sent to UDP port 546 on the client).
INFORMATION-REQUEST		Sent by client that want parameters but not address from DHCP server.

In addition, there is a RELAY-FORW message sent by a DHCP relay-agent to relay messages to the server, and a RELAY-REPL message that is the server response to a relay agent.

All DHCP messages share an identical fixed format header and variable format area for options.

DHCP Unique Identifier (DUID)

Each DHCP client and server has a DUID. DUID's are used to identify clients for the selection of configuration parameters and association of Identity Association (IA) with clients. The DUID is used by clients to identify a server in messages, where such identification is required.

The DUID is an opaque value, up to 128 bytes long. DUID is carried as an option and is unique across all DHCP clients and servers, and stable for any specific client or server (it does not change) over time even as a result of hardware changes to a device.

The RFC proposes three different types of DUID's.

- Link-layer address + time (DUID-LLT)
- Vendor-assigned unique ID based on enterprise number (DUID-EN)
- Link-layer address (DUID-LL)

DUID-LLT

The DUID-LLT is recommended use for all general purpose computing devices such as desktop computers and laptops, and also printers, routers or other devices with a writable non-volatile storage.

The DUID-LLT is generated using a hardware type as assigned by IANA according to RFC826. In addition a link-layer address of an interface is used. This address is selected only once and augmented with a timer value (seconds from 1/1 2000 modulo 2^32) to create a unique value. This value can not be changed; it is persistent even if the source link-layer interface is replaced, and it does not change when the system restarts or even if the timer is adjusted. Once selected, the value is used forever for the particular host when creating the DUID-LLT. An interface can be moved to a new devices, and

therefore the timer value serves as a randomizer to prevent to different hosts that use the same link-layer address from obtaining the same DUID-LLT.

DUID-EN

The DUID-EN is constructed from the vendor enterprise id (as registered by IANA, for Waystream 9303) and a unique identifier assigned by the vendor (for example a serial number). The DUID-EN must be assigned at manufacturing and stored in non-eraseable storage.

DUID-LL

The DUID-LL is constructed from a link-layer address of an interface that is permanently connected to the device. A hardware type, as defined in RFC826 is also used with the DUID-LL. The same DUID-LL should be used to configure all interfaces on the device, and it must be persistent through restarts.

Identity Association (IA)

The IA serves as a device unique identifier for a particular interface on a device. IPv6 configuration information for that interface is associated with the IA including the IPv6 address and T1 and T2 timers. The IA must be consistent across restarts of the device.

The DHCP server determines what IPv6 addresses to assign to an IA according to the server address assignment policies in addition to information about the link to which the client is attached, the DUID supplied by the client, other information from options provided by the client, other information provided as options by the relay agent (if present).

DHCP Client-Server Operation

The client first creates an IA and assigns it an IAID. The client then transmits a Solicit message containing an IA option describing the IA. Servers that can assign addresses to the IA respond to the client with an Advertise message.

The Solicit Message

The client sets the “msg-type” field to SOLICIT. The client generates a transaction ID and inserts this value in the “transaction-id” field.

The client MUST include a Client Identifier option to identify itself to the server. The client includes IA options for any IAs to which it wants the server to assign addresses. The client MAY include addresses in the IAs as a hint to the server about addresses for which the client has a preference. The client MUST NOT include any other options in the Solicit message, except as specifically allowed in the definition of individual options.

The client uses IA_NA options to request the assignment of non-temporary addresses and uses IA_TA options to request the assignment of temporary addresses. Either IA_NA or IA_TA options, or a combination of both, can be included in DHCP messages.

The client SHOULD include an Option Request option to indicate the options the client is interested in receiving. The client MAY additionally include instances of those

options that are identified in the Option Request option, with data values as hints to the server about parameter values the client would like to have returned.

The client includes a Reconfigure Accept option if the client is willing to accept Reconfigure messages from the server.

The Advertise Message

The server first determines if it will respond to the client, based on its administrative policy. For example, if the server policy state that it may only respond to a client that is willing to accept a Reconfigure message and the client indicates that it will not accept such messages, the server will discard the Solicit message.

The server sets the "msg-type" field to ADVERTISE and copies the contents of the transaction-id field from the Solicit message received from the client to the Advertise message. The server includes its server identifier in a Server Identifier option and copies the Client Identifier from the Solicit message into the Advertise message.

The server MAY add a Preference option to carry the preference value for the Advertise message. The server implementation SHOULD allow the setting of a server preference value by the administrator. The server preference value MUST default to zero unless otherwise configured by the server administrator.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server includes options the server will return to the client in a subsequent Reply message. The information in these options may be used by the client in the selection of a server if the client receives more than one Advertise message. If the client has included an Option Request option in the Solicit message, the server includes options in the Advertise message containing configuration parameters for all of the options identified in the Option Request option that the server has been configured to return to the client. The server MAY return additional options to the client if it has been configured to do so.

If the Solicit message from the client included one or more IA options, the server MUST include IA options in the Advertise message containing any addresses that would be assigned to IAs contained in the Solicit message from the client.

The Request Message

A valid Request message causes the server to create the bindings for the client, and record the IAs and other information requested by the client.

The Reply Message

If the client includes a Rapid Commit option in the Solicit message, it will expect a Reply message that includes a Rapid Commit option in response. The client discards any Reply messages it receives that do not include a Rapid Commit option. If the client receives a valid Reply message that includes a Rapid Commit option, it processes the message.

If it does not receive such a Reply message and does receive a valid Advertise message, the client processes the Advertise message.

This means that the client may request rapid commit, but will follow the normal four-message process if no server responds to the rapid commit.

Rapid-commit should only be permitted where there is a single DHCP server responding to client Solicit messages. Otherwise, each server will allocate addresses for the client, but the client will only chose one of the addresses and the servers will not know that some addresses remain unused.

The client should perform Duplicate Address Detection on each address received by the server. A client must send a Decline message if it detects that the addresses assigned to it is already in use by another node. This will make the server mark the address declined and not use it for other clients.

The client extracts the following information from the Reply message;

- T1 and T2 times
- New addresses in the IA option to the IA
- Update lifetimes for any addresses in the IA option that is already recorded in the IA
- Discard any addresses from the IA that has a valid lifetime of 0
- Leave unchanged any information about addresses that are recorded in the IA but that were not included

The Confirm Message

The Confirm message can be used by clients to verify that its previously received configuration is still valid and usable. A client may issue a Confirm message whenever it has (or suspect it has) moved to another link. For example;

- After reboot
- When physically connected to a wired connection
- When returning from sleep mode
- When changing access point (using wireless technology)

If the server responds that the client is not on link, it has to perform a normal DHCP Solicit process.

Relay Agent Behaviour

If a server determines that the Solicit message was received through a relay agent, the server unicasts the Relay-reply message directly back to the relay agent.

A DHCP server should not permit unicast communication with clients if relay agents are used. This ensures that relay agent options are added to all DHCP messages from the client.

A relay agent can use a unicast address or multicast address of the DHCP server. There is a site-local multicast address reserved (FF05::1:3) for communication with DHCP servers. More commonly, a unicast address is likely to be used.

A relay agent relays both messages from a client and messages (Relay-forward) messages from other relay agents.The source address of the DHCP packet is used as peer-address for Relay-forward messages. The client DHCP message (excluding IP and UDP headers) is copied into a Relay Message option of the new message.

If the message is received from a client, the relay agent puts a prefix of the link on which the client should be assigned an address in the link-address field. This address is used by the server to determine the link from which the client should be assigned an address.

If the relay agent cannot use the address in the link-address field to identify the interface of the client, the relay agent must include an Interface-id option in the Relay-forward message. This option will then be included by the server in its Relay-reply message. The link-address field must be filled in regardless of use of Interface-id option.

It is possible to have multiple relay agents, in which case the relay reply will contain nested information for each relay agent.

The relay-agent can add additional options to the packet that provide identification information that leads the DHCP server to the correct decision. In DHCPv4 the commonly used “option-82” is one example.

In DHCPv6, the following relay-agent options provide equivalent information:

- Subscriber-ID option (38); Customer administrative information, for example a customer-ID or other information that is not network dependent.
- Remote-ID option (37): Identify remote host end of a circuit, typically used with the relay-agents DUID to identify which switch/router that relays the message

Note: *Interface-id option(18) also may be added by the agent*

The Address Lease

At time T1, the client will issue a Renew message to extend the lifetime of its IA. Recommended to be .5 of the lifetime.

At time T2, the client will issue a Rebind message to extend the lifetime of its IA. This will only happen if there was no Reply received at time T1. Recommended to be 80% of the lifetime.

DHCP Options

The following DHCP options are of particular interest.

Table 25 DHCPv6 Options

Option Value	Option Name	Description	Source
1	Client Identifier	The DUID of a client	RFC3315
2	Server Identifier	The DUID of a server	RFC3315
3	Non-temporary address	IA_NA, A non-temporary IPv6 address option	RFC3315
4	Temporary address	IA_TA, A temporary IPv6 address option	RFC3315
5	IA Address option	The IPv6 address	RFC3315
6	Option Request	Option Identifies a list of options for which parameters are requested	RFC3315
9	Relay message option	Carries a DHCP message in a Relay-forward or relay-reply message	RFC3315
12	Server Unicast Option	Indicates if the server accept unicast communication	RFC3315
14	Rapid Commit Option	Use two message exchange for address assignment	RFC3315

Table 25 DHCPv6 Options

Option Value	Option Name	Description	Source
15	User Class Option	Identify the type or category of user or application the client represents	RFC3315
16	Vendor Class Option	Identify the hardware vendor (by enterprise id) of the client	RFC3315
17	Vendor Specific Information Option	Any vendor (by enterprise id) specific information	RFC3315
18	Interface Id option	Identifier for interface on which client message was received (for relay agents)	RFC3315
38	Subscriber-ID option	Relay agent option; Customer administrative information	RFC4580
37	Remote-ID Option	Relay agent option; Identify the remote host end of the circuit	RFC4649

Neighbour Discovery

Neighbour Discovery replaces ARP. It is the protocol by which IPv6 hosts determine the link-layer address (Ethernet MAC address) of its neighbours.

Notable difference between Neighbour Discovery and relevant IPv4 protocols (ARP, ICMP redirect etc.) are:

- Discovery and link-layer address of routers are automatically provided through Router Advertisements.
- Prefixes are automatically provide and SLAAC operation simplifies address assignment.
- Coordination of MTU values.
- Address resolution uses multicast instead of broadcast, reducing address-resolution-related interrupts on nodes other than the target node. Non-IPv6 nodes are not affected at all.
- Redirects provide link-layer address of next-hop directly.
- Redirects are assumed to be “on-link”. In IPv4 the redirect has to go to a destination within the same prefix (netmask) as the redirected host.
- Unreachability detection is part of the base functionality, which improves robustness of packet delivery.
- Routers are identified by their IPv6 link-local addresses. This allows hosts to be renumbered without affecting the router association. The IPv6 link-local address of a router is used in a hosts default route. The host will still use a global unicast IPv6 address for Internet communication, but to look-up the link-layer (MAC address) of the router, the routers IPv6 link-local address is used in the hosts destination cache/default router table.
- The Hop limit is always set to 255. Any Neighbour Discovery packet with any other hop limit value is ignored by hosts. This prevents Neighbour Discovery packets from outside of the local network from being processed by hosts on the network.

- Address resolution using ICMPv6 is more media-independent than ARP is (Ethernet centric).

IPv6 neighbour discovery also performs a number of other tasks not described here. For full details of the protocol, see RFC 4861.

Purpose of Neighbour Discovery

Neighbour Discovery performs or provides the data for a number of different operations on the local network, for example within a VLAN or Ethernet LAN segment.

- *Router Discovery:* Allowing hosts to detect routers and build a list of default routers.
- *Prefix Discovery:* Routers announce the IPv6 prefixes used on the network (LAN).
- *Parameter Discovery:* Parameters such as the MTU of the network can also be provided.
- *Address Auto configuration:* Providing the necessary information for stateless address auto configuration (SLAAC).
- *Address resolution:* Determining the link-layer address of neighbours on the network.
- *Next-hop determination:* Determining the appropriate next-hop to use for a destination.
- *Neighbour unreachability detection:* Determining if a neighbour is still reachable on the network.
- *Redirect:* Communicating a better next-hop to use for a certain destination.

Neighbour Discovery Packet Types

Neighbour Discovery operates using ICMPv6. Unlike ARP, neighbour discovery packets are not sent to the broadcast address. Instead, multicast addresses are used. Some addresses are defined in the standard, like the all-nodes and all-routers addresses. In addition, IPv6 offers a mechanism by which a unicast IPv6 address also has a corresponding multicast IPv6 address on each network. This address can be calculated, and is therefore used to reach a particular host in some circumstances.

The following five packet types are used by Neighbour Discovery:

- *Router Solicitation:* When an interface becomes enabled, hosts may used this message to request routers to generate a Router Advertisement message immediately rather than at the next scheduled advertisement.
- *Router Advertisement:* Send by routers to inform hosts about its presence (so hosts can add the router to its the default router table) and to provide information about the network (configured prefixes, use of stateful or stateless address assignment, MTU etc.). Router Advertisements are sent regularly and also as response to a Router Solicitation message.
- *Neighbour Solicitation:* Used to determine the link-layer address of a neighbour or to verify that a neighbour is still reachable. Also used for Duplicate Address Detection to determine if a selected IPv6 address is unique on the network.
- *Neighbour Advertisement:* Response to a Neighbour Solicitation message or sent to inform about a link-layer address change.

- *Redirect:* Sent by routers to instruct hosts to use a different next-hop for a particular destination.

Router Advertisements

Router Advertisements (RA) are sent by routers on multicat capable networks (such as Ethernet).

A host that receives an RA will automatically add the router to its list of Default Routers.

Since RAs are regularly generated, a host will soon learn about all routers on its connected interfaces.

The RA message contains several vital pieces of information.

- The M and O bits that indicate

—M: Hosts should use DHCPv6 to obtain addresses (stateful or Managed address assignment)

—O: Hosts should use DHCPv6 to obtain additional information.

- A list of prefixes for the network (excluding the link local IPv6 prefix) to allow the host to determine which prefixes that are directly reachable and which that has to go through a router.

—Flags for each prefix to indicate if automatic address assignment should take place.

- MTU of the network

—Hop limit to use in outgoing packets

Neighbour Solicitation

Neighbour Solicitation is used by nodes to ask another node for its link-layer address (MAC address). Neighbour Solicitation messages are multicast to the “solicited-node multicast address” which is the IPv6 multicast address version of the target nodes IPv6 unicast address (for which the link-layer address is sought).

This message is also used for duplicate address detection as described by RFC 4862 on Stateless Address Auto configuration.

Stateless Address Auto Configuration

Stateless Address Auto configuration (SLAAC) is a mechanism within IPv6 that allows hosts to select and configure their IP address without manual or DHCP server intervention. SLAAC relies on information obtained by the Neighbour Discovery Router Advertisement messages.

SLAAC is defined in RFC4862.

The information in this section is derived from RFC4862 and other standards, and combined with analysis and conclusions. Always refer to the standard document for accurate protocol description.

The specification also contains the mechanism for Duplicate Address Detection (DAD).

SLAAC Automatic Configuration

The automatic configuration process includes:

- Generating a link-local (FE80::0) address
- Generating a global address using SLAAC and DAD

The stateless auto configuration mechanism does not require manual configuration of hosts, and only minimal configuration of routers. SLAAC only needs information from Router Advertisement messages (prefix information) in combination with information that is already available to the host. By combining the announced prefixes with a self-generated interface ID, a complete IPv6 address is formed.

Stateless auto configuration can be used when there is no concern with the exact addresses hosts use as long as they are routeable and unique. If stateful address assignment with better control is desired, DHCPv6 must be used instead. It is possible to combine the two so that they are used simultaneously.

IPv6 addresses are leased to an interface. All addresses assigned has a lifetime (which may be infinite). When the lifetime expires, it can be reassigned to another host and must not be used. Therefore IPv6 addresses have two stages; preferred (unrestricted use) and deprecated (binding will become invalid). In deprecated state the address should not be used - no new communication should use it. This two stage approach allows smoother renumbering as hosts can gradually switch over from deprecated to preferred addresses.

Through Duplicate Address Detection (DAD), a host can determine if its intended address is unique on the link. DAD is always used, regardless of the way the address has been obtained - through SLAAC or DHCPv6.

Routers do not use SLAAC. Since they announce Router Advertisement messages they have to be manually configured. Only hosts may use SLAAC. Routers will, however, generate their link-local (FE80::0) address using the mechanisms of RFC4862 and perform DAD on all addresses prior to assigning them to an interface.

SLAAC Operation

Auto configuration is performed only on multicast capable links. When the interface is enabled a link-local address if formed by appending an identifier of the interface to the well-known link-local prefix FF80::/10.

Before the tentative link-local address can be assigned, it has to pass DAD to ensure that it is unique on the link. DAD is performed by sending a Neighbour Solicitation message with this tentative address. If another node is already using the address, it will respond a Neighbour Advertisement message that prevents use of the address.

If a node determines that its tentative link-local address is already in use, the node must stop and wait for manual configuration of the link-local address interface identifier.

Once the link-local address has been obtained the next step is to get information from a Router Advertisement (RA) message about what sort of auto configuration a host may perform or to determine that no routers are present. A host may therefore issue a Router Solicitation (RS) message to trigger an RA response.

The RA response contains zero or more prefixes that are used on the link. Each prefix also has a flag to indicate if SLAAC is allowed for that prefix. The M and O bits of the RA message also indicate if the host should use DHCPv6 to obtain addresses or other

configuration from DHCP. The contents of the RA message therefore may suggest combined use of SLAAC and DHCPv6 for host configuration.

Creating Link-Local Addresses

The link-local address is formed by combining the well-known link-local prefix FE80::0 with an interface identifier.

1. The left-most (prefix length) bits of the address are those of the link-local prefix.
2. The bits in the address to the right of the link-local prefix are set to all zeroes.
3. If the length of the interface identifier is N bits, the right-most N bits of the link-local address are replaced by the interface identifier.
4. If the sum of the link-local prefix length and N is larger than 128, auto configuration fails and manual configuration is required.
5. The length of the interface identifier is link dependent and given in separate standards documents per link type.

A link-local address never times out, it has infinite lifetime.

Using the MAC Address

On Ethernet networks, the MAC address of an interface can serve as the interface identifier. RFC4291 defines rules for creating modified EUI-64 format identifiers from MAC addresses.

The 48-bit MAC address contains two main sections, the assigned company id (0008.ae for Waystream) which is the “c”-bits in the example below and the manufacturer selected identifier in the “m”-bits (the unique part of the MAC).

Example 81 MAC Address “c” and “m” bits

```
|0              1|1              3|3              4|
|0              5|6              1|2              7|
+----------------+----------------+----------------+
|cccccc0gcccccccc|cccccccmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+
```

There is also one bit within the company id which represents global/local scope of the address (marked in bold text). This is generally always set to 0 in MAC address to create global uniqueness of the address. Next to it is the “g”-bit, which defines individual/group bit.

To convert this MAC address into an interface identifier for IPv6, the MAC address is split in the middle, between the company id (“c”) and manufacturer id (“m”). Between them is the hexadecimal values 0xFF and 0xFE inserted (marked in bold text). The global scope bit is also changed to local scope (marked in italic text).

Example 82 MAC Address converted to IPv6 Interface id

```
|0             1|1             3|3             4|4             6|
|0             5|6             1|2             7|8             3|
+----------------+----------------+----------------+----------------
--+
|cccccc1gcccccccc|cccccccc11111111|11111110mmmmmmmm|mmmmmmmmmmmmmmmm
mm|
+----------------+----------------+----------------+----------------
--+
```

This is then appended to the link-local prefix of FE80:: to form the unique link-local address of a node.

In the table below are examples of MAC address and corresponding Link-local address. Note that leading 0 often are omitted when link-local addresses are printed, and also the change in the first part of the MAC address as the global/local scope bit is changed.

Table 26 MAC Address and Link-local Address

MAC-address	Link-local address
0008.ae50.5ac0	fe80::208:aeff:fe50:5ac0/64

Duplicate Address Detection

Duplicate Address Detection (DAD) is performed on all unicast addresses prior to assigning them on an interface regardless of DHCPv6, SLAAC or manual configuration with the following exceptions:

- If DAD has been administratively turned off on the interface if the address in question is an anycast address (there are then multiple hosts that has the same address DAD uses Neighbour Solicitation (NS) and Neighbour Advertisement (NA) messages.
- If DAD detects a duplicate, the node must obtain a new identifier through manual configuration.

An address is regarded as tentative until DAD has been performed and no objections found. Packet processing of packets addressed to the tentative address must not occur.

Before sending the DAD NS, the node must join the all-nodes (FF02:1) multicast address and the solicited-node multicast address of the tentative address, using MLD. This ensures that the node will receive any NA from other nodes already using the tentative address and that two nodes attempting to use the same address simultaneously should detect each other's presence.

To check for DAD, the node sends the DAD NS message every RetransTimer milliseconds. If a NA message for the tentative address is received in response, the tentative address is not unique and the DAD has failed - the address cannot be used. This should generate an error message.

If the DAD failure is for a link-local (FE80::) address, IPv6 operation on the interface should be disabled (not send any IP packets, drop any received packets and not forward any packets to the interface). In this case, the duplicate probably means a duplicate link-layer (MAC) address on the link which requires administrator attention, provided that the interface identifier is derived from the hardware address. If it is not, then other rules may apply.

Creating Global Addresses

A global address is formed by appending an interface identifier to a prefix of appropriate length. Prefixes are obtained from the prefix information option in the Router Advertisement message.

If there is no router on the link, there might still be a DHCPv6 server to provide addresses to hosts, and hosts may want to attempt use this service in absence of RA messages.

RA messages with prefix information options are processed as follows:

- If the autonomous flag is not set, the prefix is ignored.
- Link-local prefixes are also ignored.
- If the preferred lifetime is greater than the valid lifetime, ignore the prefix.
- If the prefix has not already been used to create an address on the interface, and if the valid lifetime is not 0, form an address by combining the prefix with the interface identifier. The sum of the prefix length and interface identifier must equal 128 bits or the prefix must be ignored. If an address is successfully formed, it is added to the list of addresses assigned to the interface.
- If the formed address already exists (and was created by SLAAC), the preferred lifetime of the address is reset to the value from the recent RA message. The action to perform for the valid lifetime then depends on the Valid Lifetime value in the received advertisement and the time to the valid lifetime expiration of the previously SLAAC configured address. The protocol defines rules to prevent denial of service attacks where bogus advertisements attempt to cause prefix to expire.

IPv6 Addressing Commands

To configure IPv6 addressing, the following commands are implemented:

- **ipv6 address**
- **ipv6 nd**

For more details, refer *iBOS Command Reference*.

IPv6 Routing

Support for IPv6 routing is described in the following:

- "IPv6 Static Routing" on page 204
- "OSPFv3" on page 204
- "IPv6 GRE Tunnels" on page 206

IPv6 Static Routing

For IPv6 static addresses, the following commands are implemented:

- **ipv6 route**
- **show ipv6 route**

The following example configures two static routes:

```
ibos(config)# ipv6 route 3ffe:506::/64 2001:db8:1:1::24
ibos(config)# ipv6 route 3ffe:507::/64 2001:db8:1:1::24 32
```

For more details, refer *iBOS Command Reference*.

OSPFv3

OSPF version 3 is introduced for IPv6 and has the same fundamental mechanisms as version 2. There are, however, some significant differences:

- Protocol processing per-link, not per-subnet
- Removal of addressing semantics
- Addition of Flooding scope
- Explicit support for multiple instances per link
- Use of link-local addresses
- Authentication changes
- Packet format changes
- LSA format changes
- Handling unknown LSA types
- Stub area support

The OSPFv3 protocol is fully described in RFC 2740.

OSPFv3 Commands

To configure OSPFv3 for iBOS, the following commands are used:

- **ipv6 ospf**
- **ipv6 ospf cost**
- **ipv6 ospf dead-interval**
- **ipv6 ospf hello-interval**

- **ipv6 ospf priority**
- **ipv6 ospf retransmit-interval**
- **ipv6 ospf transmit-delay**
- **ipv6 router ospf area**
- **router ospfv3**
- **show ipv6 ospf**

OSPFv3 Configuration Example

The following example configures basic OSPF routing between two servers, ASR1 and ASR2.

Uplink for ASR1 is gigabitethernet2 and interface connected to ASR2 is gigabitethernet1. The OSPFv3 configuration for ASR1 is as follows:

```
router ospfv3
   router-id 10.0.0.10
    interface gigabitethernet1
      ipv6 address 2001::101/110
      ipv6 router ospf area 1
      no shutdown
    interface gigabitethernet2
      ipv6 address 2002::201/64
      ipv6 router ospf area 1
      no shutdown
```

ASR2 is connected to ASR1 using gigabitethernet1, OSPF is configured downstream on gigabitethernet2. The OSPFv3 configuration for ASR2 is:

```
router ospfv3
   router-id 20.0.0.20
    interface gigabitethernet1
      ipv6 address 2001::102/110
      ipv6 router ospf area 1
      no shutdown
    interface gigabitethernet2
      ipv6 address 2003::301/64
      ipv6 router ospf area 1
      no shutdown
```

IPv6 Forwarding

Support for IPv6 forwarding is described in the following:

- “IPv6 Unicast” on page 206
- “IPv6 GRE Tunnels” on page 206
- “IPv6 DHCP Relay” on page 206

IPv6 Unicast

To forward unicast traffic on an interface configured for IPv6, iBOS uses the system IPv6 routing table to calculate the best path to the destination configured by **ipv6 route** command. The system routing table consist of routes known by different routing protocols, static routes and connected subnets. The best path is stored into the system forwarding table that is then used for forwarding of IPv6 packets. To display data from the forwarding table, the following **show ipv6 forwarding** command is used, for example:

Example 83 IPv6 Forwarding Table

```
ibos# show ipv6 forwarding
Destination              Gateway              Interface       Proto
-------------------------------------------------------------------
--------------
2002:e51d:2:1::/64         connected             gigabitethernet1
local
2001:db8:1:1::/64         2002:e51d:2:1::3      gigabitethernet1
static
```

IPv6 GRE Tunnels

A 4-IN-4 GRE tunnel can be created by configuring the IP address for the tunnel interface. See “GRE Tunnels” on page 176.

Additionally, a 6-IN-4 GRE tunnel also can be created by configuring the IPv6 address.

Example 84 6-IN-4 GRE Tunnel

```
ibos(config)# interface tunnel1
ibos(config-if-tun)# tunnel mode gre
ibos(config-if-tun)# gre 192.168.2.25 source-interface loopback0
ibos(config-if-tun)# ipv6 address 2001:db8::101/64
```

IPv6 DHCP Relay

Support for DHCPv6 relay to external server has been added.

The following IPv6 related commands have been added.

Table 27 New iBOS Commands for IPv6

Command	Description
ipv6 dhcp relay script	Enables the IPv6 DHCPv6 relay script on an interface

For more information, refer *iBOS Command Reference.*

Configuring MPLS

Multiprotocol Label Switching (MPLS) is a high performance packet forwarding technology that integrates the performance and traffic capabilities of Layer 2 switching with the scalability, flexibility and performance of Layer 3 network routing.

This chapter is for network administrators and application developers intending to configure the MPLS protocols on Waystream hardware. Using topology illustrations and configuration examples, basic and advanced configuration is described.

MPLS configuration is described in the following sections:

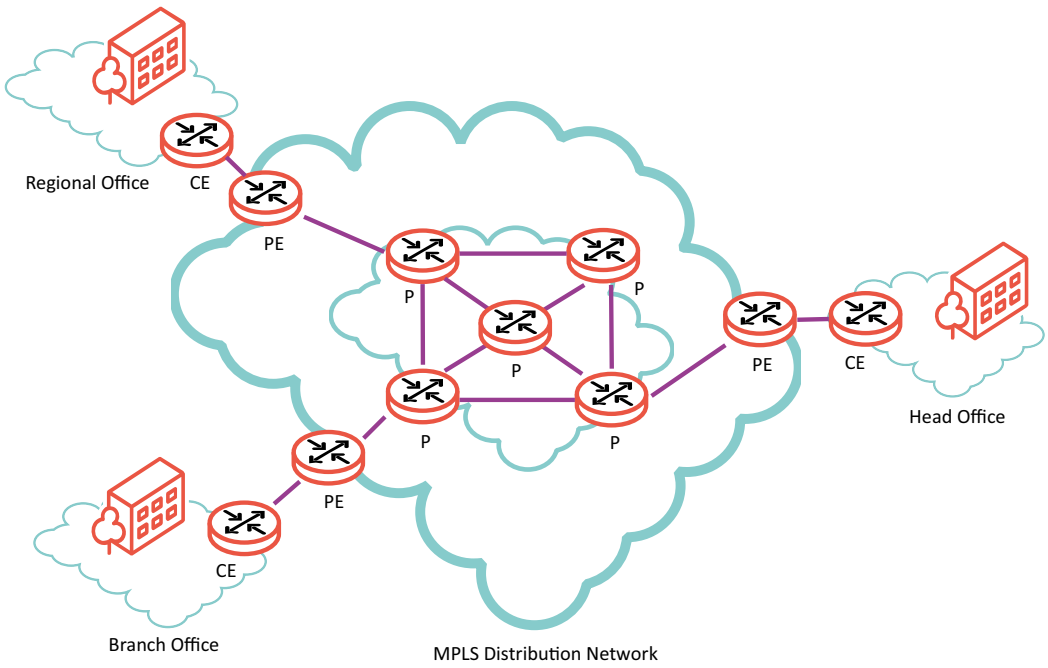
- "What is MPLS" on page 208
- "MPLS Forwarding" on page 209
- "MPLS Standards" on page 211
- "MPLS LDP" on page 211
- "LDP Configuration Examples" on page 213
- "MPLS Layer-3 VPNs" on page 218
- "MPLS Layer 2 VPN" on page 227
- "MPLS Traffic Class" on page 233

What is MPLS

Multiprotocol Label Switching (MPLS) is a standards-approved technology for speeding up network traffic flow and making it easier to manage. MPLS involves setting up a specific path for a given sequence of packets, identified by a label put in each packet, thus saving the time needed for a router to look up the address to the next node to forward the packet to. MPLS is called multi-protocol because it works with the Internet Protocol (IP), Asynchronous Transport Mode (ATM), and frame relay network protocols. With reference to the standard model for a network (the Open Systems Interconnection, or OSI model), MPLS allows most packets to be forwarded at the Layer 2 (switching) level rather than at the Layer 3 (routing) level. In addition to moving traffic faster overall, MPLS makes it easy to manage a network for quality of service (QoS). For these reasons, MPLS is adopted as networks begin to carry more and different mixtures of traffic.

MPLS is best summarized as a "Layer 2.5 networking protocol" as it sits between the traditional Layer 2 and Layer 3 parts of the OSI data model. MPLS provides additional features that allow transport of data across the network. The intra-AS routing is usually handled by the IGP (for example, OSPF, RIP). BGP requires the routes to be reachable before it can advertise it to another AS. The routes are advertised to the other ASs through network layer reachability information. "MPLS in the Network" on page 208 shows an service provider distribution network with core routers (P), provider edge routers (PE) and customer edge routers (CE) that uses MPLS in the core network.

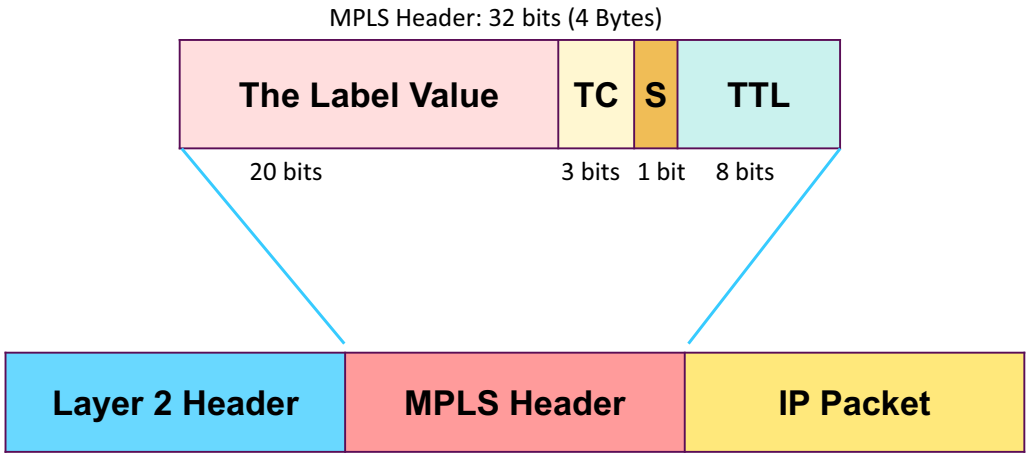
Figure 32 MPLS in the Network



MPLS Forwarding

The heart of the MPLS Forwarding Plane is the MPLS Header that is inserted between the Layer 2 header and the IP header of the data packet. The following figure shows this important component.

Figure 33 MPLS Header in Data Packet



Note: *The label value is just one component of the MPLS header.*

The MPLS Header includes the following components:

- Label value.

The 20-bit value serves as the basis for packet forwarding in the MPLS cloud. MPLS will use this value for a quick lookup in the MPLS forwarding table

- Traffic Class (TC) field.

These 3-bits are the Traffic Class Bits and typically carry the IP Precedence value from the IP Packet.

- Bottom of Stack Bit (S-bit)

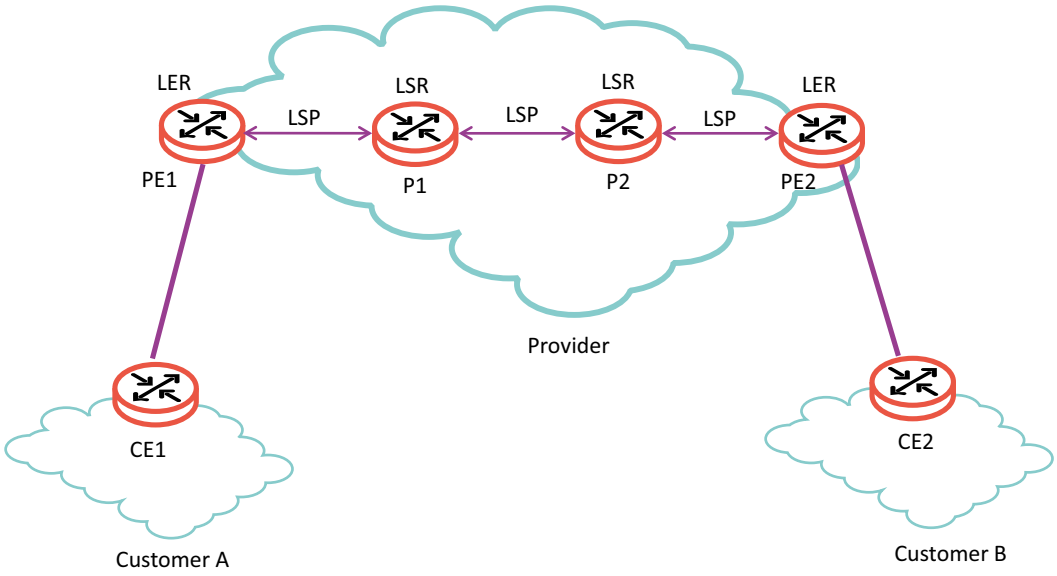
There are many instances when MPLS headers are stacked within a packet. This bit is set on the bottom header to indicate the bottom of the stack has been reached.

- Time-To-Live field

This field is used for loop prevention and possibly path-tracing in the MPLS cloud. This value decrements with each hop and packet discards occur when the value is zero.

To describe the basic MPLS Forwarding process, the following typical MPLS topology is used:

Figure 34 MPLS Basic Forwarding



IP packets arriving at the PE1 router (Label Edge Router) will have one, or more, MPLS headers applied. To determine the label or labels to be imposed, the incoming packet is first associated with a Forwarding Equivalence Class (FEC) and then mapped to a Next Hop Label Forwarding Entry (NHLFE), which contain the label and nexthop information required to forward the packet. This procedure is known as FEC to NHLFE (FTN) and in this example, it is simply the operation of performing a lookup in the IP routing table and associated FTN table.

The label identifies the egress Label Edge Router (LER) and the Label Switching Path (LSP) that the packet will follow. When the packet arrives at the Label Switching Router (LSR) P1, the LSR looks up the label in the Incoming Label Map (ILM) table and swaps it with the appropriate label for the outgoing interface.

At router P2, which is the penultimate LSR, the router performs a lookup in the ILM table and removes the MPLS header. It then forwards the packet as an IP packet to PE2. This functionality is known a Penultimate Hop Popping (PHP) and is the default behavior. PHP saves workload for the PE2 LER which simply performs the standard IP forwarding.

Note: *The PHP functionality may be disabled by configuring* **explicit null** *on PE2 which will forward the packet to the PE2 router as an MPLS packet with the topmost label set to Explicit NULL, and the traffic class (TC) setting will be intact.*

MPLS Standards

MPLS support in iBOS conforms to the following MPLS standards:

Table 28 MPLS Standards Supported by iBOS

Standard	Description
RFC 3031	Multiprotocol Label Switching Architecture
RFC 3032	MPLS Stack Encoding
RFC 3270	MPLS Support of Differentiated Services
RFC 3443	Time To Live (TTL) Processing in MPLS Networks
RFC 3469	Framework for Multi-Protocol Label Switching (MPLS)-based Recovery
RFC 4364	BGP/MPLS IP Virtual Private Networks (VPNs)
RFC 4379	Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures
RFC 4950	ICMP Extensions for Multiprotocol Label Switching
RFC 5462	"EXP field" renamed to "Traffic Class field"
RFC 6291	Guidelines for the Use of the "OAM" Acronym in the IETF

MPLS LDP

The Label Distribution Protocol (LDP) is a routing protocol often used in MPLS solutions. The iBOS LDP daemon (ldpd) runs on top of the iBOS layer and uses iBOS services to obtain routing information. Routers send Hello packets to establish Hello Adjacencies with other nearby routers. This opens the way for sessions between routers to be established during which routers exchange labels in preparation for forwarding packets.

LDP generates labels for and exchanges labels between peer routers. It works in with other routing protocols (OSPF and BGP) to create label-switched paths (LSP) used when forwarding packets. A label-switched path is the path taken by all packets that belong to the Forwarding Equivalence Class (FEC) corresponding to that LSP. This is analogous to establishing a virtual circuit in ATM (Asynchronous Transfer Mechanism). In this way, LDP assigns labels to every destination address and destination prefix provided by iBOS. The LDP interface to the MPLS forwarder adds labels to, and deletes labels from, the forwarding tables.

LDP Adjacencies

LDP defines a mechanism for discovering adjacent, LDP capable Label Switching Routers (LSR) that participate in label switching (adjacencies). Whenever a new router comes up it sends out a hello packet to a specific multicast address in order to announce itself to the network. Every router directly connected to the network receives the packet. Receipt of a hello packet from another LSR creates a Hello Adjacency with that LSR. To create a Hello Adjacency with an LSR that cannot send/receive multicast packets, LDP allows a router to be manually configured to send unicast Hello packets to non-multicast LSRs. This non-multicast LSR is a targeted peer. Adjacencies are

maintained by sending out periodic Hello packets to the multicast group and to all targeted peers. Hello packets are sent using UDP.

LDP Session

LDP capable LSRs establish a session before exchanging label information. All the session messages are sent using TCP to ensure reliable delivery. After the LSRs establish a session and negotiate options, a given pair of routers may exchange label information. The labels exchanged over a session are valid only during the lifetime of the session and routers release them when session is closed.

Forwarding Equivalence Class

A Forwarding Equivalence Class (FEC) section defines a set of packets that are forwarded on the same path by the MPLS network. Two common methods to define FEC are by advertising the IPv4 routes using:

- *Host Address:* The LSR uses the address of the destination host to create this FEC. This means that all the packets going to this destination will take the same LSP.
- *Prefix:* The LSR uses destination prefix to create this FEC. This means that all the packets take the LSP corresponding to the longest matching prefix.

Label Generation

An LDP Label is a 20-bit number the LSR uses to forward a packet to its destination. When an LSR creates a new FEC, the router generates new labels and distributes them to its peers. A router keeps both incoming and outgoing labels in its database.

Label Distribution Modes

iBOS LDP implementation supports two label distribution modes:

- *Downstream Unsolicited:* In this mode, next hop LSRs distribute labels to peers without waiting for a label request.
- *Downstream on Demand:* In this mode, a LSR distributes a label to a peer only if there is a pending label request from the peer.

Label Retention Mode

iBOS LDP implementation supports the following label retention modes:

- *Liberal Retention Mode:* In this mode, the LSR retains all labels received from all sources. This mode helps in fast LSP setup in case of a change in next hop.

LSP Control

LSPs can be set up in the following two ways:

- *Ordered Control:* In this mode, an LSR distributes a label for a FEC to its peer only if it has a corresponding label from its next hop or it is the egress node.
- *Independent Control:* In this mode, an LSR may distribute a label to its peers without waiting for a corresponding label from its next hop.

LDP Configuration Examples

To see details about the LDP commands used in these examples, refer *iBOS Command Reference*.

Protocol names in the samples indicate the protocol daemons on which the configuration is performed. iBOS uses a common CLI for all protocol daemons and users can perform the configuration in command modes without having to switch protocols. For commands used to switch command modes, refer "Contexts" on page 20.

The **enable-ldp ipv4** command is used to enable LDP for IPv4 on a specified interface.

Enable Label Switching

To enable label switching on iBOS, the following tasks need to be performed:

1. Enable label-switching on the interface.
2. Enable LDP on an interface in the LDP daemon.
3. Run an IGP (Internal Gateway Protocol), for example, OSPF, to distribute reachability information within the MPLS cloud.
4. Configure the transport address.

Figure 35 Enable LDP Configuration Topology

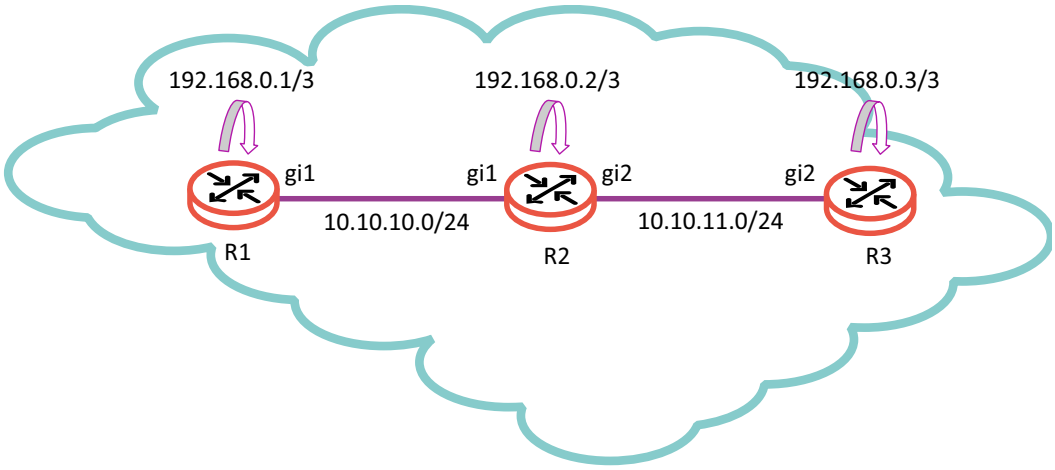


Table 29 Configure label switching on Device R1

Command	Description
`iBOS#configure terminal`	Enter Configuration mode
`iBOS(config)#interface gi1`	Specify the interface (gigabitethernet1) to be configured.
`iBOS(config-if)#label-switching`	Enable label switching on interface gi1.
`iBOS(config-if)#exit`	Return to the Configure mode.
`iBOS(config)#interface lo`	Specify the loopback (lo) interface to be configured.
`iBOS(config-if)#ip address 192.168.0.1/32`	Set the IP address of the loopback interface to 192.168.0.1/32.

Table 30 Configure LDP on Device R1

Command	Description
`iBOS(config)#router ldp`	Enter Router mode for LDP.
`iBOS(config-router)#router-id 192.168.0.1`	Set the router ID to IP address 192.168.0.1.
`iBOS(config-router)#transport-address ipv4 192.168.0.1`	Configure the transport address to be used for a TCP session over which LDP will run on an IPv4 interface. It is recommended to use the loopback address as the transport address.
`iBOS(config-router)#exit`	Return to the Configure mode.
`iBOS(config)#interface gigabitethernet1`	Enable LDP on gigabitethernet1.
`iBOS(config-if)#enable-ldp ipv4 iBOS(config-if)#exit`	Return to Configure mode.

Table 31 Configure OSPF on device R1

Command	Description
`iBOS(config)#router ospf 100`	Configure the routing process and specify the Process ID (100). The Process ID should be a unique positive integer identifying the routing process.
`iBOS(config-router)#network 10.10.10.0/24 area 0` `iBOS(config-router)#network 192.168.0.1/32area 0`	Define the interface on which OSPF runs and associate the area ID (0) with the interface.

Table 32 Configure label switching on device R2

Command	Description
`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface (lo) to be configured.
`iBOS(config-if)#ip address 192.168.0.2/32`	Set the IP address of the loopback interface to 192.168.0.2/32.
`iBOS(config-if)#exit`	Return to Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gi1) to be configured.
`iBOS(config-if)#label-switching`	Enable label switching on interface gi1.
`iBOS(config-if)#exit`	Return to Configure mode
`iBOS(config)#interface gigabitethernet2`	Specify the interface (gi2) to be configured.
`iBOS(config-if)#label-switching`	Enable label switching on interface gi2.
`iBOS(config-if)#exit`	Return to Configure mode.

Table 33 Configure LDP on device R2

Command	Description
`iBOS(config)#router ldp`	Enter Router mode for LDP.
`iBOS(config-router)#router-id 192.168.0.2`	Set the router ID to IP address 192.168.0.2.
`iBOS(config-router)#transport-address ipv4 192.168.0.2`	Configure the transport address to be used for a TCP session over which LDP will run on an IPv4 interface. It is recommended to use the loopback address as the transport address.
`iBOS(config-router)#exit`	Return to the Configure mode.
`iBOS(config)#interface gigabitethernet1` `iBOS(config-if)#enable-ldp ipv4`	Enable LDP on gigabitethernet1.
`iBOS(config-if)#exit`	Return to Configure mode.

Command	Description
`iBOS(config)#interface gigabitethernet2` `iBOS(config-if)#enable-ldp ipv4`	Enable LDP on gigabitethernet2.
`iBOS(config-if)#exit`	Return to Configure mode.

Table 34 Configure OSPF on device R2

Command	Description
`iBOS(config)#router ospf 100`	Configure the routing process and specify the Process ID (100). The Process ID should be a unique positive integer identifying the routing process.

`iBOS(config-router)#network 10.10.10.0/24 area 0`
`iBOS(config-router)#network 10.10.11.0/24 area 0`
`iBOS(config-router)#network 192.168.0.2/32 area 0`
	Define the interfaces on which OSPF runs and associate the area ID (0)with them.

Table 35 Configure label switching on device R3

Command	Description
`iBOS#configure terminal`	Enter configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface (lo) to be configured.
`iBOS(config-if)#ip address 192.168.0.3/32`	Set the IP address of the loopback interface to 192.168.0.3/32.
`iBOS(config-if)#exit`	Exit the Interface mode and return to the Configure mode.
`iBOS(config)#interface gigabitethernet2`	Specify the interface (gi2) to be configured.
`iBOS(config-if)#label-switching`	Enable label switching on interface gi2.
`iBOS(config-if)#exit`	Exit the Interface mode and return to the Configure mode

Table 36 Configure LDP on device R3

Command	Description
`iBOS(config)#router ldp`	Enter Router mode for LDP.
`iBOS(config-router)#router-id 192.168.0.23`	Set the router ID to IP address 192.168.0.3.
`iBOS(config-router)#transport-address ipv4 192.168.0.3`	Configure the transport address to be used for a TCP session over which LDP will run on an IPv4 interface. Note: It is preferable to use the loopback address as transport address. In addition, use the parameter “ipv6” if you are configuring an IPv6 interface.
`iBOS(config-router)#exit`	Return to the Configure mode.
`iBOS(config)#interface gigabitethernet2`	Enter the Interface mode for gi2.
`iBOS(config-if)#enable-ldp ipv4`	Enable LDP.
`iBOS(config-if)#exit`	Return to Configure mode.

Table 37 Configure OSPF on device R3

Command	Description
`iBOS(config)#router ospf 100`	Configure the routing process and specify the Process ID (100). The Process ID should be a unique positive integer identifying the routing process.
`iBOS(config-router)#network 10.10.11.0/24 area 0` `iBOS(config-router)#network 192.168.0.3/32 area 0`	Define the interfaces on which OSPF runs and associate the area ID (0)with them.

MPLS Layer-3 VPNs

This section contains configurations for MPLS Layer-3 Virtual Private Networks (VPN). It also contains a brief overview to the concepts of MPLS Layer-3 VPN and simple topology diagrams.

The MPLS Layer-3 VPN solution provides address space and routing separation via the use of per-VPN Routing and Forwarding tables (VRFs), and MPLS switching in the core and at the edge of the network. VPN customer routing data is imported into the VRFs utilizing the Route Target BGP extended community. This routing data is identified by a Route Distinguisher (RD) and is distributed among Provider Edge (PE) routers using Multiprotocol BGP extensions.

Requirements for MPLS L3 VPN

The iBOS MPLS Layer-3 VPN solution requires the following iBOS features:

- BGP
- LDP
- MPLS
- OSPF

To check that your iBOS version includes these modules, run the **show features** command:

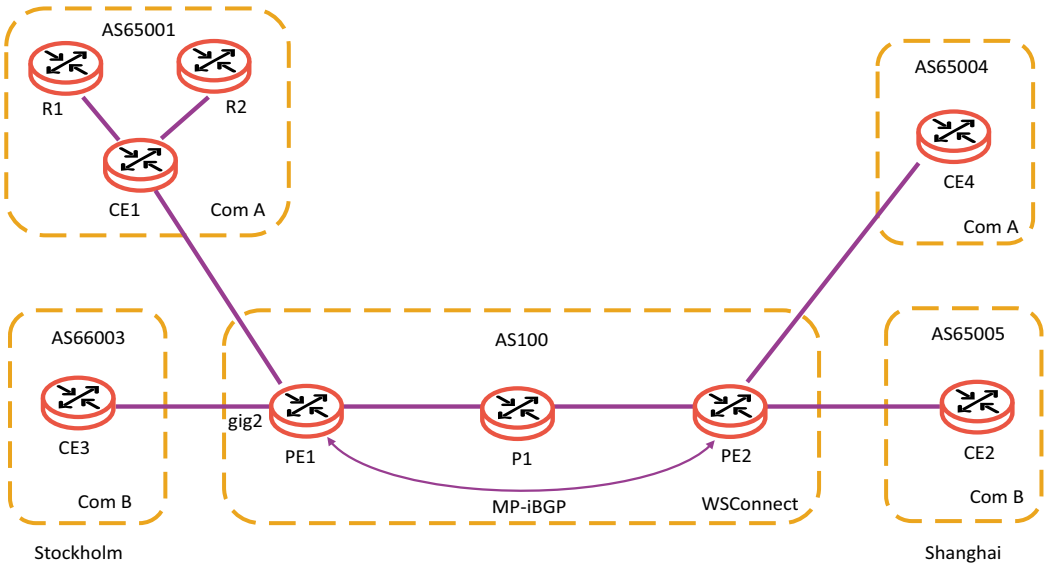
```
mpc> show features
Feature                    Description
-------------------------------------------------------------
BGP                        Border Gateway Protocol
LDP                        Label Distribution Protocol
MPLS                       Multiprotocol Label Switching Protocol
OSPF                       Open Shortest Path First Protocol
mpc>
```

If you are missing some of the above features, then you must upgrade the iBOS software.

MPLS VPN Topology

The following figure illustrates a Virtual Private Network in a WSConnect Service Provider Network. The figure includes the MPLS VPN terms defined in this section.

Figure 36 Provider WSConnect with Customers ComA and ComB



Terminology	Description
Service Provider.	The organization that owns the infrastructure that provides leased lines to customers, offering them a Virtual Private Network Service. In the above illustration, WSConnect is the service provider providing services to clients ComA and ComB.
Customer Edge (CE) Router.	A router at a customer's site that connects to the Service Provider via one or more Provider Edge routers. In the above illustration, CE1, CE2, CE3 and CE4 are all CE routers connected directly to the CConnect network.
Provider Edge (PE) Router.	A provider's router connected to a CE router through a leased line or dial-up connection. In the above illustration, PE1 and PE2 are the PE routers, because they link the CConnect service provider to its clients.
Provider Core Router (P).	The devices in the core of the service provider network, which are generally not Provider Edge routers. In the above illustration, the P router is the provider device, not connected to any customer, and is the core of the WSConnect network.
Site.	A contiguous part of the customer network. A site connects to the provider network through transmission lines, using a CE and PE router. In the above illustration, R1, R2 and CE3 comprise a Customer network, and are seen as a single site by the WSConnect network.
Customer Router.	In the illustration above, R1 and R2 are the Customer routers, and are not directly connected to the WSConnect network.

MPLS-VPN Routing Process

The iBOS MPLS-VPN routing process involves the following steps:

1. Service Providers provide VPN services from PE routers that communicate directly with CE routers via an Ethernet Link.
2. Each PE router maintains a Routing and Forwarding table (VRF) for each customer. This guarantees isolation, and allows the usage of uncoordinated private addresses. When a packet is received from the CE, the VRF that is mapped to that site is used to determine the routing for the data. If a PE has multiple connections to the same site, a single VRF is mapped to all of those connections.
3. After the PE router learns of the IP prefix, it converts it into a VPN-IPv4 prefix by prepending it with an 8-byte Route Distinguisher (RD). The RD ensures that even if two customers have the same address, two separate routes to that address can be maintained. These VPN-IPv4 addresses are exchanged between the PE routers through MP-BGP.
4. A unique Router ID (usually the loopback address) is used to allocate an LDP-specific tunnel label, and to enable VPN packet forwarding across the backbone.
5. Based on routing information stored in the VRF table, packets are forwarded to their destination using MPLS. Each PE router allocates a unique BGP-specific VPN label to every route in each VRF (even if they have the same next hop), and propagates these labels, together with 12-byte VPN-IPv4 addresses, through MP-BGP.
6. Ingress PE routers prepend a two-level label stack to the VPN packet, which is forwarded across the Provider network. This label stack contains a BGP-specific label from the VRF table (the VRF is associated with the incoming interface) that specifies the BGP next hop and an LDP-specific label from the global FTN table, specifying the IP next hop.
7. The Provider router in the network switches the VPN packet, based on the top label or the LDP-specific label in the stack. This top label is used as the key to lookup in the incoming interface's Incoming Labels Mapping table (ILM). If there is an outbound label, the label is swapped, and the packet is forwarded to the next hop. If no outbound label exists, the router is the penultimate router so it pops the LDP-specific label and forwards the packet with only the BGP-specific label to the egress PE router. The last operation is also known as Penultimate Hop Popping (PHP).
8. The egress PE router pops the BGP-specific label, performs a single label lookup in the associated VRF table, and sends the packet to the appropriate CE router.

Configure MPLS Layer 3 VPN

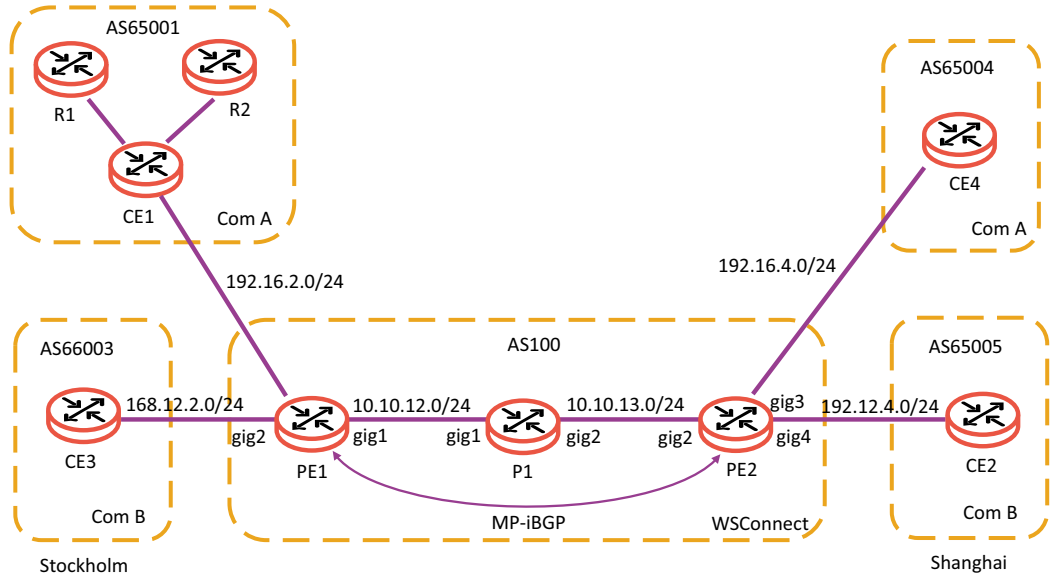
To configure MPLS Layer 3 VPN in iBOS, the following tasks are performed:

1. Establish a connection between the Provider Edge (PE) routers.
2. Configure PE routers as iBGP neighbours.
3. Optionally, configure the MPLS Traffic Class, see "MPLS Traffic Class" on page 233.
4. Create a VRF for each VPN.

5. Associate interfaces to VRFs.
6. Configure VRF Route Distinguisher and Route Targets.
7. Configure CE neighbour for the VPN.
8. Verify MPLS to VPN configuration.

"WSConnect Example Topology" on page 221 illustrates the topology used for the configuration examples.

Figure 37 WSConnect Example Topology



In this example, the WSConnect MPLS-VPN backbone has two customers — ComA and ComB. Both customers have sites in Stockholm and Shanghai. The following topology shows BGP4 address assignment between PE and CE routers. The configuration example provisions a customer VPN service across the MPLS-VPN backbone.

Establish Connections between PE Routers

To establish this connection, the following steps are involved:

1. Enable label switching for the Labeled Switched Path (LSP) between PE1 and PE2.

```
PE1(config)#interface eth1
PE1(config-if)#label-switching
!
P
P(config)#interface eth1
P(config-if)#label-switching
!
P(config)#interface eth2
P(config-if)#label-switching
!
PE2
PE2(config)#interface eth2
PE2(config-if)#label-switching
!
```

2. Enable IGP to establish connections between the two Provider Edge routers PE1 and PE2

```
PE1(config)# router ospf 100
PE1(config-router)# network 10.10.12.0/24 area 0
!
P(config)#router ospf 100
P(config-router)# network 10.10.12.0/24 area 0
P(config-router)# network 10.10.13.0/24 area 0
!
# configure terminal
PE2(config)# router ospf 100
PE2(config-router)# network 10.10.13.0/24 area 0
```

3. Set up a Label-Switched Path (LSP) by configuring LDP between Provider Edge routers PE1 and PE2.

The configuration for router PE1 is as follows:

```
PE1(config)#interface eth1
PE1(config-if)#enable-ldp
```

The configuration for router P is as follows:

```
P(config)#interface eth1
P(config-if)#enable-ldp
P(config)#interface eth2
P(config-if)#enable-ldp
```

The configuration for router PE2 is as follows:

```
PE2(config)#interface eth2
PE2(config-if)#enable-ldp
```

Configure PE Routers as BGP Neighbours

BGP is the preferred protocol to transport VPN routes because of its multi-protocol capability and its scalability. Its ability to exchange information between indirectly connected routers supports keeping VPN routing information out of the Provider (P) routers. The P routers carry information as an optional BGP attribute. Additional attributes are transparently forwarded by any P router. The MPLS-VPN forwarding model does not require the P routers to make routing decisions based on VPN addresses. They forward packets based on the label value attached to the packet. The P routers do not require VPN configuration in order to carry this information.

To configure PE1, do the following:

```
PE1(config)# interface lo
PE1(config-if)# ip address 2.2.2.2/32
!
PE1(config)# router bgp 100
PE1(config-router)# neighbour 3.3.3.3 remote-as 100
PE1(config-router)# neighbour 3.3.3.3 update-source 2.2.2.2
PE1(config-router)# address-family vpnv4 unicast
PE1(config-router-af)# neighbour 3.3.3.3 activate
```

To configure PE2, do the following:

```
PE2(config)# interface lo
PE2(config-if)# ip address 3.3.3.3/32
```

Create VRF

Each PE router in the MPLS-VPN backbone is attached to a site that receives routes from a specific VPN, so the PE router must have the relevant VRF configuration for that VPN. To create the VRF, use the **ip vrf** command.

This command creates a VRF RIB (Routing Information Base), assigns a VRF-ID, and switches the command mode to VRF mode on the iBOS daemon. The following example demonstrates the use of the command to create a VRF named ComB on router PE1:

```
PE1(config)# ip vrf ComB
(config-vrf)#
```

Associate Interfaces to VRFs

After the VRFs are defined on the PE router, the PE router needs to recognize which interfaces belong to which VRF. The VRF is populated with routes from connected sites. More than one interface can belong to the same VRF. To associate the interfaces (connected to the CE routers) to the VRFs, use the **ip vrf forwarding** command. In the following example, interface eth1 is associated with the VRF named ComB:

```
PE1(config)# interface eth1
PE1(config-if)# ip vrf forwarding ComB
```

Configure VRF—RD and Route Targets

After the VRF is created, configure Router Distinguishers and the Route Targets.

Configure Route Distinguishers

Route Distinguishers (RDs) make all customer routes unique. The routes must be unique, so that MP-BGP treats the same prefix from two different VPNs as non-comparable routes. To configure RDs, a sequence of 64 bits is prepended to the IPv4 address in the multiprotocol BGP update. BGP considers two IPv4 addresses with different RDs as non-comparable, even if they have the same address and mask.

To assign a particular value to the RD for each VRF on the PE router, use the **rd** command. The following example shows the addition of RD on VRF ComB:

```
PE1(config)# ip vrf ComB
PE1(config-vrf)# rd 1:1
```

Configure Route Targets

Any routes learned from customers are advertised across the network through multiprotocol BGP, and any routes learned through multiprotocol BGP are added into the appropriate VRFs. The route target helps PE routers identify which VRFs should receive the routes. Use the **route-target** command to configure the route targets.

The route-target command creates lists of import and export route-target extended communities for the VRF. It specifies a target VPN extended community. Execute the command once for each community. All routes with the specific route-target extended community are imported into all VRFs with the same extended community as an import route-target.

The following example demonstrates the route-target configuration for ComB.

```
PE1(config)#ip vrf ComB
PE1(config-vrf)#route-target both 100:1
```

Note: *Once the RD is configured, it is NOT possible to change the value of the RD. The only way to change an RD assigned to a VRF is to delete and then recreate the VRF before the new RD can be added.*

Configure CE Neighbour for the VPN using BGP and OSPF

To provide a VPN service, the PE-router must be configured so that any routing information learned from a VPN customer interface can be associated with a particular VRF. This is achieved using a standard routing protocol process (OSPF, BGP or static routes etc). Use any one of the following configurations (BGP and OSPF) to configure the CE neighbour.

Configure CE Neighbour using BGP

The BGP sessions between PE and CE routers can carry different types of routes (VPN-IPv4, IPv4 routes). Address families are used to control the type of BGP session. Configure a BGP address family for each VRF on the PE-router, and a separate address family to carry VPN-IPv4 routes between PE routers. All non-VPN BGP neighbours are defined using the IPv4 address mode. Each VPN BGP neighbour is defined under its associated Address Family mode. To enter the address-family mode, use the **address-family** command.

A separate address family entry is used for every VRF, and each address family entry can have multiple CE routers within the VRF. The PE and CE routers must be directly connected for BGP4 sessions; BGP multihop is not supported between PE and CE routers.

The following example places the router in address family mode, and specifies company names, ComA and ComB, as the names of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. This configuration is used when BGP is used for PE and CE.

```
PE1(config)# router bgp 100
PE1(config-router)# address-family ipv4 vrf ComA
PE1(config-router-af)# neighbour 192.16.3.3 remote-as 65001
!
PE1(config-router)# address-family ipv4 vrf ComB
PE1(config-router-af)# neighbour 168.12.0.2 remote-as 65003
```

Configure CE Neighbour using OSPF

Unlike BGP and RIP, OSPF does not run different routing contexts within one process. Thus, for running OSPF between the PE and CE routers, configure a separate OSPF process for each VRF that receives VPN routes through OSPF. The PE router distinguishes routers belonging to a specific VRF, by associating a particular customer interface to a specific VRF and to a particular OSPF process.

To redistribute VRF OSPF routes into BGP, redistribute OSPF under the BGP VRF address family submode.

The following example places the router in address family mode, and specifies company names, ComA and ComB, as the names of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. This configuration is used when BGP is used for PE and CE.

```
PE1(config)# router ospf 101 comA
PE1(config-router)# network 192.16.3.0/24 area 0
PE1(config-router)# redistribute bgp
PE1(config)# router ospf 102 comB
PE1(config-router)# network 192.12.0.0/24 area 0
PE1(config-router)# redistribute bgp
PE1 (config)# router bgp 100
PE1(config-router)# address-family ipv4 vrf ComA
PE1(config-router-af)# redistribute ospf
!
PE1(config-router)# address-family ipv4 vrf ComB
PE1(config-router-af)# redistribute ospf
```

Verify the MPLS-VPN Configuration

Use the **show ip bgp neighbour** command to validate the neighbour session between the CE and the PE routers. Use the **show ip bgp vpnv4 all** command to display all the VRFs and the routes associated with them.

The following is sample output for the **show running-config** command for the PE1, CE1 and P routers displaying the complete configuration (based on the topology in the diagram above).

Note: *In this example, BGP was used to configure the PE to CE link.*

```
!
hostname CE1
password zebra
log stdout
!
interface lo
!
interface eth0
!
interface eth1
!
interface sit0
!
interface ppp0
!
interface eth2
ip address 168.12.2.2/24
!
interface eth3
!
router bgp 65003
 router-id 10.10.10.10
redistribute connected
neighbour 168.12.2.1 remote-as 100
!
-------------------------------------------------------------------
!
hostname PE1
password zebra
log stdout
!
ip vrf ComB
rd 1:1
route-target both 100:1
```

```
!
ip vrf ComA
rd 1:2
route-target both 100:2
!
interface lo
ip address 2.2.2.2/32
!
interface eth0
ip address 192.16.3.2/24
ip vrf forwarding ComA
!
interface eth1
label-switching
ip address 10.10.12.10/24
enable-ldp
!
interface eth2
ip address 168.12.2.1/24
ip vrf forwarding ComB
!
interface sit0
!
interface ppp0
!
router bgp 100
neighbour 3.3.3.3 remote-as 100
neighbour 3.3.3.3 update-source 2.2.2.2
!
address-family vpnv4 unicast
neighbour 3.3.3.3 activate
exit-address-family
!
address-family ipv4 vrf ComB
neighbour 168.12.2.2 remote-as 65003
no neighbour 168.12.2.2 send-community extended
exit-address-family
!
address-family ipv4 vrf ComA
neighbour 192.16.3.3 remote-as 65001
no neighbour 192.16.3.3 send-community extended
exit-address-family
!
router ldp
advertisement-mode downstream-on-demand
loop-detection
!
router ospf 100
redistribute connected
network 10.10.12.0/24 area 0
!
-----------------------------------------------------------------
!
hostname P
password zebra
log stdout
!
interface lo
!
```

```
interface eth0
!
interface eth1
label-switching
ip address 10.10.12.50/24
enable-ldp
!
interface eth2
label-switching
ip address 10.10.13.50/24
enable-ldp
!
interface eth3
!
interface eth4
!
interface sit0
!
interface ppp0
!
router ldp
advertisement-mode downstream-on-demand
loop detection
!
router ospf 100
redistribute connected
network 10.10.12.0/24 area 0
network 10.1.13.50/24 area 0
```

MPLS Layer 2 VPN

MPLS Layer 2 VPNs provide an alternative to private networks that are provisioned using dedicated lines.

iBOS deploys Layer 2 services over MPLS to build a topology of point-to-point connection that connect end customer sites in a VPN. Services provisioned with these Layer 2 VPNs are known as a Virtual Private Wire Service (VPWS). A VPWS instance is configured on each associated edge device for each Layer 2 VPN.

In iBOS, the MPLS Layer 2 VPN solution is implemented using Ethernet over MPLS (EoMPLS) and is compliant with RFC 4448.

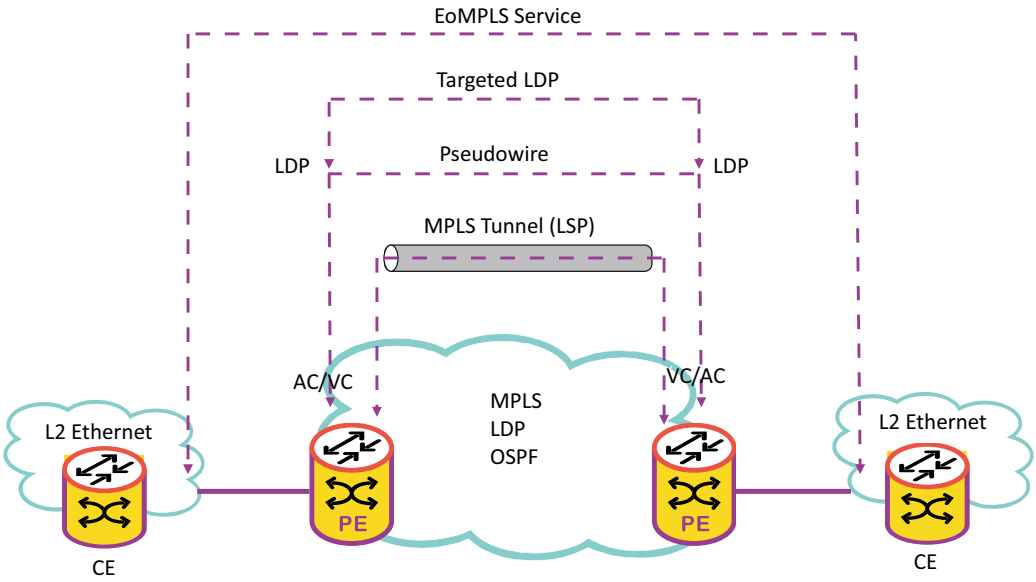
EoMPLS

EoMPLS provides an L2VPN solution that enables bidirectional point-to-point connections to emulate Ethernet connectivity over a MPLS core network. This solution includes the following components (listed in order of configuration):

1. A pseudowire to implement a point-to-point connection between two virtual circuits (VC) and a standard MPLS tunnel for the transport between the pseudowire endpoints (VCs).
2. A client network is bound to a pseudowire endpoint by associating a physical port or tagged VLAN interface with the pseudowire. This is known as binding an attachment circuit (AC) to a virtual circuit (VC).

3. Ethernet frames received from a client network are encapsulated with at least one pseudowire label and one MPLS Tunnel label, i.e. a MPLS stack of two labels is prepended to each Ethernet frame before it is sent over the MPLS core network.
4. The pseudowire is setup with targeted LDP (Label Distribution Protocol) between the two Provider Edge (PE) routers hosting the VCs.
5. The MPLS core network is setup by using LDP and an IGP (e.g. OSPF).

Figure 38 EoMPLS Solution



To use the EoMPLS solution, MPLS tunnels (LSPs) must be setup between the Provider Edge (PE) routers using, for example, LDP in conjunction with an Interior Gateway Protocol (IGP).

Pseudowires

A pseudowire emulates a bidirectional point-to-point connection for a specific service between a pair of Virtual Circuits (VC). The VCs are the endpoints of the connection and reside on two different PE routers. The PE routers are situated at the edge of the MPLS core network.

A pseudowire is bidirectional, i.e. traffic in both directions is allowed. The bidirectional property of the pseudowire is implemented by using two unidirectional MPLS LSPs, one in each direction. A VC bundles these two LSPs together in each endpoint.

Virtual Circuits

A virtual circuit (VC) implements the terminating endpoint of a pseudowire and provides the functionality to send and receive packet to/from the remote endpoint. Further, a VC is bound to an attachment circuit (AC) and provides the functionality to send and receive packets to/from the AC, i.e. to/from the client network.

To configure the VC on an interface, the **mpls-l2-circuit** command is used.

Attachment Circuits

An attachment circuit (AC) is used to connect a client network to a virtual circuit (VC) and a pseudowire. An AC is bound to one and only one VC, and vice versa, in case of point-to-point connection emulation. Further, all packets received on a AC are forwarded to the VC and sent over the associated pseudowire. In the opposite direction, all packets received on the pseudowire are forwarded to the VC and sent on the associated AC.

A pseudowire may use two different types of attachment circuits; a physical port or a tagged VLAN interface. These two types are referred to as; port mode and VLAN mode, and are configured with the iBOS commands **mpls-l2-circuit** and **mpls-l2-circuit vlan** respectively.

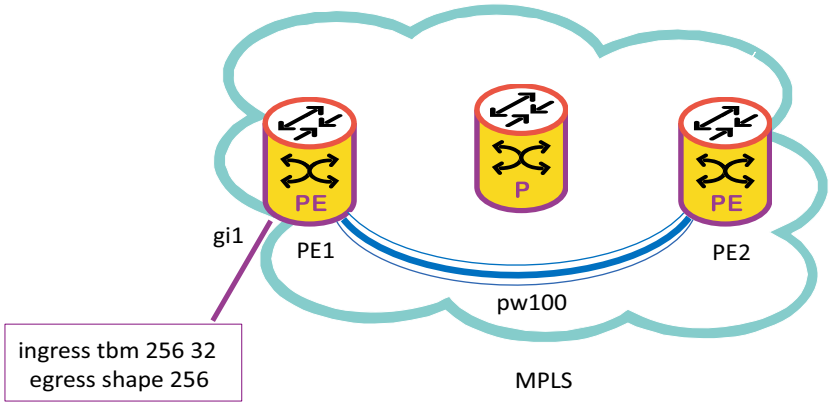
Traffic Control using a Policy Map on Pseudowires

Traffic control for either ingress rate-policing and egress shaping is supported on point-to-point pseudowire services. The ingress policy filters all packets arriving on the attachment circuit can be filtered before being sending them over the pseudowire. Egress shaping is performed on all packets arriving on the pseudowire before being sent out on the attachment circuit. The policy commands for pseudowires are:

- **policy-map ac-port**
- **policy-map ac-vlan**

To enforce a policy for a given pseudowire attachment circuit on a physical Ethernet port, use **policy-map ac-port**.

Figure 39 Policy Map on Pseudowire Port

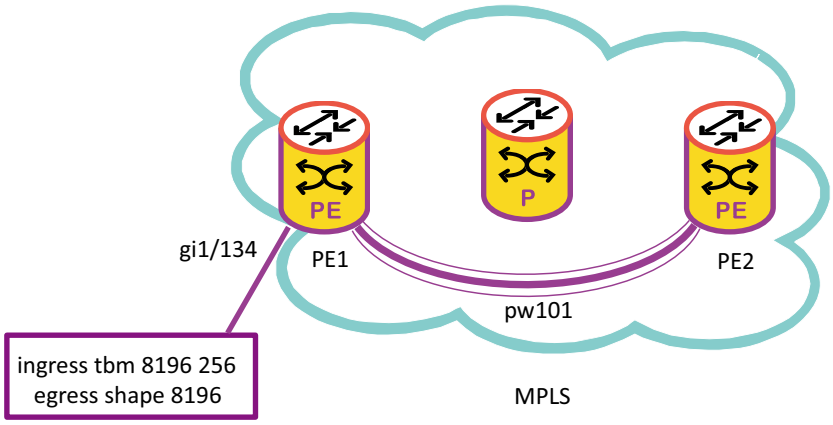


To configure the policy pw100-bw on pseudowire pw-100 for a port gigabitethernet1, do the following:

```
PE2#configure terminal
PE2(config)# interface gigabitethernet1
PE2(config-if-gi)# policy pw100-bw condition ingress tbm 256 32
PE2(config-if-gi)# policy pw100-bw condition egress shape 256
PE2(config-if-gi)# policy-map ac-port policy pw100-bw
PE2(config-if-gi)# mpls-l2-circuit pw100 ethernet
```

To enforce a policy for a given pseudowire attachment circuit that is bound to a specific VLAN ID on a physical Ethernet port, use policy-map ac-vlan.

Figure 40 Policy Map on Pseudowire VLAN



To configure policy pw101-bw on pseudowire pw101 for VLAN 1234, do the following:

```
PE2#configure terminal
PE2(config)# interface gigabitethernet1
PE2(config-if-gi)# policy pw101-bw condition ingress tbm 8196 256
PE2(config-if-gi)# policy pw101-bw condition egress shape 8196
PE2(config-if-gi)# policy-map ac-vlan 1234 policy pw101-bw
PE2(config-if-gi)# mpls-l2-circuit pw101 vlan 1234 ethernet
```

Targeted LDP

Targeted LDP is used as the signalling protocol between the two PE routers hosting the two virtual circuits implementing the endpoints of the pseudo-wire.

The signaling procedure includes distribution of VC labels, pseudowire configuration (e.g. control-word) as well as attachment circuit information.

To configure a pseudowire's endpoints using targeted LDP, use the following iBOS commands:

```
ibos(config)# router ldp
ibos(config)# transport-address ipv4 <source IP>
ibos(config)# targeted-peer ipv4 <destination IP>
ibos(config)# exit
```

Configuring EoMPLS

To configure EoMPLS, the procedure is as follows:

1. Perform basic MPLS configuration on the PE and P routers.

For details on basic MPLS configuration, see "LDP Configuration Examples" on page 213

2. Create a virtual circuit with a unique identifier and specify the IP address of the remote endpoint. The remote endpoint address is used by LDP to setup a targeted LDP session. If required, configure the VC with the control-word (CW) capability.
3. Bind an interface to the virtual circuit in one of the two ways:

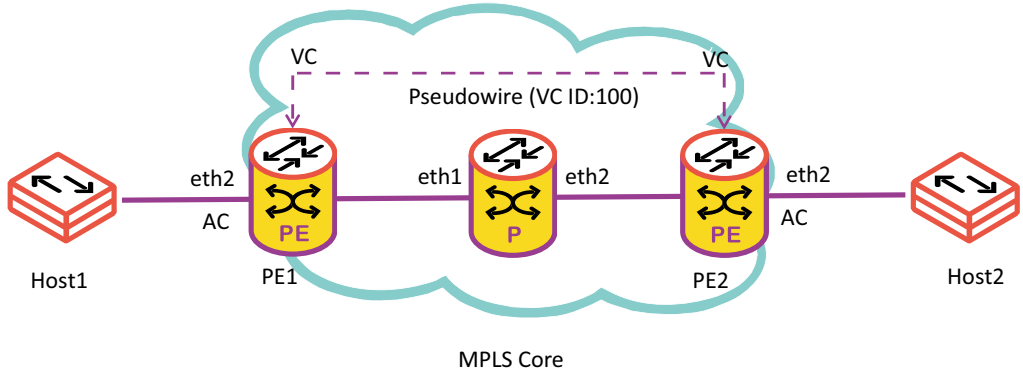
a. Bind a physical Ethernet port to the virtual circuit. All traffic received on the port will be passed to the virtual circuit and tunneled to the remote endpoint. This is known as port mode.

b. Bind a physical Ethernet port together with a VLAN identifier to the virtual circuit. All traffic received on the port tagged with the specified VLAN identifier will be passed to the virtual circuit and tunneled to the remote endpoint. This is known as VLAN mode.

Note: *Only symmetrical configuration is supported, i.e. both PE routers must use the same configuration.*

4. When the virtual circuit has been configured and bound to an attachment circuit, LDP will start a targeted LDP session to setup the pseudowire.
5. Verify that the pseudowire is up and running.

Figure 41 EoMPLS Configuration Example



EoMPLS Configuration Example

For the above example, configure router P as follows:

1. Configure IGP (OSPF) on the router P:

```
P#configure terminal
P(config)# interface loopback0
P(config-lo)# ip address 2.2.2.2/32
P(config-lo)# exit
P(config)# interface ethernet1
P(config-eth)# ip address 10.1.1.2/24
P(config-eth)# exit
P(config)# interface eth2
P(config-eth)# ip address 20.1.1.2/24
P(config-eth)# exit
P(config)# router ospf 100
P(config)# network 10.1.1.0/24 area 0
P(config)# network 20.1.1.0/24 area 0
P(config)# network 2.2.2.2/32 area 0
```

2. Configure MPLS and LDP on the router P:

```
P# configure terminal
P(config)# router ldp
P(config)# transport-address ipv4 2.2.2.2
P(config)# exit
P(config)# interface eth1
```

```
P(config-eth)# label-switching
P(config-eth)# enable-ldp ipv4
P(config)# exit
P(config-eth)# interface eth2
P(config-eth)# label-switching
P(config-eth)# enable-ldp ipv4
P(config)# exit
```

To configure router PE1, do the following:

1. Configure IGP (OSPF):

```
PE1#configure terminal
PE1(config)# interface loopback0
PE1(config-lo)# ip address 1.1.1.1/32
PE1(config-lo)# exit
PE1(config)# interface ethernet2
PE1(config-eth)# ip address 10.1.1.1/24
PE1(config-eth)# exit
PE1(config)# router ospf 100
PE1(config)# network 10.1.1.0/24 area 0
PE1(config)# network 1.1.1.1/32 area 0
```

2. Configure MPLS and LDP:

```
PE1# configure terminal
PE1(config)# router ldp
PE1(config)# transport-address ipv4 1.1.1.1
PE1(config)# targeted-peer ipv4 3.3.3.3
PE1(config)# exit
PE1(config)# interface ethernet2
PE1(config-eth)# label-switching
PE1(config-eth)# enable-ldp ipv4
PE1(config)# exit
```

3. Configure virtual circuit (with optional control-word):

```
PE1# configure terminal
PE1(config)# mpls l2-circuit t1 100 3.3.3.3 [control-word]
```

4. Bind the attachment circuit to either a port or to aVLAN on the VC:

   a. Bind attachment circuit to VC (port mode):

```
PE1# configure terminal
PE1(config)# interface ethernet1
PE1(config)# mpls-l2-circuit t1
```

   b. Bind attachment circuit to VC (VLAN mode):

```
PE1# configure terminal
PE1(config)# interface ethernet1
PE1(config)# mpls-l2-circuit t1 vlan 2
```

To configure router PE2, do the following:

1. Configure IGP (OSPF):

```
PE2#configure terminal
PE2(config)# interface loopback0
PE2(config-lo)# ip address 3.3.3.3/32
PE2(config-lo)# exit
PE2(config)# interface ethernet2
PE2(config-eth)# ip address 201.1.1.3/24
PE2(config-eth)# exit
PE2(config)# router ospf 100
PE2(config)# network 20.1.1.0/24 area 0
PE2(config)# network 3.3.3.3/32 area 0
```

2. Configure MPLS and LDP:

```
PE2# configure terminal
PE2(config)# router ldp
PE2(config)# transport-address ipv4 3.3.3.3
PE2(config)# targeted-peer ipv4 1.1.1.1
PE2(config)# exit
PE2(config)# interface ethernet1
PE2(config-eth)# label-switching
PE2(config-eth)# enable-ldp ipv4
```

3. Configure virtual circuit (with optional control-word):

```
PE2# configure terminal
PE2(config)# mpls l2-circuit t1 100 1.1.1.1 [control-word]
```

4. Bind attachment circuit to VC (port mode):

```
PE2# configure terminal
PE2(config)# interface ethernet2
PE2(config-eth)# mpls-l2-circuit t1
```

5. Bind attachment circuit to VC (VLAN mode):

```
PE2# configure terminal
PE2(config)# interface ethernet1
PE2(config-eth)# mpls-l2-circuit t1 vlan 2
```

MPLS Traffic Class

This section describes configuration of the MPLS Traffic Class (TC) feature in IBOS.

Note: *The MPLS TC field was formerly known as MPLS Experimental (EXP) field [RFC3032] but was later renamed MPLS TC [RFC5462].*

From a traffic management and QoS perspective. it may be desirable to:

1. Map traffic in accordance with IP DSCP or VLAN PCP to MPLS TC and appropriate outgoing queue when said traffic enters the MPLS domain and is tunnelled over a MPLS Label Switched Path (LSP)
2. Assign appropriate priority to incoming MPLS packets using the TC field from the topmost MPLS header
3. Assign appropriate egress priority when performing MPLS forwarding
4. Assign appropriate egress priority and (depending on LSP model) optionally rewrite IP DSCP or VLAN PCP when traffic leaves the MPLS domain (exits a LSP).

To take full advantage of the TC filtering and priority assignment, the following is required:

1. Configure MPLS TC Trust on all interfaces enabled for MPLS operation (i.e. label-switching). This feature is by default disabled:
   a. when disabled, all received MPLS packets will be processed using the default priority and the packets are enqueued in the same internal queue
   b. if this feature is enabled, the hardware will classify and assign priority according to the value of the MPLS TC field in the topmost MPLS header and the configured TC to priority map
2. Configure MPLS TC to priority mapping (or use the default mapping)

3. Configure MPLS TC to IP DSCP mapping (or use the default mapping)
4. Configure MPLS TC to VLAN PCP mapping (or use the default mapping)
5. Configure IP DSCP and VLAN PCP (dot1p) for non-MPLS interfaces, as well as policies and conditioners. While not directly related to MPLS TC configuration, but will affect the assignment of the MPLS TC field if the traffic (IP or VLAN) is to be tunnelled over an MPLS core.

Note: *Items 2 to 4 are global system settings, while item 1 (MPLS TC Trust mode) is configured per interface.*

Configure MPLS Traffic Class Trust

MPLS TC trust mode is configured on an interface to enable MPLS TC filtering and priority assignment according to the value of the TC field in the topmost MPLS header. The mapping of MPLS TC to priority is configured with the **cos map tclass** command.

```
mpc# conf t
mpc(config)# interface tengigabitethernet0/1
mpc(config-if-te)# tclass trust
```

To disable this functionality, use the negate command **no** with the configuration command, i.e. **no tclass trust**.

Configure MPLS TC to Priority Mapping

The MPLS TC to priority mapping is used by the hardware when MPLS TC Trust is enabled to assign priority to incoming MPLS packets.

This mapping is also used to map IP DSCP or VLAN PCP to MPLS TC when traffic is entering the MPLS domain, i.e. when an IP packet is classified and assigned a priority, this priority is then used to determine the MPLS TC value to be used when the IP packet is sent over an LSP.

The commands **cos map dscp** and **cos map dot1p** are used to map priority for incoming traffic. For more information on these commands, refer *iBOS Command Reference*.

To configure MPLS TC to priority mapping, use the **cos map tclass queue** command:

```
mpc# conf t
mpc(config)# cos map tclass <0-7> queue <0-7>

The table below shows the default TC to priority queue mapping:

TCLASS queue    TCLASS queue    TCLASS queue    TCLASS queue
------ ------   ------ ------   ------ ------   ------ ------
     0      0        1      1        2      2        3      3
     4      4        5      5        6      6        7      7
```

Configure MPLS TC to IP DSCP Mapping

The MPLS TC to IP DSCP mapping is used when IP traffic tunneled over a MPLS core exits the LSP and the LSP is operating in uniform LSP model (configured with **mpls propagate-ttl**) and in accordance with RFC3270 and RFC3443.

In this configuration, the DSCP in the IP header may be updated according to the received MPLS TC value in the topmost MPLS header to reflect any traffic class change (as well as TTL change) derived from packet transport over the MPLS core.

This feature may be used for basic IPv4 transport over MPLS or for IPv4 L3VPN over MPLS.

To configure TC to IP DSCP mapping, use the **cos map tclass dscp** command:

```
mpc# conf t
mpc(config)# cos map tclass <0-7> dscp <0-63>

The table below shows the default TC to IP DSCP mapping:

TCLASS  DSCP    TCLASS  DSCP    TCLASS  DSCP    TCLASS  DSCP
------ ------   ------ ------   ------ ------   ------ ------
     0     0         1     8         2    16         3    24
     4    32         5    40         6    48         7    56
```

Configure MPLS TC to VLAN PCP Mapping

The MPLS TC to VLAN PCP Mapping is used when VPWS traffic tunnelled over a MPLS core exits the pseudowire and the tunnel LSP is operating in uniform LSP model (**mpls propagate-ttl**) and in accordance with RFC3270 and RFC3443.

In this configuration, the VLAN PCP in the VLAN header may be updated according to the received MPLS TC value in the topmost MPLS header to reflect any traffic class change derived from packet transport over the MPLS core.

This feature may be used for VPWS (EoMPLS) when the pseudowire is bound to an attachment circuit operating in VLAN (tagged) mode. Further, the physical port used as attachment circuit must be enabled to trust VLAN PCP (configured with **dot1p trust** command).

Also, the priority will be set to default if VLAN PCP trust is not enabled or if the attachment circuit is operating in port mode. In this case, the MPLS TC setting for priority (queue) 0 will be used in the outgoing MPLS packets when the pseudowire payload is tunnelled over the MPLS core.

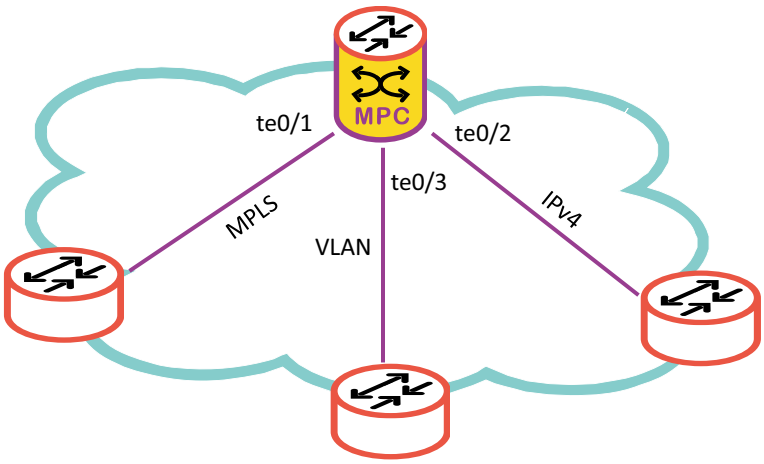
To configure TC to VLAN PCP (dot1p) mapping, use the **cos map tclass dot1p** command:

```
mpc# conf t
mpc(config)# cos map tclass <0-7> dot1p <0-7>

The table below shows the default TC to VLAN PCP mapping:

TCLASS   PCP    TCLASS   PCP    TCLASS   PCP    TCLASS   PCP
------ ------   ------ ------   ------ ------   ------ ------
     0     0         1     1         2     2         3     3
     4     4         5     5         6     6         7     7
```

MPLS Traffic Class Example

This example shows the configuration necessary to setup a P router (or Label Switch Router - LSR) to use the MPLS traffic class feature. The same procedure also applies to a PE router (or Label Edge Router - LER).

Figure 42 MPLS Traffic Class Example



1. Configure the MPLS TC to a priority queue map. In this example, we alter the default setting to map TC0 to queue 7, TC1 to queue 6, and so on.

```
mpc# conf t
mpc(config)# cos map tclass 0 queue 7
mpc(config)# cos map tclass 1 queue 6
mpc(config)# cos map tclass 2 queue 5
mpc(config)# cos map tclass 3 queue 4
mpc(config)# cos map tclass 4 queue 3
mpc(config)# cos map tclass 5 queue 2
mpc(config)# cos map tclass 6 queue 1
mpc(config)# cos map tclass 7 queue 0
```

2. Enable MPLS TC trust on MPLS enabled interfaces. Packets received on those interfaces will use the MPLS TC value in the topmost header to assign a priority to the packet in accordance with the mapping table configured in step 1).

```
mpc# conf t
mpc(config)# interface tengigabitethernet0/1
mpc(config-if-te)# tclass trust
mpc(config-if-te)# exit
mpc(config)# interface tengigabitethernet0/2
mpc(config-if-te)# ip dscp trust
mpc(config-if-te)# exit
mpc(config)# interface tengigabitethernet0/3
mpc(config-if-te)# dot1p trust
mpc(config-if-te)# exit
```

An interface is enabled for MPLS operation if **label-switching** has been configured. All MPLS packets received on an interface will be discarded if the interface is not enabled to do label switching.

3. Configure MPLS TC to IP DSCP mapping. This step is required to support the correct mapping of MPLS TC to IP DSCP when both the MPLS Label Switched Path (LSP) is operating in uniform model, and the tunnelled traffic is plain IPv4 traffic or IPv4 L3VPN traffic.

```
mpc# conf t
mpc(config)# cos map tclass 0 dscp 11
mpc(config)# cos map tclass 1 dscp 12
mpc(config)# cos map tclass 2 dscp 21
mpc(config)# cos map tclass 3 dscp 22
mpc(config)# cos map tclass 4 dscp 31
mpc(config)# cos map tclass 5 dscp 32
mpc(config)# cos map tclass 6 dscp 41
mpc(config)# cos map tclass 7 dscp 44
```

The MPLS TC to IP DSCP mapping is a global configuration and it will be used by all VRF instances in case of IPv4 L3VPN, i.e. the same mapping is used by all VPNs.

4. Configure MPLS TC to VLAN PCP mapping. This step is required to support the correct mapping of MPLS TC to VLAN PCP when the MPLS Label Switched Path (LSP) is operating in uniform model and the tunnelled traffic is a pseudowire with a VLAN attachment circuit.

```
mpc# conf t
mpc(config)# cos map tclass 0 dot1p 7
mpc(config)# cos map tclass 1 dot1p 6
mpc(config)# cos map tclass 2 dot1p 5
mpc(config)# cos map tclass 3 dot1p 4
mpc(config)# cos map tclass 4 dot1p 3
mpc(config)# cos map tclass 5 dot1p 2
mpc(config)# cos map tclass 6 dot1p 1
mpc(config)# cos map tclass 7 dot1p 0
```

The MPLS TC to VLAN PCP mapping is global and will be used by all pseudowire instances.

BGP Configuration

BGP configuration is described in the following sections:

- “Introduction to BGP” on page 240
- “BGP Routing” on page 241
- “Advanced iBGP” on page 248
- “BGP Messages” on page 254
- “BGP Configuration Examples” on page 256
- “Extended BGP” on page 330

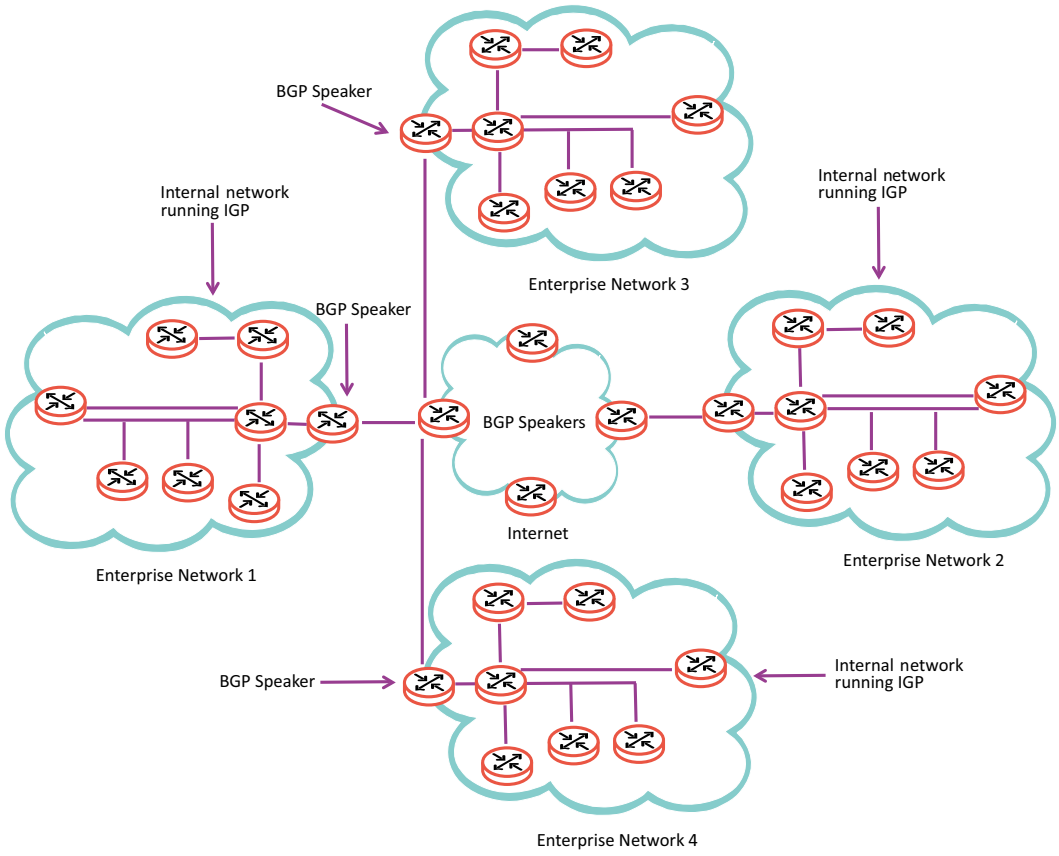
Introduction to BGP

The Border Gateway Protocol (BGP) is used for routing exchange between different autonomous systems (AS). The AS is a network with a specific routing policy.

BGP can also be used for internal routing exchange in combination with an IGP. BGP is not a replacement for IGP, such as OSPF or IS-IS, but in larger networks BGP can be used to reduce the amount of prefixes carried by the IGP to achieve faster convergence.

Routing exchange with BGP between routers in the same AS is referred to internal BGP (iBGP) while routing exchange with BGP between routers in different AS is called external BGP (eBGP).

Figure 43 BGP in the Network



Unlike IGP protocols where routers are able to discover each other by use of hello-packets and routing updates (for example RIP and OSPF), BGP must be configured. BGP has no dynamic establishment of peers and requires static configuration of all parameters including filters and other information to define the routing information to be exchanged.

The BGP protocol is based on TCP connections between routers participating in the routing exchange. TCP provides reliability and it is not necessary to implement configuration for re-transmission, acknowledgment or sequencing because TCP already handles these features. In addition, BGP maintains its own keep-alive mechanism for faster detection of peer unavailability.

The iBOS BGP implementation is BGP4 for IPv4 and includes classless inter-domain routing (CIDR) and route aggregation.

BGP Routing

There are two types of BGP routing; Internal BGP (iBGP) and External BGP (eBGP).

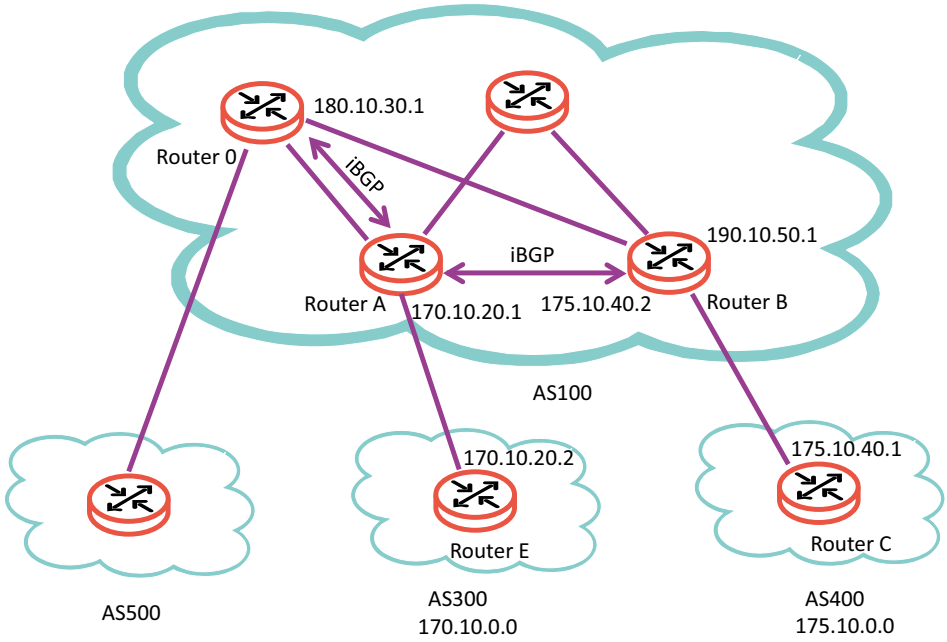
Internal BGP Routing

Internal BGP (iBGP) is used between routers in the same AS to carry large amounts of routing information. In modern networks that include MPLS, BGP can also be used to carry information about routing domains within the network.

When the network has multiple egress points where external BGP (eBGP) is used, it is important that routers within the network have the correct routing table to determine which egress point to use depending on the destination. Redistribution of BGP into the IGP protocol is not recommended as BGP usually contains much more routing information than IGP protocols are designed to handle. Therefore, BGP is preferably used between routers in the network to convey this information and to allow the best routing decision to be made about each packet.

iBGP must be full mesh. This means that all routers in the AS that use BGP must be configured with one peer to every BGP device in the network. The reason for this full mesh peering is to prevent that BGP routes learned from another router in the same AS are retransmitted to routers in the same AS.

Figure 44 Fully Meshed iBGP in the Network



To reduce the impact of this full mesh peering requirement, iBGP is usually set up with route reflectors. Using route reflectors means that iBGP devices to only peer with the route reflector instead of peering with every other router in the network. Another method to reduce the number of peers for each iBGP router is to use BGP Confederations.

iBGP peers are typically established between router loopback interfaces. When the loopback interface is redistributed in the IGP routing, the IGP protocol can converge

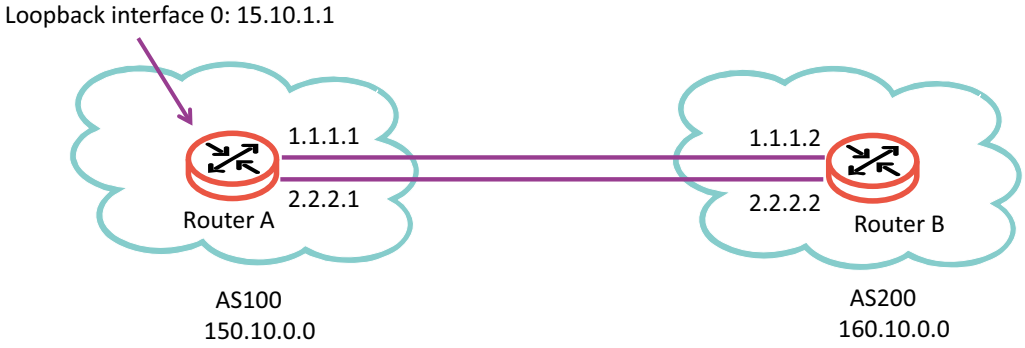
and for available backup connections, the BGP peer can continue to be active even when a link in the network has been lost. An update source configuration statement is used to set up the source interface for BGP peers.

External BGP Routing

When two BGP routers belong to different AS, they communicate with eBGP. In such cases, the link between the routers are either a shared media (such as a Gigabit Ethernet segment on an exchange point) or a point-to-point connection. In these cases, peering is typically done with the interface address on the remote side.

If, however, multiple connections exist, such as multiple parallel WAN connections, eBGP peering using loopback interfaces is possible.

Figure 45 eBGP Routing in the Network



Comparison BGP and IGP

BGP carries routing information. The information for a specific destination includes the next-hop address of the router in the network to which the packet must be sent in order to reach the destination.

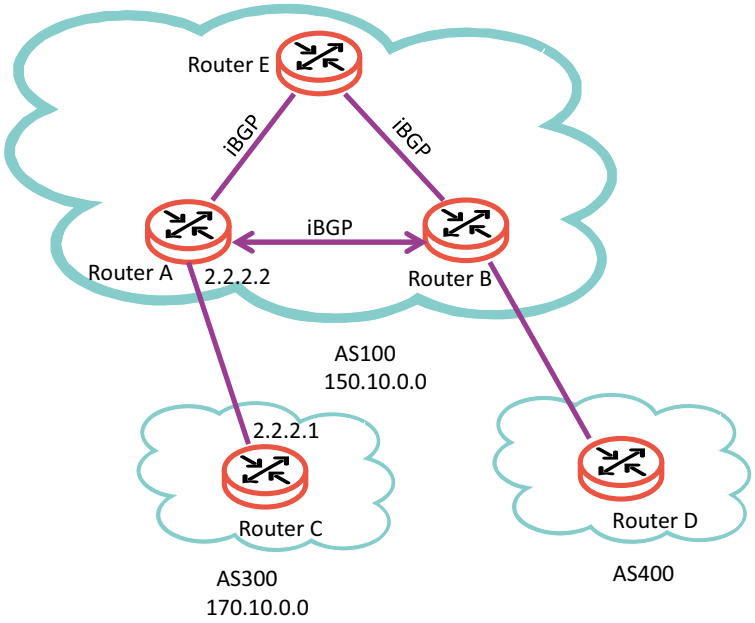
With IGP, the resulting routing table states the next-hop router along the path to the destination. BGP, however, carries indirect routing information. A BGP route next-hop address is normally the address of the router that announced the route with BGP. To get to the destination of a BGP route, the packet must first be sent to the router that announced it by BGP. Therefore, the next-hop address of a BGP route is not necessarily the address of a directly connected neighbour. The next-hop address is usually the loopback interface or the egress interface of the next-hop router.

With IGP, the next-hop address needs to be known so that each router can forward a packet for a BGP route towards its next hop address as indicated by the IGP.

BGP Synchronization

Because BGP relies on the IGP to determine the actual forwarding path, all routers between the sender of a packet and the destination must have the correct routing table. For example, all routers in the central part of the network must have received correct routing information, otherwise packets may be dropped or sent along the wrong path. Therefore, BGP and the IGP need to be synchronized.

Figure 46 BGP Synchronization



Synchronization may be a problem if BGP routes are redistributed into an IGP. A common practice today is to avoid such redistribution and only rely on indirect routing with the IGP carrying links and loopbacks within the network

BGP Filtering of Routes

BGP carries a lot of information about each route. Most of this information can be used to filter routes and select which routes to use.

BGP filters are typically used in eBGP to control the routing information that is announced to, and received from, peers.

The general principle is to not trust peering partners to control what is announced by BGP. Filtering enhances the stability of the network.

RIPE and other organization operate routing registries where ISPs are encouraged to register what routes they are announcing at various exchange points. In this way, peering partners can go to one source and construct routing filters according to the information in a routing registry.

BGP Parameters

BGP is considered one of the most complex routing protocols in use due to the large number of parameters that accompany every route. Also, many devices are involved in the final path selection so there is not a single metric value in BGP.

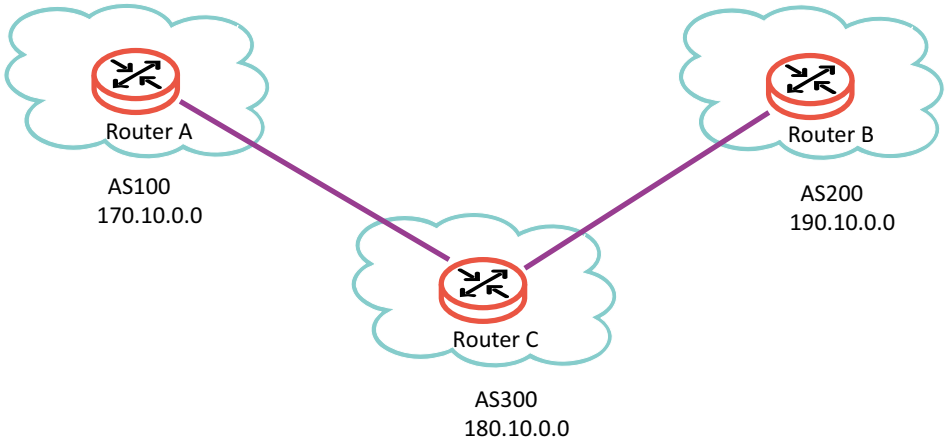
AS Paths

Every route announced by BGP has an AS-path. The AS-path is built by adding the AS number to the route every time the route is announced with eBGP. In this way, it is possible to determine the ASs that the traffic must pass through to reach the destination.

A rule of thumb is the shorter the AS-path. the better the route.

In the following figure, the path for network 170.10.0.0 from router B is 300 100. If AS200 would announce the route to yet another AS, 200 would be added to the AS-path in that announcement resulting in an AS-path of 300 100 200.

Figure 47 BGP AS-Path



BGP Route Origin

A route can be learned by BGP in three ways.

- Interior means that the route was administratively configured to be originated in BGP
- Exterior means that the route was learned by redistribution from eBGP. It is uncommon to see any routes learned this way
- Incomplete means that the origin of the route is unknown. This is typically the result if routing from an IGP protocol is redistributed into BGP for announcement.

Generally speaking, Interior is better than Exterior, and Exterior is better than Incomplete.

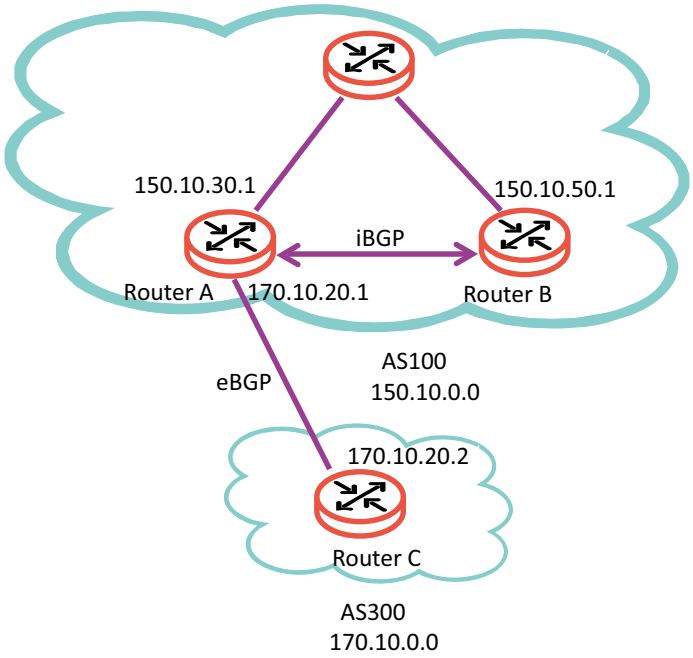
BGP next-hop

As mentioned above, the next-hop address of a BGP route is the address of the router to which packets must be forwarded. For AS external routes, it is typically the address of the eBGP neighbour (in the remote AS). This can be ineffective because the eBGP neighbour address must be known in the IGP.

If the router in the AS knows where to forward packets to the destination, the feature **next-hop-self** allows the router to set itself as the next-hop rather than the eBGP neighbour.

In the following figure, the next-hop address for 170.10.0.0 in AS 300 would be 172.10.20.2. If this address is not known in the IGP protocol running in Router B, communication will not work. The solution is to configure Router A as next-hop-self for the BGP routes learned from router C.

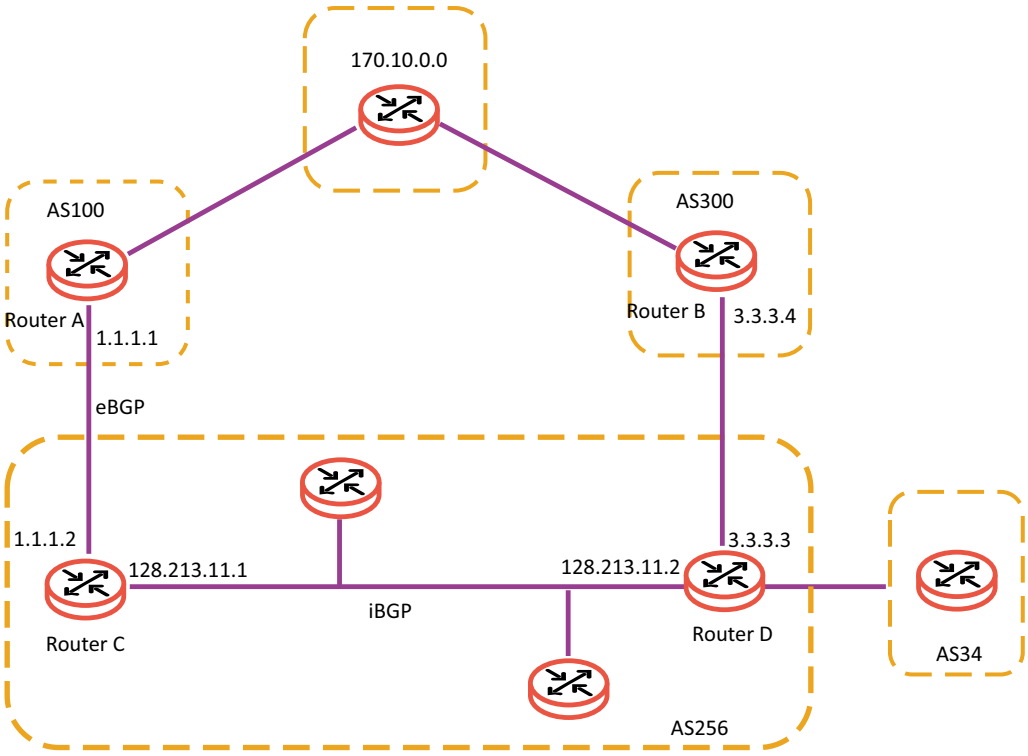
Figure 48 BGP AS-path



BGP local-preference

Local-preference is an administrative parameter than allows the network administrator to configure preference for a certain path in relation to other paths when there are multiple routes to the same destination. The parameter is configured to a numerical value in the range 1-4294967295, and the default value is 100. A higher value reflects a better route selection.

Figure 49 BGP local-preference



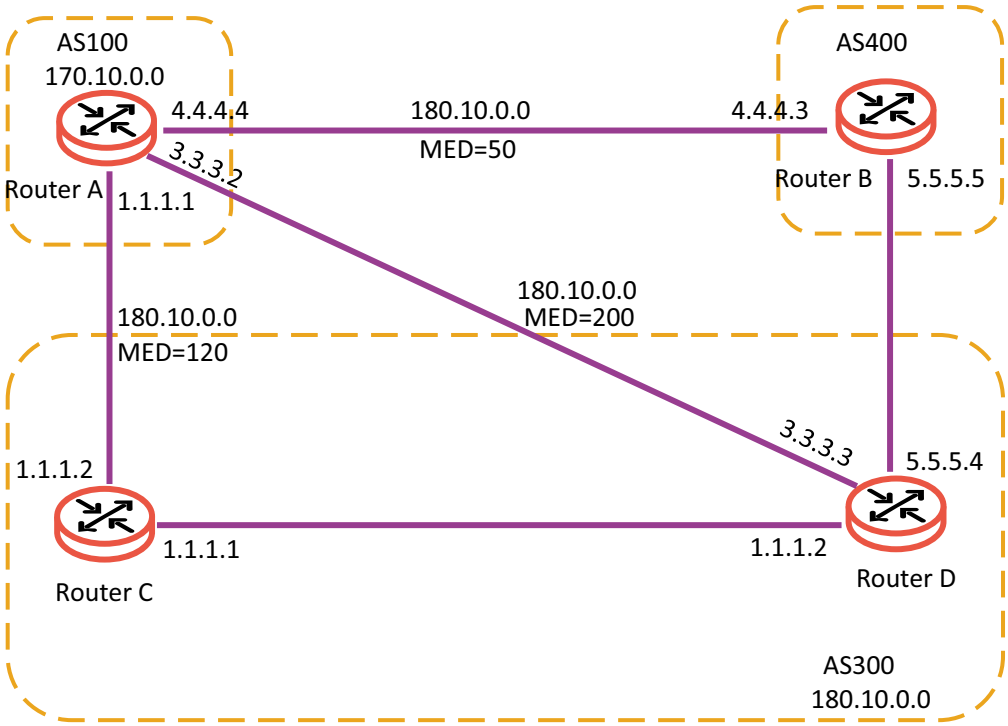
In the example, local-preference may be used to ensure that traffic from AS256 to 170.10.0.0 pass through AS300 by setting the local preference for 170.10.0.0 to a higher value when received in Router D.

Commands used to configure BGP local preference are **bgp default local-preference** and **bgp g-shut-local-preference**.

BGP Multi Exit Discriminator

If there are multiple connections between two AS, the MED value can be set on announced routes to indicate to the routers in the other AS the preferred connections for sending traffic towards a specific destination.

Figure 50 BGP MED



MED values only survive into the next AS. The lower the value the better. If the MED attribute is missing in a received update message iBOS will take the default value 0 (zero) when it updates the internal routing database.

MED values are only used in the path selection for routes from the same AS unless the router is configure to always compared MED.

Also note that there are configuration related to MED applicable if confederations are used in the network.

BGP Community

Community is a method of marking routes with a tag. This tag is useful to control and filter routing. Applying and using communities for filtering of routes is one of the most powerful features of BGP.

In addition to certain well-known communities, it is possible to design the routing policy of the network based on your own communities.

Marking a set of routes with a community makes it possible to apply a routing policy to that community and thereby indirectly to all the marked routes.

A community consists of routes that share a common attribute. It is possible for a route to belong to multiple communities and this can be determined through configuration.

The community is set on the route at the ingress point which then can be used to perform numerous filtering actions on the routes at other points in the network. For instance, marking all routes received from a specific AS with a unique community allows those routes to be selected in another part of the network by only specifying the community.

For example, if an ISP offers a transit service, the ASs that subscribe to the transit service get a specific community on all routes announced by the service's transit-customer-ASs. On exchange points and private peers, the transit provider can construct an egress route filter in BGP that permits (announces) all routes marked with the transit-community. If one of the customer ASs cease to subscribe to the service if a customer AS adds more prefixes to their routing, only the incoming route filter needs to be updated to remove the transit-community from the received routes. The routes will not be announced on any peer that requires the community to permit the route. In a network with numerous peering-points, the community method greatly reduces the amount of manual configuration.

Communities may also be exchanged with eBGP. Many transit providers allow their customers to set communities on announced routes and control how the routes are announced to avoid any interaction from the transit provider staff.

There is a set of well-known, predetermined communities with specific functions that can be defined using BGP.

To allow better structure for community space, iBOS supports the BGP Extended Communities Attribute that extends the range and adds a type field. This support also includes the opaque extended community and four-byte AS specific extended community attribute.

Note: *Changing between 2 and 4 byte AS requires reconfiguration of any Virtual Routing and Forwarding (VRF) in the router as a consequence because the pre-existing route distinguisher (RD) will be configured with the current (2-byte or 4-byte) capability and must be reconfigured if the capability is changed.*

No-export Community

The no-export community means that the route with the community should not be announced to other eBGP peers.

No-advertise Community

The no-advertise community means that the route should not be announced to any peer

Internet Community

The internet community means that the route should be announced to all routers in the network.

Selecting the Best Route

In BGP, the following step-by-step evaluations are performed for two identical routes to determine which one that is the best.

1. If the next-hop is unreachable by IGP, drop the route
2. Prefer the path with highest local-preference
3. Prefer a path originated by BGP on this router
4. Prefer the path with the shortest AS-path
5. Prefer the path with the lowest origin (Interior < Exterior < Incomplete)
6. Prefer the path with the lowest MED
7. Prefer the eBGP route over the iBGP route
8. Prefer the path with the closest IGP neighbour
9. Prefer the path with the lowest BGP router-ID

The last option selects a route in the odd situation where all the other options are equal.

Advanced iBGP

As the network grows, the full mesh requirement for iBGP becomes a problem. In a network with N BGP speaking routers, there are N^2 iBGP peers. The time consumed to update all routers with new information makes convergence slow, especially when there are hundreds of thousands of routes to calculate.

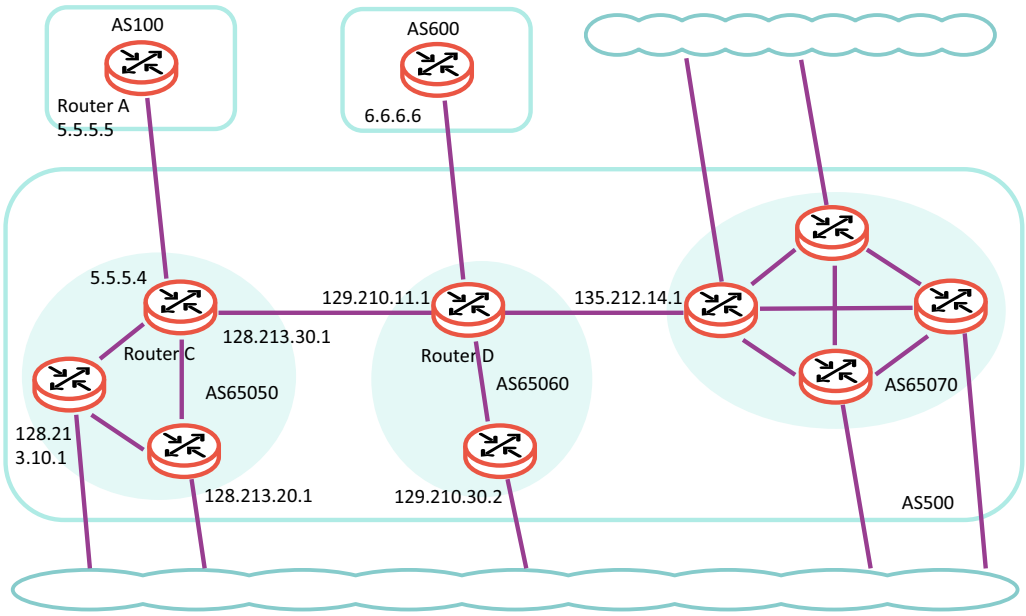
Two techniques to resolve these problems exist; **route reflectors** and **confederations**. The most popular solution is route reflectors.

Router Reflectors

A route reflector is a router that deliberately ignores the rule about re-transmitting routing information to other peers in the same AS. However, it only does this after deliberate configuration.

Route reflection is similar to the use of areas in OSPF and IS-IS. One or more routers in the topology are configured as route reflectors and other routers beneath the route reflectors in the topology are route reflector clients.

Figure 51 Route Reflector



When a route reflector receives a route it is handled as follows:

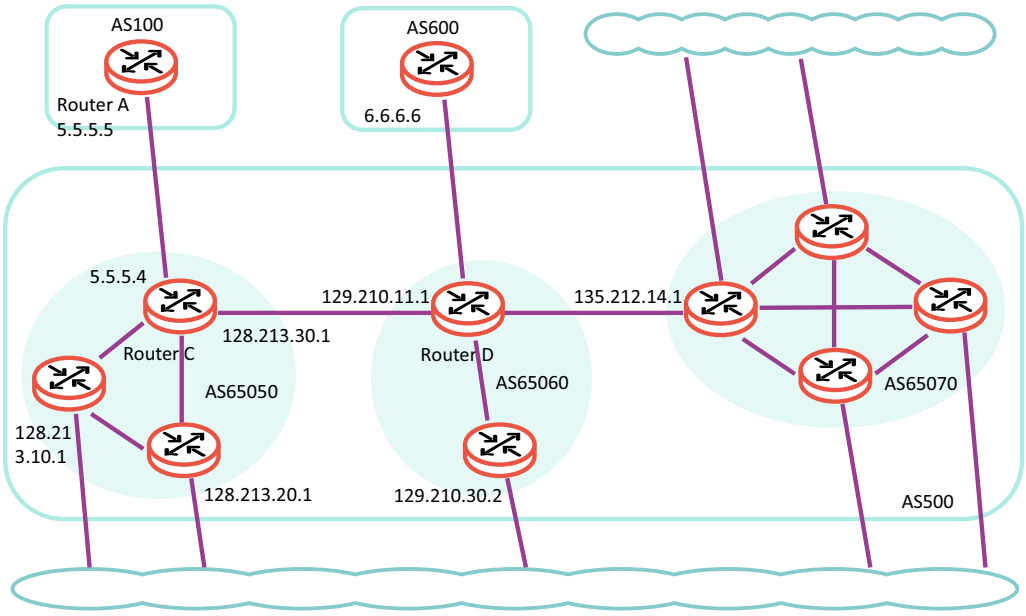
- A route from an eBGP peer is advertised to all route reflector clients and non-client peers for the route reflector
- A route from a non-client peer is advertised to all clients
- A route from a client is advertised to all clients and non-client peers. This means that the clients do not need to be fully meshed

For example, a pair of core layer routers in a POP is configured as route reflectors. All distribution routers in the POP are route reflector clients and do not participate in the network full mesh because they receive all routing information from their route reflectors. The distribution routers only peer with the core route reflectors while the core route reflectors peer in a full-mesh iBGP with all other route reflectors and non-route reflector clients in the network.

BGP Confederation

Confederation is another method of reducing the size of the BGP full mesh. Confederations resemble the area topology of OSPF or IS-IS that use eBGP instead.

Figure 52 BGP Confederations



When running BGP confederations, routers in different parts of the network are configured to belong to a private-AS number. Routers belonging to the same private-AS peer iBGP with full mesh. Between the private-ASs, eBGP communications is used. In essence, each POP or other sub-part of the network becomes its own AS.The only difference from a standard AS is that when such a sub-part communicates with a real external AS, the true AS is replaced and the internal private-AS hierarchy is hidden. To the world, the AS appears as a single normal network.

Asymmetrical Traffic

ASs establish peering with many other ASs in a complex multi-dimensional hierarchy so asymmetric traffic patterns are common.

Many ISPs have multiple external connections. The preferred connection may not be symmetric which means that traffic from network A to network B will pass through one set of ISPs while traffic from network B to network A can pass through a different set of ISPs.

For troubleshooting routing problems, the **traceroute** utility is an invaluable tool but this often requires the cooperation of network engineers in distant ASs to track down a routing issue or the source of high delays to traffic.

A common approach from ISPs is “hot-potato routing”. Each packet is considered as a hot-potato to be disposed of as soon as possible so that the closest exchange point is selected to deliver each packet out of the network.

MD5 Verification

To enhance BGP security, an MD5 digest is carried in a TCP segment, which acts as a signature for the segment, to incorporate information known only to the connection end-points. To enable the MD5 flag, use the **neighbour password** command.

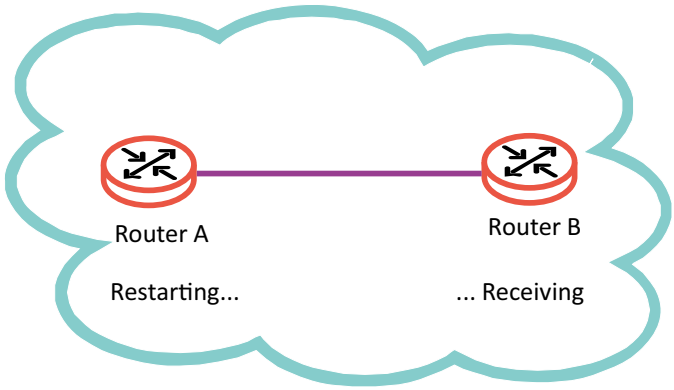
Graceful Restart

The graceful restart mechanism minimizes negative effects on routing caused by BGP restarts. The graceful restart feature allows a restarting router and its neighbours to continue forwarding packets without disrupting network performance. Neighbouring routers assist in the restart so that the restarting router can quickly resume full operation. Graceful reset is a refinement of graceful restart that helps to ensure smooth restarts when a configuration change forces a BGP peer to reset.

Without the graceful restart capability, all BGP peers during a BGP restart will detect that a session has gone down and come back up. This will cause an invalidation of the associated portion of the IP forwarding cache, a BGP route re-computation, and generation of BGP routing updates. Forwarding tables will become corrupted and unstable.

Graceful restart provides a way for BGP to minimize the negative effects on routing caused by a BGP restart. It allows the restarting BGP router to temporarily retain routing information and continue forwarding packets when BGP restarts. In this way, the router continues to operate across the TCP connection even while the routing is rebuilding routing and forwarding tables.

Figure 53 Graceful Restart Router



In the figure above, two routers (Router A and Router B) support the graceful restart capability (capability code 64). The following describes the sequence of events during a graceful restart:

1. The BGP process in Router A restarts
2. Router B detects the TCP session failure with Router A and marks routes from A as stale in its RIB. It also starts the restart timer. The default time is 120 seconds.
3. Router A does not have BGP routes in its RIB at restart and must re-acquire them from its peer routers. However, it retains preserved stale routes in its FIB.
4. Once Router A re-establishes a BGP session with Router B from the open message sent from A-with the restart state bit set, a restart time of n, and forwarding state = set-Router B resets the restart timer and starts the stale-path timer. The default time is 360 seconds. During this time, Routers A and B continue to forward traffic using the last updated FIB table.
5. Router B begins sending update messages to Router A.

6. Router A starts an update delay timer. Once Router A gets an End-of-RIB (EOR) indication from all of its peers, it starts the BGP route selection process.
7. Router A begins sending update messages, which contain prefix information, to Router B.
8. Router A sends an EOR indication to Router B, so that Router B, in turn, can start its route selection process.
9. Once Router B completes its route selection process, any stale entries in BGP are refreshed with newer information or removed from the BGP RIB and FIB. Router B is now converged.
10. While Router B waits for an EOR, it also monitors the stale-path time. If the timer expires, all stale routes are removed, and normal BGP processes are started.

Note: *An EOR marker is an update message with an empty, withdrawn NLRI. It can be used by a BGP speaker to indicate to its peer the completion of the initial routing update after the session is established. For the IPv4 unicast address family, the EOR marker is an update message with the minimum length [BGP-4]. For any other address family, it is an update message that contains only the MP_UNREACH_NLRI attribute [BGP-MP] with no withdrawn routes for that <AFI, SAFI>*

The following commands are used for configuring graceful restart and graceful reset.

Command	Description
bgp graceful-restart	Enables BGP graceful-restart capabilities. The default restart time is 120 seconds, and the default stale-path time is 360 seconds.
neighbour capability graceful-restart	Configures the router to advertise the graceful restart capability to the neighbours.
restart bgp graceful	The BGP daemon (BGPd) notifies the routing engine it has shut down gracefully so that the routing engine preserves the routes installed by BGP.
bgp update-delay	Specifies the update-delay value for a graceful-restart capable router.
bgp graceful-restart graceful-reset	In Router mode, invokes graceful restart when a configuration change forces a peer reset.

For more detailed information, refer *iBOS Command Reference.*

Example Configuration for Graceful Restart and Graceful Reset

The following shows an example of using the above commands to configure a graceful restart and graceful reset.

```
iBOS# configure terminal
iBOS(config)# router bgp 100
iBOS(config-router)# bgp graceful-restart restart-time 150
iBOS(config-router)# bgp graceful-restart graceful-reset
iBOS(config-router)# neighbour 1.1.1.1 capability graceful-restart
iBOS# restart bgp graceful
```

Note: *The entire configuration should be saved before issuing the restart bgp graceful command. In addition, the* **bgp graceful-restart** *command must be enabled before the graceful-reset feature, as shown in the following example.*

BGP Peer Reset and Graceful Restart Events

All events that cause BGP peer reset (all-session reset) can trigger graceful restart. The following are the valid configuration change commands that cause a peer reset (all-session reset) and invoke graceful restart:

- **bgp router-id A.B.C.D**
- **no bgp router-id A.B.C.D**
- **bgp extended-asn-cap**
- **no bgp extended-asn-cap**

BGP and MPLS

iBOS supports BGP-MPLS VPNs for IPv4 and IPv6. This method lets service providers use an IP backbone to provide IPv4 and IPv6 VPN connectivity to their customers. Customer data packets are tunnelled through the backbone, so that core routers are unaware of IPv4 and IPv6 VPN routes.

The BGP module supports IPv4 and IPv6 VPN route exchange, and the routing engine supports IPv4 and IPv6 FTN and ILM tables required to tunnel customer data.

All CLIs specific to IPv4 VRF are applicable to IPv6 VRF, and all CLIs specific to the VPNv4 address-family are applicable to the VPNv6 address-family.

Inter AS BGP MPLS VPNs

iBOS also supports Provider Edge routers (PEs) in multi autonomous system (multi-AS) backbones, as follows:

- eBGP redistribution of labelled VPN-IPv6 routes from an AS to a neighbouring AS
- Multihop eBGP redistribution of labelled VPN-IPv6 routes between source and destination ASs, with eBGP redistribution of labelled IPv6 routes from an AS to a neighbouring AS

iBOS BGP VPN is a fully scalable implementation of IETF 2547-bis. iBOS BGP can be used either as a PE/P node in the Service provider network, or as CPE to connect to the Service provider edge using eBGP. Other iBOS routing protocols, such as OSPF, are also extended to work as Customer Edge-Provider Edge (CE-PE) routing protocols.

iBOS BGP uses multiple forwarding tables to provide network Isolation of VPN traffic inside the service provider network. A separate default forwarding table can be used to contain public Internet routes and site-based VRF tables for VPN routes. The PE router maintains a VRF table, per site, that shares the same routes. When an IP packet is received on a PE-CE interface, the corresponding VRF is used for the destination IP address lookup.

CE-PE Interaction

The CE devices learn and advertise routes to other sites in its VPNs. iBOS BGP and OSPF can be used for CEPE route exchange. BGP in PE installs the routes learned in the corresponding VRF table of the site, based on the incoming interface.

PE-PE Interaction

At the PE device, iBOS BGP installs the VPN routes that were learned from the attached CE in VRF tables. iBOS BGP uses Multiprotocol BGP (MP-BGP) to distribute VPN routes to other PE BGPs across the SP network. Routes that are exported to BGP from VRF tables are converted into unique addresses in the VPNv4/v6 address family using route distinguishers (RDs). The VPN-IP route is a 12-byte quantity that consists of an RD and the IPv4 address. iBOS BGP uses Route Targets to advertise by PE devices with the Router Target attribute. Every VRF is associated with a set of route targets. The import route target list specifies the routes that can be installed in a VRF. Routes advertised to other PE devices are tagged with the export route targets.

MPLS LSP

IPv4 signalled MPLS Label Switched Paths (LSPs) can be used without any specific tunnel configuration.

BGP Messages

There are four important BGP message types:

- OPEN
- UPDATE
- NOTIFICATION
- KEEPALIVE

OPEN Messages

An OPEN message from the BGP speaker is sent after a TCP connection is formed between the peers. This message contains the version, autonomous system number (ASN), hold time, BGP identifier, length of optional parameters, and optional parameters.

UPDATE Messages

UPDATE messages are exchanged between the BGP speakers, once the two BGP speakers have established neighbour adjacency. UPDATE messages typically contain the routes that must be advertised to the corresponding peer. It also contains routes that must be removed from the routing table. Hence, the UPDATE message controls the routing information that goes into the routing table.

NOTIFICATION Messages

When an error occurs in a BGP session, and the session must be torn down, NOTIFICATION messages are sent from the corresponding peer. Once the other peer receives the message, the TCP connection is torn down, and the speakers are no longer peers with each other.

KEEPALIVE Messages

This message serves as the connectivity check between BGP speakers. The value of this KEEPALIVE timer highly depends on the hold time advertised in the OPEN message. BGP requires that the KEEPALIVE time is one third of the hold time, that is, hold time is three times the KEEPALIVE time.

BGP Configuration Examples

Configuring a basic BGP network consists of a few required tasks and many optional tasks. A BGP routing process must be configured and BGP peers must be configured, preferably using the address family configuration model.

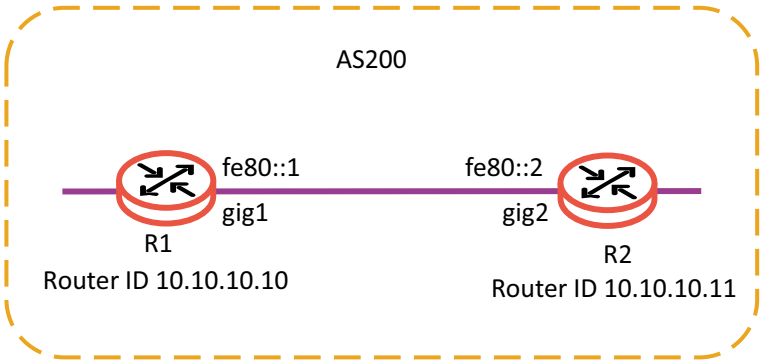
The BGP configuration is explained in the following:

- “Enable BGP Routers in the Same Autonomous System” on page 257
- “Enable BGP Between Different Autonomous Systems” on page 257
- “Configure AS Path Filtering” on page 258
- “Configure Route-Map” on page 258
- “Redistribution of Routing into BGP” on page 260
- “Configuring Route Reflector” on page 260
- “Configure Route-Maps” on page 261
- “Configure Route Reflectors” on page 263
- “Configure Route Server” on page 266
- “Configure BGP Confederations” on page 271
- “BGP Authentication” on page 274
- “Enable eBGP Multihop” on page 275
- “Redistribute with OSPF” on page 278
- “Add Multiple Instances of Same AS to Administer Route Selection” on page 279
- “Remove Multi-Exit Disc Attribute from Update Messages” on page 281
- “Configure BGP Four-Byte AS” on page 282
- “Configure BGP Extended Community Attribute” on page 285
- “Configure IPv6 Route Refresh” on page 291
- “Configure Nexthop Tracking” on page 294
- “” on page 296
- “” on page 299
- “Configure BGP-MPLS Graceful Restart” on page 301
- “” on page 305
- “Configure BGP Weight per Peer” on page 307
- “Configure BGP-OSPF as PE-CE Protocol for VPNs” on page 309
- “BGP Multipath for IPv4 and IPv6” on page 312
- “BGP Graceful Shutdown” on page 326

Enable BGP Routers in the Same Autonomous System

Figure 54 shows the minimum configuration required to enable BGP on an interface. R1 and R2 are two routers belonging to the same AS, AS200, that connect to network 10.10.10.0/24. First, define the routing process and the AS number to which the routers belong. Then, define BGP neighbours to start exchanging routing updates.

Figure 54 Enable BGP for Same AS



To enable BGP for the same AS, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router)# neighbour 10.10.10.11 activate` `iBOS(config-router)# neighbour 10.10.10.11 remote-as 200`	Enable TCP session for BGP neighbour. 10.10.10.11 is the IP address of the neighbour (R2), and 200 is the neighbour's AS number.

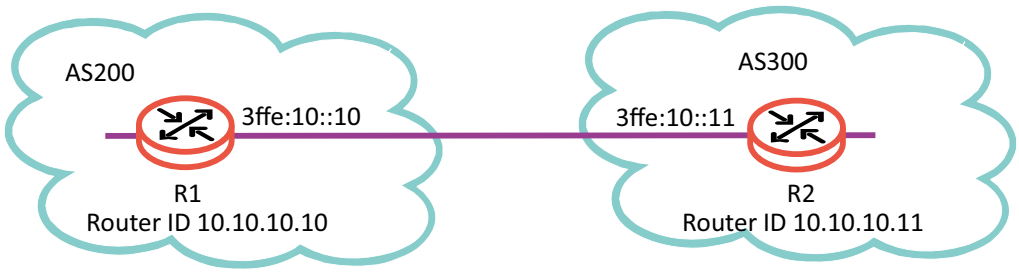
2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 200`	Define the routing process. The number 200 specifies the AS number of R2.
`iBOS(config-router)# neighbour 10.10.10.10 activate` `iBOS(config-router)# neighbour 10.10.10.10 remote-as 200`	Define BGP neighbours, and establish a TCP session. 10.10.10.10 is the IP address of the neighbour (R1), and 200 is the neighbour's AS number.

Enable BGP Between Different Autonomous Systems

This example shows the minimum configuration required for enabling BGP on an interface, when the routers belong to different autonomous systems. R1 and R2 are two routers in different autonomous system, AS200 and AS300, connecting to network 10.10.10.0/24.

Figure 55 Routers in Different Autonomous Systems



5. To configure router R1, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router)# neighbour 10.10.10.11 activate` `iBOS(config-router)# neighbour 10.10.10.11 remote-as 300`	Define BGP neighbours, and establish a TCP session. 10.10.10.11 is the IP address of the neighbour (R2), and 300 is the neighbour's AS number.

6. To configure router R2, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 300`	Define the routing process. The number `300` specifies the AS number of R2.
`iBOS(config-router)#neighbour 10.10.10.10 activate` `iBOS(config-router)#neighbour 10.10.10.10 remote-as 200`	Define BGP neighbours, and establish a TCP session. `10.10.10.10` is the IP address of the neighbour (R1), and `200` is the neighbour's AS number.

Configure AS Path Filtering

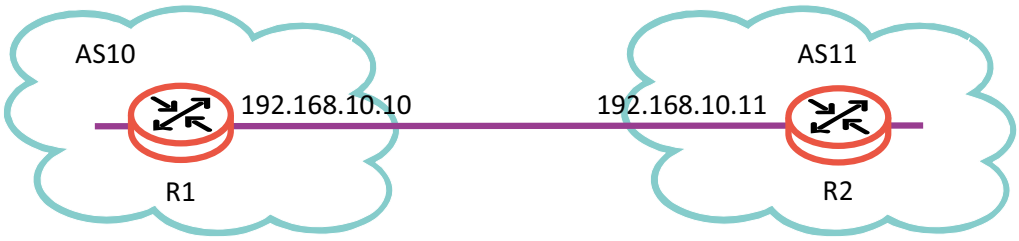
To configure BGP path filtering, use the following commands:

```
ibos(config)# ip as-path access-list access-list-number {permit |
deny} as-regexp
ibos(config)# router bgp as-number
ibos(config)# neighbour {ip-address | peer-group-name} filter-list
access-list-number {in | out}
```

Configure Route-Map

Use route maps to filter incoming updates from a BGP peer. In this example, a prefix-list `ws` on router R1 is configured to deny entry of any routes with the IP address 1.1.1.0/M (M = 26, 27, 28). To test the filter, router R2 is configured to generate network addresses `1.1.1.0/27` and `1.1.2.0/24`. To verify, use the **show ip bgp** command on R1; it displays R1 receiving updates from only `1.1.2.0/24`.

Figure 56 Route Map



To configure the route-map feature, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# ip prefix-list ws seq 5 deny 1.1.1.0/24`	Create an entry in the prefix-list. The ws parameter is the name of the map that is created above. 5 specifies the sequence number or position of this specific route map. deny specifies that the packets are to be rejected.
`iBOS(config)# ip prefix-list ws seq 10 permit any`	Create another entry in the ws map. 10 specifies the sequence number or position of this specific route map. The permit parameter any specifies accept all packets of any length.
`iBOS(config)# route-map ws 1 permit`	Enter Route-map mode to set the match operation.
`iBOS(config-route-map)# match ip address prefix-list ws`	Set the match criteria. In this case, if the route-map name matches ws, the packets from the first sequence are denied.
`iBOS(config-route-map)# exit`	Exit Route-map mode, and return to Configure mode.
`iBOS(config)# router bgp 10`	Define the routing process, and establish a TCP session. The number 10 specifies the AS number of R1.
`iBOS(config-router)# neighbour 192.168.10.11 remote-as 11`	Define BGP neighbours, and establish a TCP session. 192.168.10.11 is the IP address of the neighbour (R2), and 11 is the neighbour's AS number.
`iBOS(config-router)# neighbour 192.168.10.11 route-map ws in`	Apply a route map to routes. 192.168.10.11 specifies the IP address of BGP neighbour. The ws parameter is the name of the route map, and in specifies that the access list applies to incoming advertisements.

2. Configure router R2:

`iBOS(config)# router bgp 11`	Define the routing process. The number `11` specifies the AS number of R2.
`iBOS(config-router)# neighbour 192.168.10.10 remote-as 10`	Define BGP neighbours, and establish a TCP session. `192.168.10.10` is the IP address of the neighbour (R1), and `10` is the neighbour's AS number.
`iBOS(config-router)# network 1.1.1.0/27`	Specify the network to be advertised by the BGP routing process.
`iBOS(config-router)# network 1.1.2.0/24`	Specify the network to be advertised by the BGP routing process.

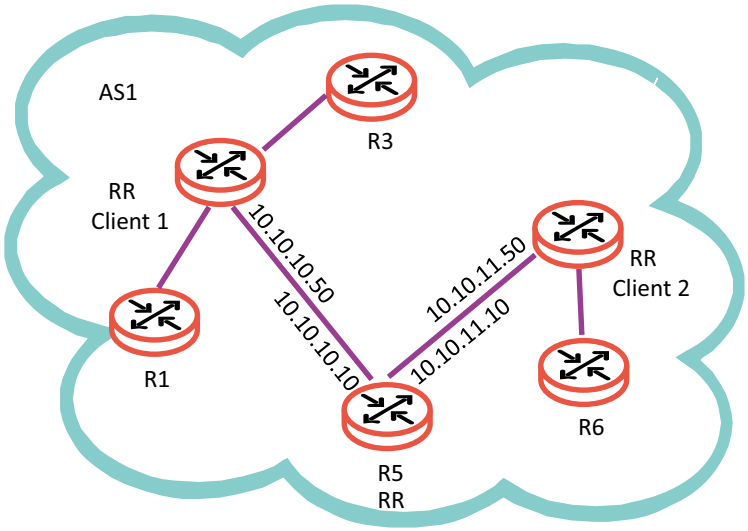
Redistribution of Routing into BGP

Generally, routing information in your network should be either in BGP or in the IGP. Redistribution between the two can cause problems, particularly if a circle redistribution is established where BGP routes are redistributed into the IGP and then redistributed back into BGP. Redistribution of routes from BGP also removes all BGP attributes such as AS-path and communities.

Configuring Route Reflector

A set of route reflector clients can be part of a cluster, commonly identified by the router-id of the route reflector. If there are multiple route reflectors (for redundancy) then a common cluster-id must be configured so that each route reflector can identify updates from another route reflector in the cluster. All route reflectors serving a cluster must be fully meshed and they should have an identical set of clients and nonclient peers.

Figure 57 Route Reflectors



In this example, R2, R5, and R4 would have to maintain a full mesh among themselves, but by making R5 the Route Reflector, R2 (RR Client1) has an iBGP session with the RR only, but not with R4 (RR Client 2). The routes learned from R2 are advertised to the other clients, and to iBGP peers outside the cluster; the iBGP routes learned from iBGP peers outside the cluster are advertised to R2. This reduces the iBGP peer connections in AS1.

To configure Route Reflector, do the following:

1. Configure router R5:

`R5# configure terminal`	Enter Configure mode.
`R5(config)# router bgp 1`	Define the routing process. The number 1 identifies the AS number of R5.
`R5(config-router)# neighbour 10.10.10.50 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.10.10.50 is the IP address of one of the neighbours (R2), and 1 is the neighbour's AS number.

`R5# configure terminal`	Enter Configure mode.
`R5(config-router)# neighbour 10.10.10.50 route-reflector client`	Configure R5 as the Route-Reflector (RR) and neighbour R2 as its client.
`R5(config-router)# neighbour 10.10.11.50 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.10.11.50 is the IP address of one of the neighbours (R4), and 1 is the neighbour's AS number.
`R5(config-router)# neighbour 10.10.11.50 route-reflector client`	Configure R5 as the Route-Reflector (RR) and neighbour R4 as its client.

2. Configure router R2:

`R2(config)# router bgp 1`	Define the routing process. The number 1 specifies the AS number of R2.
`R2(config-router)# neighbour 10.10.10.10 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.10.10.10 is the IP address of the neighbour (R5), and 1 is the neighbour's AS number.

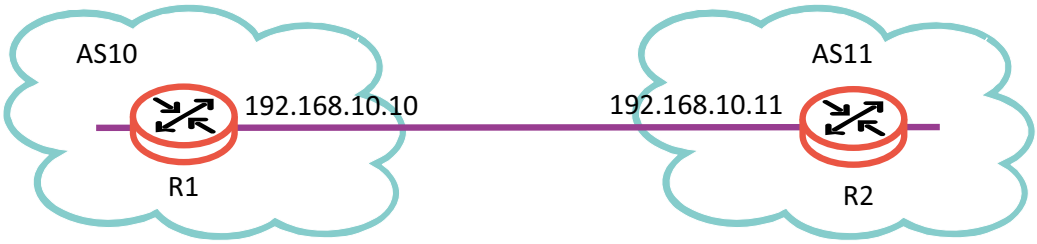
3. Configure router R4:

`R4(config)# router bgp 1`	Define the routing process. The number 1 identifies the AS number of R4.
`R4(config-router)# neighbour 10.10.11.10 remote-as 1`	Define BGP neighbour, and establish a TCP session. 10.10.11.10 is the IP address of the neighbour (R5), and 1 is the neighbour's AS number.

Configure Route-Maps

Route maps are used to filter incoming updates from a BGP peer. In this example, a prefix-list `ws` on router R1 is configured to deny entry of any routes with the IP address 1.1.1.0/M (M = 26, 27, 28).

Figure 58 Configure Route-Maps



To test the filter, router R2 is configured to generate network addresses 1.1.1.0/27 and 1.1.2.0/24.

To configure router R1, do the following:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# ip prefix-list ws seq 5 deny 1.1.1.0/27`	Create an entry in the prefix-list. The ws parameter is the name of the map that is created above. 5 specifies the sequence number or position of this specific route map. deny specifies the packets are to be rejected.
`iBOS(config)#ip prefix-list ws seq 10 permit any`	Create another entry in the ws map. 10 specifies the sequence number or position of this specific route map. The permit parameter any specifies accept all packets of any length.
`iBOS(config)# route-map ws 1 permit`	Enter Route-map mode to set the match operation.
`iBOS(config-route-map)# match ip address prefix-list ws`	Set the match criteria. In this case, if the route-map name matches ws, the packets from the first sequence are denied.
`iBOS(config-route-map)# exit`	Exit Route-map mode, and return to Configure mode.
`iBOS(config)# router bgp 10`	Define the routing process, and establish a TCP session. The number 10 specifies the AS number of R1.
`iBOS(config-router)# neighbour 192.168.10.11 activate` `iBOS(config-router)# neighbour 192.168.10.11 remote-as 11`	Define BGP neighbours, and establish a TCP session. 192.168.10.11 is the IP address of the neighbour (R2), and 11 is the neighbour's AS number.
`iBOS(config-router)# neighbour 192.168.10.11 route-map ws in`	Apply a route map to routes. 192.168.10.11 specifies the IP address of BGP neighbour. The ws parameter is the name of the route map, and in specifies that the access list applies to incoming advertisements.

To configure router R2, do the following:

`iBOS(config)#router bgp 11`	Define the routing process. The number 11 specifies the AS number of R2.
`iBOS(config-router)# neighbour 192.168.10.10 activate` `iBOS(config-router)# neighbour 192.168.10.10 remote-as 10`	Define BGP neighbours, and establish a TCP session. 192.168.10.10 is the IP address of the neighbour (R1), and 10 is the neighbour's AS number.
`iBOS(config-router)# network 1.1.1.0/27`	Specify the network to be advertised by the BGP routing process.
`iBOS(config-router)# network 1.1.2.0/24`	Specify the network to be advertised by the BGP routing process.

To verify the filter, the command **show ip bgp** is used on router R1 to show that router R1 is receiving updates from only 1.1.2.0/24.

```
iBOS# show ip bgp
BGP table version is 3, local router ID is 10.77.77.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal, l - labeled
             S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

ipv4 unicast:
  Network          Next Hop            Metric    LocPrf    Weight Path
*> 1.1.2.0/24      10.77.15.2               0                   0 11 i
Total number of prefixes 1
```

Configure Route Reflectors

The following examples of BGP route reflectors are described:

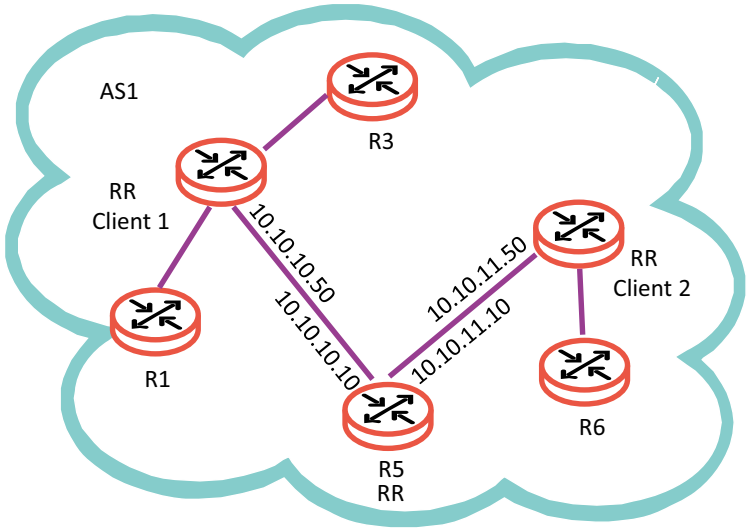
- "Configure Route Reflectors to Reduce iBGP Mesh Inside AS" on page 263
- "Configure Route Reflector to be Transparent to Clients" on page 264

Configure Route Reflectors to Reduce iBGP Mesh Inside AS

Route Reflectors are used to reduce the iBGP mesh inside an AS.

In this example, R2, R5, and R4 would have to maintain a full mesh among themselves, but by making R5 the Route Reflector, R2 (Client1) has an iBGP session with the RR only, but not with R4 (Client 2). The routes learned from R2 are advertised to the other clients, and to iBGP peers outside the cluster; the iBGP routes learned from iBGP peers outside the cluster are advertised to R2. This reduces the iBGP peer connections in AS1.

Figure 59 iBGP Route Reflectors



To configure the BGP Route Reflector R5, do the following:

`R5# configure terminal`	Enter Configure mode.
`R5(config)# router bgp 1`	Define the routing process. The number 1 identifies the AS number of R5.
`R5(config-router)# neighbour 10.10.10.50 activate` `R5(config-router)# neighbour 10.10.10.50 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.10.10.50 is the IP address of one of the neighbours (R2), and 1 is the neighbour's AS number.
`R5(config-router)# neighbour 10.10.10.50 route-reflector client`	Configure R5 as the Route-Reflector (RR) and neighbour R2 as its client.

`R5# configure terminal`	Enter Configure mode.
`R5(config-router)# neighbour 10.10.11.50 activate` `R5(config-router)# neighbour 10.10.11.50 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.10.11.50 is the IP address of one of the neighbours (R4), and 1 is the neighbour's AS number.
`R5(config-router)# neighbour 10.10.11.50 route-reflector client`	Configure R5 as the Route-Reflector (RR) and neighbour R4 as its client.

To configure the RR Client 1 (R2) do the following:

`R2(config)#router bgp 1`	Define the routing process. The number 1 specifies the AS number of R2.
`R2(config-router)# neighbour 10.10.10.10 remote-as 1` `R2(config-router)# neighbour 10.10.10.10 activate`	Define the BGP neighbour, and establish a TCP session. 10.10.10.10 is the IP address of the neighbour (R5), and 1 is the neighbour's AS number.

To configure the RR Client 2 (R4), do the following:

`R4(config)#router bgp 1`	Define the routing process. The number 1 identifies the AS number of R4.
`R4(config-router)#neighbour 10.10.11.10 activate` `R4(config-router)#neighbour 10.10.11.10 remote-as 1`	Define BGP neighbour, and establish a TCP session. 10.10.11.10 is the IP address of the neighbour (R5), and 1 is the neighbour's AS number.

For more details, refer descriptions for **neighbour remote-as** and **neighbour route-reflector-client** in the *iBOS Command Reference*.

To verify the configuration, use the following commands:

- **show ip bgp**
- **show ip bgp neighbours**

Configure Route Reflector to be Transparent to Clients

The basic rule of BGP is that a BGP speaker cannot advertise a route to an iBGP neighbour if that route was learned from another iBGP neighbour. Configuring a route reflector provides a means to circumvent this rule. The entire route reflector process is transparent to the clients, and no configuration is necessary on these clients.

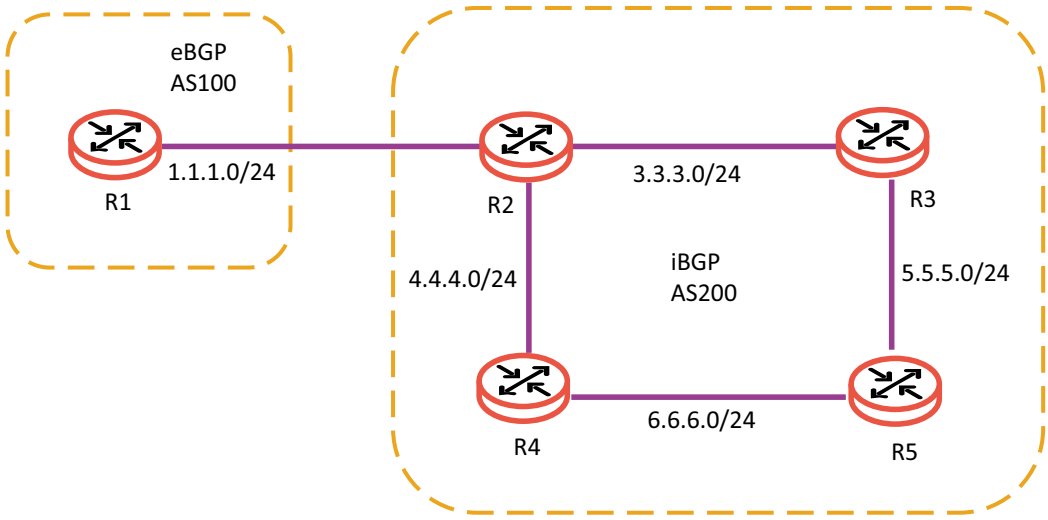
Whenever an iBGP-speaking router receives a route update, it forwards the route to the neighbour without changing the nexthop IP address, thus making it an unreachable route, unless verified by an iBGP (for example, neighbour x.x.x.x route-reflector-client).

If a router is configured as a Route Reflector, it forwards the routes received by changing the nexthop address as itself, thus making the nexthop reachable.

If a route is received from a client, the route is forwarded to the clients.

If a route is received from a non-client, the route is forwarded to the clients and non-clients.

Figure 60 eBGP and iBGP Route Reflector Topology



To configure router R1, do the following:

`R1#configure terminal`	Enter Configure mode.
`R1(config)#router bgp 100`	Define the routing process with AS number 100.
`R1(config-router)#neighbour 1.1.1.2 remote-as 200`	Define the eBGP neighbour (R2).
`R1(config-router)#network 100.100.100.100/32`	Advertise a route via eBGP connection to R2.

To configure the router R2, as router reflector (RR), do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Define the eBGP neighbour (R1).
`iBOS(config-router)#neighbour 4.4.4.1 remote-as 200`	Define the iBGP neighbour (R4).
`iBOS(config-router)#neighbour 3.3.3.1 remote-as 200`	Define the iBGP neighbour (R3).
`iBOS(config-router)#bgp cluster-id 4`	Define a cluster ID (4) when multiple Route Reflectors exist.
`iBOS(config-router)#neighbour 3.3.3.1 route-reflector client`	Configure neighbour router R3 as a RR client.
`iBOS(config-router)#neighbour 4.4.4.1 route-reflector client`	Configure neighbour router R4 as a RR client.

To configure router R3 as a RR-client, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 3.3.3.2 remote-as 200`	Define the iBGP neighbour (R2).
`iBOS(config-router)#neighbour 5.5.5.2 remote-as 200`	Define the iBGP neighbour (R5).

To configure router R4 as a RR-client, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 4.4.4.2 remote-as 200`	Define the iBGP neighbour (R2).
`iBOS(config-router)#neighbour 6.6.6.2 remote-as 200`	Define the iBGP neighbour (R5).

To configure router R5, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 5.5.5.1 remote-as 200`	Define the iBGP neighbour (R3).
`iBOS(config-router)#neighbour 6.6.6.1 remote-as 200`	Define the iBGP neighbour (R4).

Refer *iBOS Command Reference* for more details on the following commands:

- **neighbour remote-as**
- **network**
- **neighbour route-reflector-client**

To verify the configuration, use the commands:

- **show ip bgp**
- **show ip bgp summary**
- **show ip route**

Configure Route Server

At an Internet exchange point, many ISP's are connected to each other by external BGP peering. Normally, these external BGP connections are done by full mesh. As with internal BGP full-mesh formation, this method has a scaling issue.

The Route Server is a method to resolve the problem; it is very similar to Route Reflector functionality, except it uses eBGP connections, in practice. Each ISP's BGP router only peers to the Route Server. Route Server serves as BGP information exchange to other BGP routers. By applying this method, the number of BGP

connections is reduced to O(n).

Route Server also takes care of inbound and outbound filtering policies for many-to-many eBGP connections, as they would have appeared if each router were connected to no other routers.

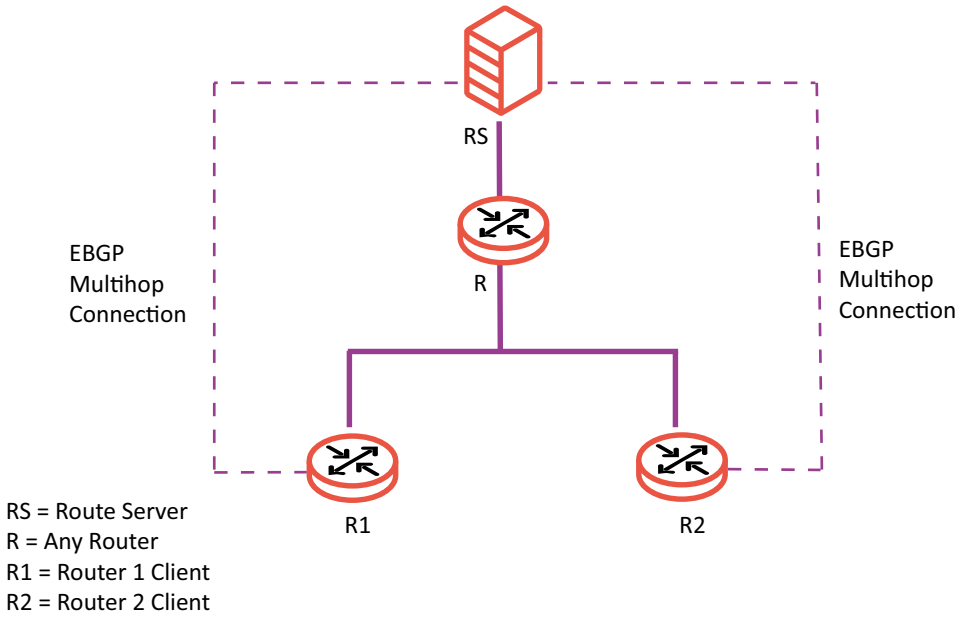
A BGP daemon can work as a normal BGP router, a Route Server, or as both.

In iBOS, the Route Server can be achieved using BGP views. iBOS does not guarantee support Route Server over a BGP-MPLS-VPN link. However, the Route Server can support IPv4 and IPv6 prefix services to different client routers through single-hop or multi-hop eBGP.

The Route Server offers transparency of next-hop, MED, and AS-number while serving the routes from one client to another. It also supports a limited route-filtering capability for inbound and outbound control messages.

When a BGP daemon is configured in Route-Server mode for a particular BGP view, it does not install any routes in the kernel for that view.

Figure 61 Route Server Topology



To configure the router RS, do the following:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router ospf`	Enter router mode to configure an OSPF routing process
`iBOS(config-router)# network 50.1.1.0/24 area 0`	Define the interface (50.1.1.0/24) on which OSPF runs, and associate the area ID (0) with the interface (area ID 0 specifies the backbone area).
`iBOS(config-router)# network 144.144.144.144/32 area 0`	Define the interface (144.144.144.144/32) on which OSPF runs, and associate the area ID (0) with the interface (area ID 0 specifies the backbone area).
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# router bgp 5 view a`	Specify a BGP routing view name.

`iBOS(config-router)# neighbour 10.60.60.60 remote-as 2`	Define the BGP neighbour, and establish a TCP session. 10.60.60.60 is the IP address of the neighbour, and 2 is the neighbour's AS number.
`iBOS(config-router)# neighbour 10.60.60.60 ebgp-multihop 255`	Define the neighbour 10.60.60.60 can be 255 hops away.
`iBOS(config-router)# neighbour 10.60.60.60 update-source lo`	Specify the loopback interface BGP sessions use for the TCP connection.
`iBOS(config-router)# neighbour 60.60.60.60 route-server-client`	Configure the router as a BGP Route Server, and specify the 60.60.60.60 neighbour as its client.
`iBOS(config-router)# neighbour 10.77.77.77 remote-as 1`	Define the BGP neighbour, and establish a TCP session. 10.77.77.77 is the IP address of the neighbour, and 1 is the neighbour's AS number.
`iBOS(config-router)# neighbour 10.77.77.77 ebgp-multihop 255`	Define the neighbour 10.77.77.77 can be 255 hops away.
`iBOS(config-router)# neighbour 10.77.77.77 update-source lo`	Specify the loopback interface BGP sessions use for the TCP connection.
`iBOS(config-router)# neighbour 10.77.77.77 route-server-client`	Configure the router as a BGP Route Server, and specify the 10.77.77.77 neighbour as its client.
`iBOS(config-router)# neighbour 10.77.77.77 route-map to_R1 out`	Apply a route map to the to_R1 outgoing route.
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# access-list lan02 permit 100.100.100.0/24`	Specify permit route 100.100.100.0/24 for access list lan02.
`iBOS(config)# access-list r2_NH permit 60.60.60.60/32`	Specify permit route 10.60.60.60/32 for access list r2_NH.
`iBOS(config)# route-map to_R1 permit 1`	Enter Route-Map mode, and permit route to_R1, sequence 1.
`iBOS(config-route-map)# match ip address lan02`	Specify match address lan02.
`iBOS(config-route-map)# set ip next-hop 10.60.60.60`	Specify 10.60.60.60 as the address of the nexthop.

Refer *iBOS Command Reference* for more details on the following commands:

- **network area**
- **router bgp view**
- **neighbour remote-as**
- **neighbour update-source**
- **neighbour route-server-client**
- **neighbour route-map**

To verify the configuration, use the command **show ip bgp view**.

When BGP is configured using the **bgp multiple-instance** command, the router will able to run multiple BGP instances. The following caveats apply:

- all BGP instances must be deleted before enabling or disabling this feature
- a specific peer cannot be a member of multiple instances
- a normal BGP instance and multiple views can be run

When BGP is configured using the **bgp multiple-instance allow-same peer** command, router can run multiple BGP instances in views only. The following caveats apply

- all BGP instances must be deleted
- a specific peer can be a member of multiple instances
- a normal BGP instance cannot be run, but multiple BGP views can be run

When BGP is configured using the **neighbour x.x.x.x route-server client** command, the x.x.x.x neighbour is treated as a Route-Server client.

These configuration commands are used to allow a BGP router to transfer all capabilities of BGP routes to all clients and to distribute routes with advanced dynamic policies.

Configure a Route Server with Multiple Views

In the following example, R1 is a member of View 1, R2 runs two views, R3 is a member of View 2, and R4 is a member of both View 1 and 2.

Figure 62 Router-Server with Multiple Views

To configure the router server with multiple views, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 100`	Define the routing process with AS number 100.
`iBOS(config-router)# neighbour 1.1.1.2 remote-as 200`	Define an eBGP neighbour (R2).

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# bgp multiple-instance allow-same-peer`	Create only BGP views in which the same peers can be members of multiple views. A normal BGP instance is not created.
`iBOS(config)# router bgp 200 view one`	Create a BGP view.
`iBOS(config-router)# neighbour 1.1.1.1 remote-as 100`	Define an eBGP neighbour.
`iBOS(config-router)# neighbour 1.1.1.1 route-server-client`	Configure a Route-Server client on R2.
`iBOS(config-router)# neighbour 4.4.4.4 remote-as 400`	Define an eBGP neighbour.
`iBOS(config-router)# neighbour 4.4.4.4 route-server-client`	Configure a Route-Server client on R2.
`iBOS(config-router)# exit`	Exit Router mode and enter Configure mode.
`iBOS(config)# router bgp 200 view two`	Create a BGP view.
`iBOS(config-router)# neighbour 3.3.3.1 remote-as 300`	Define an eBGP neighbour.
`iBOS(config-router)# neighbour 3.3.3.1 route-server-client`	Configure a Route-Server client on R2.
`iBOS(config-router)# neighbour 4.4.4.1 remote-as 400`	Define an eBGP neighbour.
`iBOS(config-router)# neighbour 4.4.4.1 route-server-client`	Configure a Route-Server client on R2.

3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 300`	Define the routing process with AS number 300.
`iBOS(config-router)# neighbour 3.3.3.2 remote-as 200`	Define an eBGP neighbour (R2).

4. Configure router R4:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 400`	Define the routing process with AS number 400.
`iBOS(config-router)# neighbour 4.4.4.2 remote-as 200`	Define an eBGP neighbour (R2).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **neighbour remote-as**
- **bgp multiple-instance allow-same-peer**
- **router bgp view**
- **neighbour route-server-client**

To verify the configuration, use the following commands:

- **show ip bgp**
- **show ip bgp summary**
- **show ip bgp view one**
- **show ip bgp view one summary**

Configure BGP Confederations

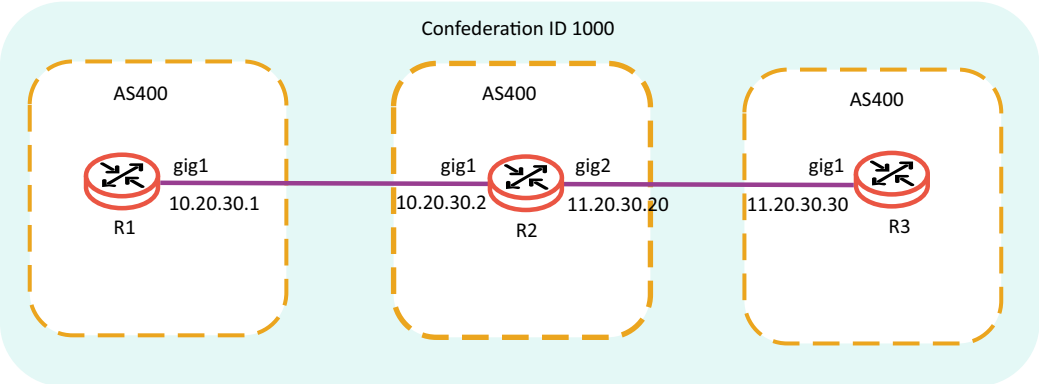
In BGP, nodes running iBGP protocols must be interconnected forming a full mesh amongst each other. Confederation solves the iBGP full-mesh network complexity and inefficiency by splitting a large autonomous system domain into smaller autonomous system domains, called member autonomous systems. Member autonomous systems can form eBGP connections among themselves, to prevent full-mesh connections among each iBGP-running node.

The **bgp confederation identifier** command tells the router that it is a member of a confederation and the confederation ID. The **bgp confederation peers** command lists the member AS to which the router is connected.

Configure BGP Confederation with Same Member AS

In the following example, R1, R2, and R3 are members of the same confederation with different AS numbers.

Figure 63 BGP Confederation



To configure the BGP confederation, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value (400) to the router.
`iBOS(config-router)# bgp confederation identifier 1000`	Specify the BGP confederation ID, the externally visible autonomous system number that identifies the BGP confederation as a whole.
`iBOS(config-router)# bgp confederation peers 400`	Specify the neighbour ASN value for confederation membership.
`iBOS(config-router)# neighbour 10.20.30.2 remote-as 400`	Specify the neighbour's IP address (10.20.30.2) and the ASN value of the neighbour (400).

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value (400) to the router.
`iBOS(config-router)# bgp confederation identifier 1000`	Specify the BGP confederation ID.
`iBOS(config-router)# bgp confederation peers 400`	Specify the neighbour ASN values for confederation membership.
`iBOS(config-router)# neighbour 10.20.30.1 remote-as 400`	Specify the neighbour's IP address (10.20.30.1) and the ASN value of the neighbour (400).
`iBOS(config-router)# neighbour 11.20.30.30 remote-as 400`	Specify the neighbour's IP address (11.20.30.30) and the ASN value of the neighbour (400).

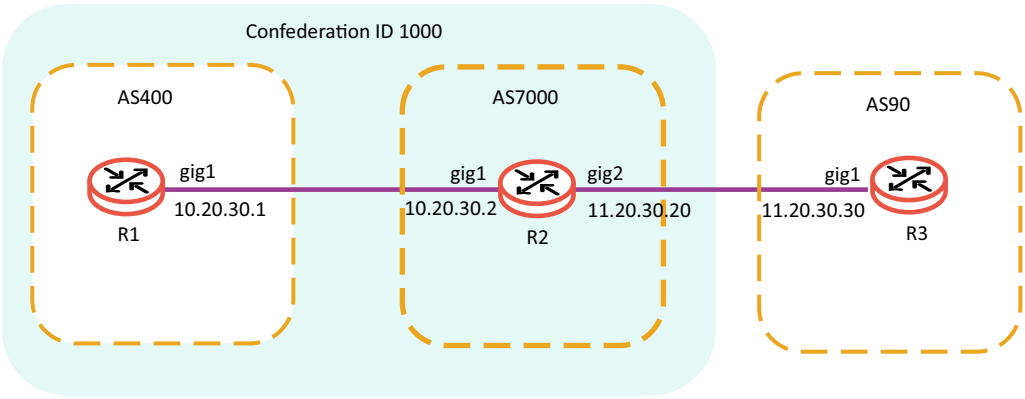
3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value (400) to the router.
`iBOS(config-router)# bgp confederation identifier 1000`	Specify the BGP confederation ID.
`iBOS(config-router)# bgp confederation peers 400`	Specify the neighbour ASN value for confederation membership.
`iBOS(config-router)# neighbour 11.20.30.20 remote-as 400`	Specify the neighbour's IP address (11.20.30.20) and the ASN value of the neighbour (400).

Configure Confederation between Members and Non-Members

In the following example, R1 and R2 are members a confederation and have the same AS number, and router R3 is a member outside the confederation with a different AS number.

Figure 64 BGP Confederation with Multiple AS



To configure BGP confederation between members and non-members, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value to the router.
`iBOS(config-router)# bgp confederation identifier 1000`	Specify the BGP confederation ID.
`iBOS(config-router)# neighbour 10.20.30.2 remote-as 7000`	Specify the neighbour's IP address (10.20.30.2) and the ASN value of the neighbour (7000).

2. To configure router R2, do the following:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2) to configure, and enter Interface mode.
`iBOS(config-if)# ip address 11.20.30.20/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router bgp 7000`	Assign the ASN value to the router.
`iBOS(config-router)# bgp confederation identifier 1000`	Specify the BGP confederation ID.
`iBOS(config-router)# bgp confederation peers 90`	Specify the neighbour ASN value for confederation membership.
`iBOS(config-router)# neighbour 10.20.30.1 remote-as 400`	Specify the neighbour's IP address (10.20.30.1) and the ASN value of the neighbour (400).
`iBOS(config-router)# neighbour 11.20.30.30 remote-as 90`	Specify the neighbour's IP address (11.20.30.30) and the ASN value of the neighbour (90).

3. To configure router R3, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 90`	Assign the ASN value (90) to the router.
`iBOS(config-router)# neighbour 11.20.30.20 remote-as 1000`	Specify the neighbour's IP address (11.20.30.20) and the Confederation ID for the neighbour (1000).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **bgp confederation identifier**
- **bgp confederation peers**
- **neighbour remote-as**

To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**

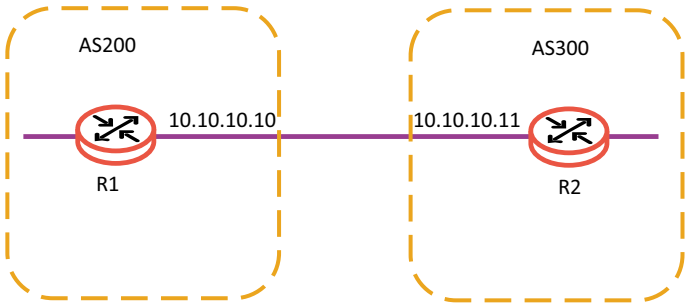
BGP Authentication

BGP authentication allows users to receive selected routing information, enhancing security of their network traffic. When BGP authentication is enabled on a router, the router verifies routing packets it receives by exchanging a password that is configured on both the sending and receiving routers.

To enable BGP authentication on TCP/IP, apply a kernel patch and specific MD5 libraries. Refer to the Installation Guide for detailed information on how to apply the MD5 authentication patch, and the required libraries.

In this example, both R1 and R2 have daveva as the password. Configure the same password on all routers that will communicate using BGP in the network.

Figure 65 BGP Authentication



To configure router R1, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config-router)#neighbour 10.10.10.11 remote-as 200`	Define BGP neighbours, and establish a TCP session. 10.10.10.11 is the IP address of the neighbour (R2), and 200 is the neighbour's AS number.
`iBOS(config-router)#neighbour 10.10.10.11 password daveva`	Specify the password.

To configure router R2, do the following:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R2.
`iBOS(config-router)#neighbour 10.10.10.10 remote-as 200`	Define BGP neighbours, and establish a TCP session. 10.10.10.10 is the IP address of the neighbour (R1), and 200 is the neighbour's AS number.
`iBOS(config-router)#neighbour 10.10.10.10 password daveva`	Specify the password.

Refer *iBOS Command Reference* for detailed description of the following commands:

- **neighbour remote-as**
- **neighbour password**

To verify the configuration, use the following commands:

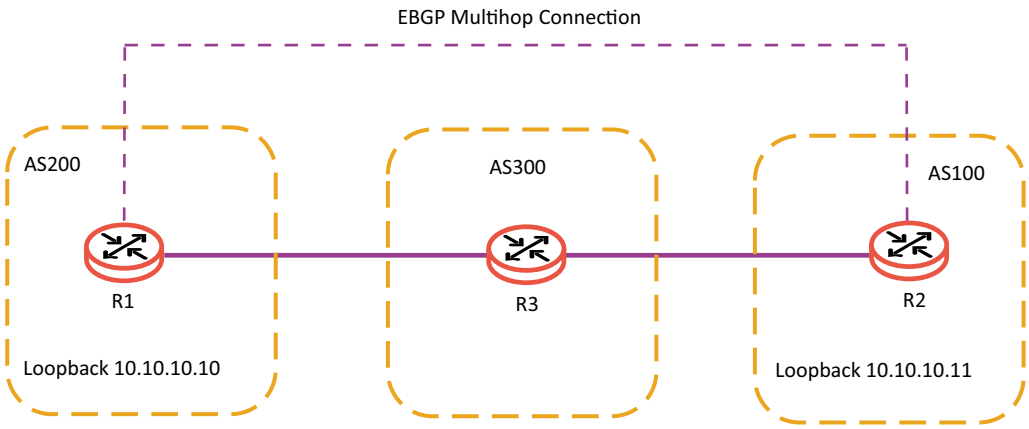
- **show ip bgp**
- **show ip bgp neighbours**

Enable eBGP Multihop

This example shows the minimum configuration required for enabling eBGP multihop on peers speaking BGP. eBGP multihop is used for routers that are not directly connected to each other. Typically, eBGP peers are directly connected, but if there is a requirement that necessitates this scenario, this configuration can be used.

The IP addresses used in the configuration should be accessible through an IGP or static routing.

Figure 66 eBGP Multihop Connection



To enable eBGP multihop, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router)#neighbour 10.10.10.11 remote-as 100`	Define BGP neighbours, and establish a TCP session. 10.10.10.11 is the IP address of the neighbour (R2), and 100 is the neighbour's AS number.
`iBOS(config-router)#neighbour 10.10.10.11 ebgp-multihop 10`	Define the neighbour 10.10.10.11 can be 10 hops away.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Define the routing process. The number 100 specifies the AS number of R2.
`iBOS(config-router)#neighbour 10.10.10.11 remote-as 200`	Define BGP neighbours, and establish a TCP session. 10.10.10.10 is the IP address of the neighbour (R1), and 200 is the neighbour's AS number.
`iBOS(config-router)#neighbour 10.10.10.10 ebgp-multihop 10`	Define that the neighbour 10.10.10.10 can be up to 10 hops away.

Refer *iBOS Command Reference* for detailed description of the following commands:

- **router bgp**
- **neighbour remote-as**
- **neighbour ebgp-multihop**

To verify the configuration, use the following commands:

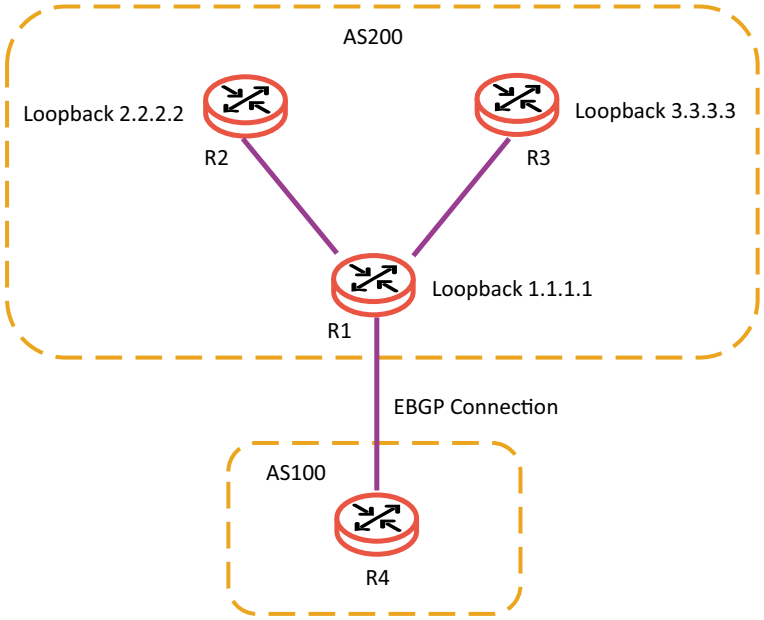
- **show ip bgp**
- **show ip bgp neighbours**

Enable Peer Groups

A BGP speaker might have the same update policies for a set of its peers. This is very useful if you have to change the update policies for all of the peers: Changing individual routers for separate policies can be very time-consuming, thus, peer groups play an important role in creating and assigning policies to a group of routers.

In the following scenario, R1, R2, and R3 belong to the same peer group. R4 and R1 are eBGP peers. R1 is the route reflector (configuration not shown), and R2 and R3 are in AS 200. R4 is in AS 100.

Figure 67 BGP Peer Groups



To enable BGP peer groups, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router-bgp)# neighbour ws peer-group`	Create a peer group named ws.
`iBOS(config-router-bgp)# neighbour ws remote-as 200`	Assign options to the peer group named ws. The option field can be one of the commands allowed at this field. Refer to the iBOS Command Reference for descriptions of options.
`iBOS(config-router-bgp)# neighbour ws activate`	Activate the peer-group for the address family ws
`iBOS(config-router-bgp)# neighbour 2.2.2.2 peer-group ws`	Define neighbour 2.2.2.2 (R2) as a peer group member.
`iBOS(config-router)#neighbour 3.3.3.3 peer-group ws`	Define neighbour 3.3.3.3 (R3) as a peer group member.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 200`	Create a TCP connection with neighbour 1.1.1.1 of AS 200.

3. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process. The number 200 specifies the AS number of R1.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 200`	Create a TCP connection with neighbour 1.1.1.1 of AS 200.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **neighbour remote-as**
- **neighbour peer-group**

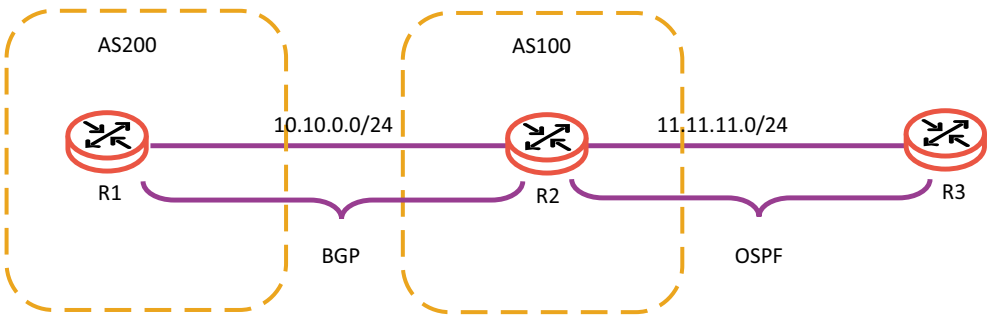
To verify the configuration, use the following commands:

- **show ip bgp**
- **show ip bgp neighbours**

Redistribute with OSPF

If there are routers that run both OSPF and BGP, certain OSPF routes might have to be sent to other eBGP peers. This can be achieved using the redistribution feature. Consider the following topology, in which R1 and R2 are eBGP peers, and R2 and R3 are OSPF peers. R2 is redistributing OSPF routes into BGP. The OSPF routes are sent to the R1 BGP routing table. This configuration assumes that all OSPF and eBGP sessions are up and running, and that only the redistribution must be configured.

Figure 68 Redistribute with OSPF



To redistribute with OSPF, configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Define the routing process. The number 100 specifies the AS number of R2.
`iBOS(config-router)#redistribute ospf`	Redistribute OSPF routes in the R2 routing table into the R1 BGP routing table.
`iBOS(config-router)#exit`	Exit Router-BGP mode.

Refer *iBOS Command Reference* for detailed description of the following commands:

- **router bgp**
- **redistribute**

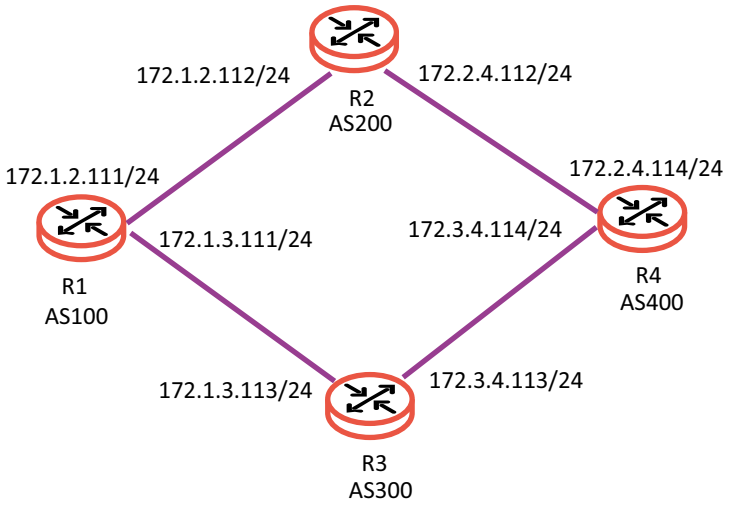
To verify the configuration, use the command **show ip bgp**.

Add Multiple Instances of Same AS to Administer Route Selection

BGP supports addition of the same AS number multiple times to influence the route selection process. This can be done using route maps, as described below.

Under normal circumstances, any route advertised by R1 is sent to R4 via two different routes, and then R4 selects the path from R2. This decision can be influenced by adding multiple instances of AS number 200 at R2.

Figure 69 Multiple Instances of Same AS



To add multiple instances of AS, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Define the routing process with AS number 100.
`iBOS(config-router)#neighbour 172.1.2.112 remote-as 200`	Define neighbour R2. 172.1.2.112 is the IP address of R2, and 200 is the AS number.

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config-router)#neighbour 172.1.3.113 remote-as 300`	Define neighbour R3. 172.1.3.113 is the IP address of R2, and 300 is the AS number.
`iBOS(config-router)#network 44.44.44.0/24`	Advertise network 44.44.44.0/24 through BGP. This route reaches R4 via R2 and R3.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#route-map mul_inst permit 10`	Define the route-map multiple instance with permit definition sequence number 10.
`iBOS(config)#set as-path prepend 200 200`	Prepend AS number 200 two times to the AS_PATH attribute in the BGP Update message.
`iBOS(config-route-map)#exit`	Exit Route-map mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 172.1.2.111 remote-as 100`	Define neighbour R1. 172.1.2.111 is the IP address of R1, and 100 is the AS number.
`iBOS(config-router)#neighbour 172.2.4.114 remote-as 400`	Define neighbour R4. 172.2.4.114 is the IP address of R2, and 400 is the AS number.
`iBOS(config-router)#neighbour 172.2.4.114 route-map mul_inst out`	Apply route-map multi_inst to all outbound routes to R4.

3. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 300`	Define the routing process with AS number 300.
`iBOS(config-router)#neighbour 172.1.3.111 remote-as 100`	Define neighbour R1. 172.1.3.111 is the IP address of R1, and 100 is the AS number.
`iBOS(config-router)#neighbour 172.3.4.114 remote-as 400`	Define neighbour R4. 172.3.4.114 is the IP address of R4, and 400 is the AS number.

4. Configure router R4:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 400`	Define the routing process with AS number 400.
`iBOS(config-router)#neighbour 172.2.4.112 remote-as 200`	Define neighbour R2. 172.2.4.112 is the IP address of R2, and 200 is the AS number.
`iBOS(config-router)#neighbour 172.3.4.113 remote-as 300`	Define neighbour R3. 172.3.4.113 is the IP address of R3, and 300 is the AS number.

Refer *iBOS Command Reference* for detailed description of the following commands:

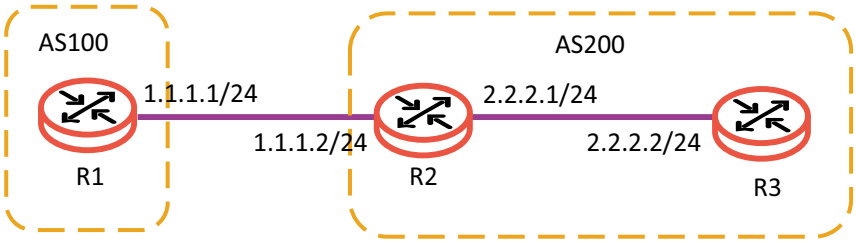
- **rset as-path prepend**
- **remote-as**
- **route-map**
- **rnetwork**

To verify the configuration, use the command **show ip bgp**.

Remove Multi-Exit Disc Attribute from Update Messages

Users can remove the Multi-Exit Disc (MED) attribute values from received update messages to be sent.

Figure 70 Remove MED Attribute



To configure routes with MED attribute values, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#route-map med permit 1`	Define the route-map MED with permit definition sequence number 1.
`iBOS(config-route-map)#exit`	Exit Route-map mode, and return to Configure mode.
`iBOS(config)#router bgp 100`	Define the routing process with AS number 100.
`iBOS(config-router)#neighbour 1.1.1.2 remote-as 200`	Define neighbour R2. 1.1.1.2 is the IP address of R2, and 200 is the AS number.
`iBOS(config-router)#neighbour 1.1.1.2 route-map med out`	Apply the route-map MED to all outbound routes to R2.

2. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 2.2.2.1 remote-as 200`	Define neighbour R2. 2.2.2.1 is the IP address of R2, and 200 is the AS number.

To remove the received and sent MED values respectively, do the following:

3. Configure router R2 and remove received MED Value:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Define neighbour R1. 1.1.1.1 is the IP address of R1, and 100 is the AS number.
`iBOS(config-router)#neighbour 2.2.2.2 remote-as 200`	Define neighbour R3. 2.2.2.2 is the IP address of R3, and 200 is the AS number.
`iBOS(config-router)#bgp bestpath med remove-recv-med`	Enable the remove received MED value option.

4. Configure router R2 and remove Send MED Value:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Define the routing process with AS number 200.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Define neighbour R1. 1.1.1.1 is the IP address of R1, and 100 is the AS number.
`iBOS(config-router)#neighbour 2.2.2.2 remote-as 200`	Define neighbour R3. 2.2.2.2 is the IP address of R3, and 200 is the AS number.
`iBOS(config-router)#bgp bestpath med remove-send-med`	Enable the remove sent MED value option.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **neighbour remote-as**
- **route-map med**
- **neighbour route-map med out**
- **bgp bestpath med remove-recv-med**
- **bgp bestpath med remove-send-med**

To verify the configuration, use the command **show ip bgp**.

Configure BGP Four-Byte AS

Extended AS numbers can be mapped to 2-byte AS numbers if the value is less than, or equal to, 65535. If the AS number is higher than 65535, it cannot be mapped to a 2-byte AS number. Therefore, if a BGP speaker is configured with a non-mappable AS number, it must enable the BGP extended ASN capability in iBOS.

Note: *Autonomous System number 23456 is a reserved IANA number for AS transition; thus, it is recommended that no system be configured with 23456 as its AS number.*

The extended ASN capability is disabled by default. However, when it is enabled, it is able to interoperate with a 2-byte AS-numbered speaker, in compliance with RFC 4893.

If a 4-byte AS number is configured in the provider's network using BGP MPLS VPN or standard IPv4/IPv6 BGP, it is recommended that the PE routers be 4-byte AS-enabled before connecting to 4-byte AS-enabled customer networks. For implications related to AS number transition issues, refer to RFC 4893.

For experimental purposes, iBOS 7 is are also capable of setting up 4-byte AS-specific extended communities and route distinguishers (RDs) with limited capabilities. However, it is recommended that 2-byte AS-specific RDs and extended communities be used for regular deployment.

BGP encodes an ASN into four octets, so that more autonomous systems can be supported. Extended ASN capability is advertised in the Open message capabilities when the 4-octet ASN capability is enabled. When the 4-octet ASN capability is enabled, the valid ASN value range is <1-4294967295>, with the exception discussed in the Note above.

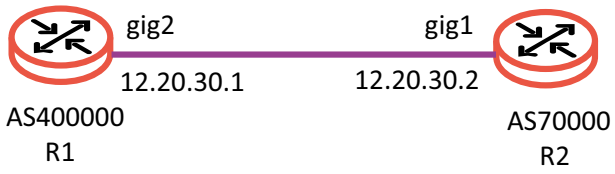
Note: *Four-octet capability is disabled by default.*

This feature requires that the standard iBOS configuration mode for BGP is enabled. If, for some reason, the mode has been changed, the configuration will not take effect. To ensure that this mode is being this, use the bgp -config-type standard

4-Octet ASN Capability Enabled on R1 and R2

In this example, 4-Octet ASN capability is enabled on BGP speakers R1 and R2.

Figure 71 4-Octet ASN on Both Routers



To configure router R1, do the following:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# bgp extended-asn-cap`	Enable 4-octet ASN capability.
`R1(config)# router bgp 400000`	Assign the ASN value (400000) to the router.
`R1(config-router)# neighbour 10.20.30.2 remote-as 70000`	Specify the neighbour's IP address (10.20.30.2) and the ASN value of the neighbour (70000).

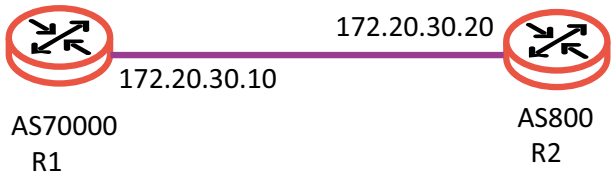
To configure router R2, do the following:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# bgp extended-asn-cap`	Enable 4-octet ASN capability.
`R2(config)# router bgp 70000`	Assign the ASN value (70000) to the router.
`R2(config-router)# neighbour 10.20.30.1 remote-as 400000`	Specify the neighbour's IP address (10.20.30.1) and the ASN value of the neighbour (400000).

4-Octet ASN Capability Enabled on R1 and Disabled on R2

In the following two examples, 4-Octet ASN capability is enabled on BGP speaker R1 and disabled on R2.

Figure 72 4-Octet ASN on One Router



To configure 4-Octet ASN on One Router, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#bgp extended-asn-cap`	Enable 4-octet ASN capability.
`iBOS(config)#router bgp 70000`	Assign the ASN value (70000) to the router.
`iBOS(config-router)#neighbour 172.20.30.20 remote-as 800`	Specify the neighbour's IP address (172.20.30.20) and the ASN value of the neighbour (800).

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#no bgp extended-asn-cap`	Disable 4-octet ASN capability.
`iBOS(config)#router bgp 800`	Assign the ASN value (800) to the router.
`iBOS(config-router)#neighbour 172.20.30.10 remote-as 23456`	Specify the neighbour's IP address (172.20.30.10) and the ASN value of the neighbour (23456).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **neighbour remote-as**
- **bgp extended asn-cap**

To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**

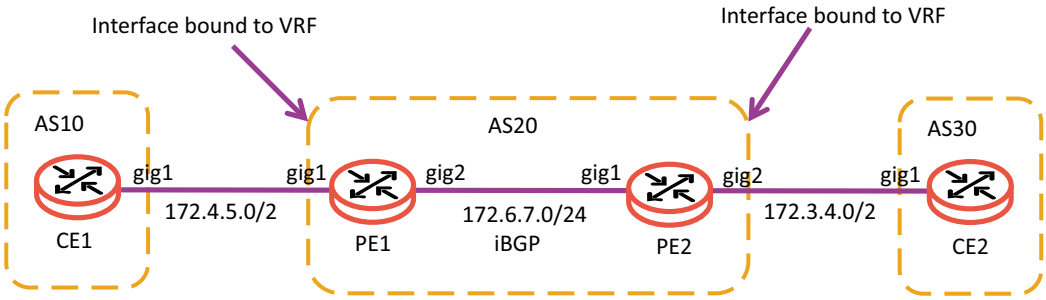
Configure BGP Extended Community Attribute

The Extended Community Attribute provides a mechanism for labelling information carried in BGP.

Configure Extended Community with 2-Byte ASN

In the following example, CE1, PE1, PE2, and CE2 are 2-byte-ASN capable, and do not support 4-byte-ASN capability.

Figure 73 Extended Communities — 2-Byte ASN



Configure router CE1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)# p address 172.4.5.115/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router bgp 100`	Assign the ASN value (100) to the router. The ASN range is <1-65535>.
`iBOS(config-router)# neighbour 172.4.5.116 remote-as 200`	Specify the neighbour's IP address (172.4.5.116) and the ASN value of the neighbour (200).

Configure router PE1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`iBOS(config-vrf)# rd 100:10`	Assign a route distinguisher (RD) for the VRF, which is a unique value on the router. The RD value can be in ASN:NN or A.B.C.D:NN format.
`iBOS(config-vrf)# route-target both 100:200`	Specify the 2-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`iBOS(config-vrf)# exit`	Exit VRF mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)# ip vrf forwarding VRF1`	Bind the interface (gigabitethernet1) to the VRF (VRF1).
`iBOS(config-if)# ip address 172.4.5.116/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`iBOS(config-if)# ip address 172.6.7.116/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)# neighbour 172.6.7.117 remote-as 200`	Specify the neighbour's (PE2) IP address (172.6.7.117) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.
`iBOS(config-router)# address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`iBOS(config-router-af)# neighbour 172.6.7.117 activate`	Enable the exchange of routing information with a peer router.
`iBOS(config-router-af)# exit`	Exit Address-Family-VPNv4 mode.
`iBOS(config-router)# address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`iBOS(config-router-af)# neighbour 172.4.5.115 remote-as 100`	Specify the neighbour's (CE1) IP address and ASN value.
`iBOS(config-router-af)# neighbour 172.4.5.115 soo 1800:300`	Enable site-of-origin (SOO). The SOO value can be in ASN:NN or A.B.C.D:NN format. If the customer AS (CE1) is multi-homed to ISP, this command ensures that the PE does not advertise the routes back to the same AS/IP address. This command is for a route-origin extended community.

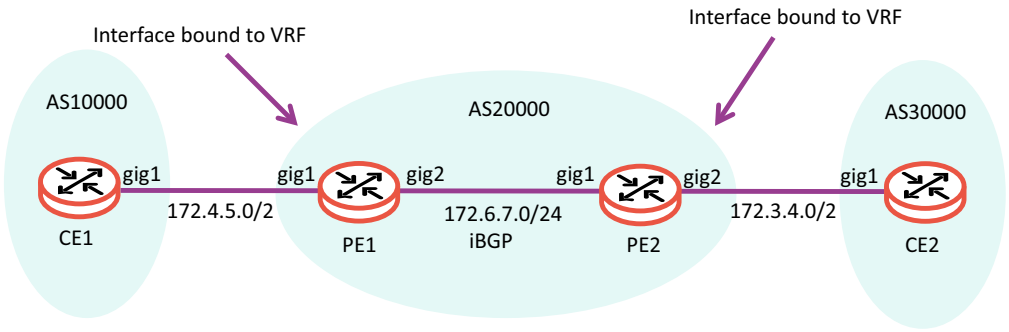
Configure router PE2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`iBOS(config-vrf)# rd 500:50`	Assign a route distinguisher (RD) for the VRF.
`iBOS(config-vrf)# route-target both 100:200`	Specify the 2-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`iBOS(config-vrf)# exit`	Exit VRF mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`iBOS(config-if)# ip vrf forwarding VRF1`	Bind the interface (gigabitethernet2) to the VRF (VRF1).
`iBOS(config-if)# ip address 172.3.4.117/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`iBOS(config-if)# ip address 172.6.7.117/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)# neighbour 172.6.7.116 remote-as 200`	Specify the neighbour's (PE1) IP address (172.6.7.116) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.
`iBOS(config-router)# address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`iBOS(config-router-af)# neighbour 172.6.7.116 activate`	Enable the exchange of routing information with a peer router.
`iBOS(config-router-af)# exit`	Exit Address-Family-VPNv4 mode.
`iBOS(config-router)# address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`iBOS(config-router-af)# neighbour 172.3.4.114 remote-as 30`	Specify the neighbour's (CE2) IP address and ASN value.

Configure Extended Community with 4-Byte ASN

In the following example, CE1, PE1, PE2, and CE2 support 4-byte ASN capability.

Figure 74 Extended Communities — 4-Byte ASN



Note: *PE1 and PE2 should both either be 4-byte-ASN capable or 2-byte-ASN capable. Support for the combination of one 4-byte-ASN capable PE with one 2-byte-ASN-capable PE is currently unavailable.*

To configure Extended Communities with 4-Byte ASN, do the following:

1. Configure router CE1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 172.4.5.115/24`	Configure the IP address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#bgp extended-asn-cap`	Enable 4-octet ASN capability.
`iBOS(config)#router bgp 100000`	Assign the ASN value (100) to the router. The ASN range is <1-65535>.
`iBOS(config-router)#neighbour 172.4.5.116 remote-as 200000`	Specify the neighbour's IP address (172.4.5.116) and the ASN value of the neighbour (200000).

2. Configure router PE1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# bgp extended-asn-cap`	Enable 4-octet ASN capability. Dynamic change from 2-byte to 4-byte capability, or vice versa, is not allowed, unless the VRF is removed.
`iBOS(config)# ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`iBOS(config-vrf)# rd 700000:10`	Assign a 4-byte route distinguisher (RD) for the VRF, which is a unique value on the router. The RD value can be in ASN:NN or A.B.C.D:NN format.
`iBOS(config-vrf)# route-target both 800000:200`	Specify the 4-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`iBOS(config-vrf)# exit`	Exit VRF mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)# ip vrf forwarding VRF1`	Bind the interface (gigabitethernet1) to the VRF (VRF1).
`iBOS(config-if)# ip address 172.4.5.116/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`iBOS(config-if)# ip address 172.6.7.116/24`	Configure the IP address on this interface
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router bgp 200000`	Assign the ASN value (200000) to the router.
`iBOS(config-router)# neighbour 172.6.7.117 remote-as 200`	Specify the neighbour's (PE2) IP address (172.6.7.117) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.
`iBOS(config-router)#address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`iBOS(config-router-af)#neighbour 172.6.7.117 activate`	Enable the exchange of routing information with a peer router.
`iBOS(config-router-af)#exit`	Exit Address-Family-VPNv4 mode.
`iBOS(config-router)#address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`iBOS(config-router-af)#neighbour 172.4.5.115 remote-as 100000`	Specify the neighbour's (CE1) IP address and ASN value.
`iBOS(config-router-af)#neighbour 172.4.5.115 soo 1800000:300`	Enable site-of-origin (SOO). The SOO value can be in ASN:NN or A.B.C.D:NN format. If the customer AS (CE1) is multi-homed to ISP, this command ensures that the PE does not advertise the routes back to the same AS/IP address. This command is for route-origin extended community.

3. Configure router PE2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#bgp extended-asn-cap`	Enable 4-octet ASN capability. Dynamic change from 2-byte to 4-byte capability, or vice versa, is not allowed, unless the VRF is removed.
`iBOS(config)#ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`iBOS(config-vrf)#rd 500000:50`	Assign a 4-byte route distinguisher (RD) for the VRF.
`iBOS(config-vrf)#route-target both 800000:200`	Specify the 4-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`iBOS(config-vrf)#exit`	Exit VRF mode, and return to Configure mode.
`iBOS(config)#interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`iBOS(config-if)#ip vrf forwarding VRF1`	Bind the interface (gigabitethernet1) to the VRF (VRF1).
`iBOS(config-if)#ip address 172.3.4.117/24`	Configure the IP address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 172.6.7.117/24`	Configure the IP address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200000`	Assign the ASN value (200000) to the router.
`iBOS(config-router)#neighbour 172.6.7.116 remote-as 200000`	Specify the neighbour's (PE2) IP address (172.6.7.116) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.
`iBOS(config-router)#address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`iBOS(config-router-af)#neighbour 172.6.7.116 activate`	Enable the exchange of routing information with a peer router.
`iBOS(config-router-af)#exit`	Exit Address-Family-VPNv4 mode.
`iBOS(config-router)#address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`iBOS(config-router-af)#neighbour 172.3.4.114 remote-as 300000`	Specify the neighbour's (CE1) IP address and ASN value.

4. Configure router CE2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 172.3.4.114/24`	Configure the IP address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#bgp extended-asn-cap`	Enable 4-octet ASN capability.
`iBOS(config)#router bgp 300000`	Assign the ASN value (300000) to the router.
`iBOS(config-router)#neighbour 172.3.4.117 remote-as 200000`	Specify the neighbour's IP address (172.3.4.117) and the ASN value of the neighbour (200000).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **ip vrf forwarding**
- **neighbour remote-as**
- **ip vrf, route-target both**
- **address-family vpnv4 unicast**
- **neighbour activate**
- **address-family ipv4 vrf**
- **neighbour soo**
- **bgp extended-asn-cap**

To verify the configuration, use the following commands:

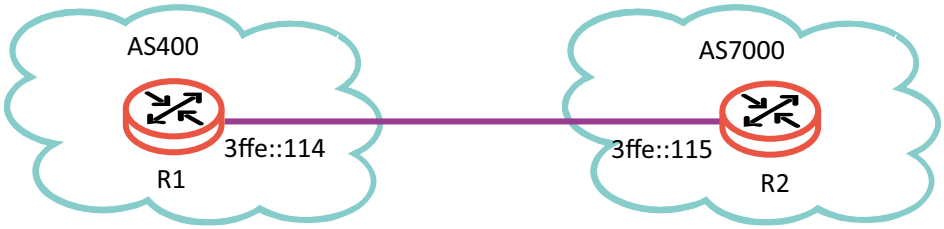
- **show ip bgp**
- **show ip bgp neighbours**
- **show ip bgp vpnv4 all**

Configure IPv6 Route Refresh

Route-Refresh capability in BGP allows the dynamic exchange of route refresh request between BGP speakers and subsequent re-advertisement of the respective Adj-RIB-Out. To advertise Route Refresh capability to a peer, a BGP speaker uses BGP Capabilities Advertisement [BGP-CAP]. By advertising the Route Refresh capability to a peer, a BGP speaker conveys to the peer that the speaker is capable of receiving, and properly handling. the ROUTE-REFRESH message from the peer.

The following example is for Route-Refresh capability for the IPv6 address family.

Figure 75 IPv6 Route Refresh



Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 10.20.30.1/24`	Configure the IP address on this interface
`iBOS(config-if)#ipv6 address 3ffe::114/64`	Configure the IPv6 address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 400`	Assign the ASN value (400) to the router.
`iBOS(config-router)#neighbour 3ffe::115 remote-as 7000`	Specify the neighbour's IPv6 address (3ffe::115) and the ASN value of the neighbour (7000).
`iBOS(config-router)#address-family ipv6 unicast`	Exchange IPv6 capabilities, and enter IPv6 mode.
`iBOS(config-router-af)#neighbour 3ffe::115 activate`	Activate the IPv6 peer.
`iBOS(config-router-af)#redistribute static`	Advertise the static routes.

Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 10.20.30.2/24`	Configure the IP address on this interface
`iBOS(config-if)#ipv6 address 3ffe::115/64`	Configure the IPv6 address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 7000`	Assign the ASN value (7000) to the router.
`iBOS(config-router)#neighbour 3ffe::114 remote-as 400`	Specify the neighbour's IPv6 address (3ffe::114) and the ASN value of the neighbour (400).

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 10.20.30.2/24`	Configure the IP address on this interface
`iBOS(config-if)#ipv6 address 3ffe::115/64`	Configure the IPv6 address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config-router)#address-family ipv6 unicast`	Exchange IPv6 capabilities, and enter IPv6 mode.
`iBOS(config-router-af)#neighbour 3ffe::114 activate`	Activate the IPv6 peer.
`iBOS(config-router-af)#redistribute static`	Advertise the static routes.
`iBOS(config-router-af)#exit`	Exit twice.
`iBOS#clear ip bgp 3ffe::115 in`	Send the Route-Refresh message to the IPv6 peer.
`iBOS#clear ip bgp * in`	Send the Route-Refresh message to all IPv4 and IPv6 peers
`iBOS#clear ip bgp 400 in`	Send the Route-Refresh message to all IPv4 and IPv6 peers in the ASN-400.

Refer *iBOS Command Reference* for detailed descriptions of following commands:

- **neighbour remote-as**
- **address-family ipv6 unicast**
- **neighbour activate**
- **redistribute static**
- **clear ip bgp**

To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**
- **show bgp ipv6**
- **show ipv6 route**

Configure Nexthop Tracking

Nexthop tracking is used to notify the BGP process asynchronously whenever there is any change in the IGP routes. It reduces the convergence time of BGP routes when IGP routes are changed.

Figure 76 BGP Nexthop Tracking

To configure BGP Nexthop Tracking, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 100.100.100.100/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router. The ASN range is <1-65535>.
`iBOS(config-router)#neighbour 200.200.200.200 remote-as 200`	Specify the neighbour's IP address (200.200.200.200) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 200.200.200.200 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.4.5.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.4.5.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.6.7.0/24 area 0`	Advertise the network in Area 0.

3. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 200.200.200.200/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#neighbour 100.100.100.100 remote-as 200`	Specify the neighbour's IP address (100.100.100.100) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 100.100.100.100 update-source lo`	Specify the routing update source.
`iBOS(config-router)#neighbour 300.300.300.300 remote-as 200`	Specify the neighbour's IP address (300.300.300.300) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 300.300.300.300 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.6.7.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.1.2.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#bgp nexthop trigger enable`	Enable Nexthop tracking.

4. Configure router R4:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.1.2.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.3.4.0/24 area 0`	Advertise the network in Area 0.

5. Configure router R5:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 300.300.300.300/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#neighbour 200.200.200.200 remote-as 200`	Specify the neighbour's IP address (200.200.200.200) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 200.200.200.200 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.3.4.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **neighbour remote-as**
- **ip address**
- **neighbour update-source**
- **network area**
- **redistribute connected**
- **bgp nexthop trigger enable**

To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**
- **show ip bgp**
- **show bgp nexthop-tracking**
- **show ip bgp scan**

Configure Nexthop Tracking Delay Timer

The delay interval between routing table walks can be configured for nexthop delay tracking. This time determines how long BGP waits before it starts walking the full BGP routing table after receiving notification from NSM about a next-hop change.

Figure 77 Topology for Nexthop Tracking Delay Timer

To configure the above example, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 100.100.100.100/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router. The ASN range is <1-65535>.
`iBOS(config-router)#neighbour 200.200.200.200 remote-as 200`	Specify the neighbour's IP address (200.200.200.200) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 200.200.200.200 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.4.5.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.4.5.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.6.7.0/24 area 0`	Advertise the network in Area 0.

3. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 200.200.200.200/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#neighbour 100.100.100.100 remote-as 200`	Specify the neighbour's IP address (100.100.100.100) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 100.100.100.100 update-source lo`	Specify the routing update source.
`iBOS(config-router)#neighbour 300.300.300.300 remote-as 200`	Specify the neighbour's IP address (300.300.300.300) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 300.300.300.300 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.6.7.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.1.2.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#bgp nexthop trigger enable`	Enable nexthop tracking.
`iBOS(config)#bgp nexthop trigger delay 20`	Configure the nexthop trigger-delay time interval.

4. Configure router R4:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.1.2.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#network 172.3.4.0/24 area 0`	Advertise the network in Area 0.

5. Configure router R5:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface lo`	Specify the loopback interface, and enter Interface mode.
`iBOS(config-if)#ip address 300.300.300.300/32`	Configure the IP address on this interface.
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#neighbour 200.200.200.200 remote-as 200`	Specify the neighbour's IP address (200.200.200.200) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 200.200.200.200 update-source lo`	Specify the routing update source.
`iBOS(config-router)#exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)#router ospf 1`	Configure the OSPF process (1).
`iBOS(config-router)#network 172.3.4.0/24 area 0`	Advertise the network in Area 0.
`iBOS(config-router)#redistribute connected`	Redistribute the connected routes.

Refer *iBOS Command Reference* for detailed description of following commands:

- **neighbour remote-as**
- **ip address**
- **neighbour update-source**
- **network area**
- **redistribute connected**
- **bgp nexthop trigger enable**

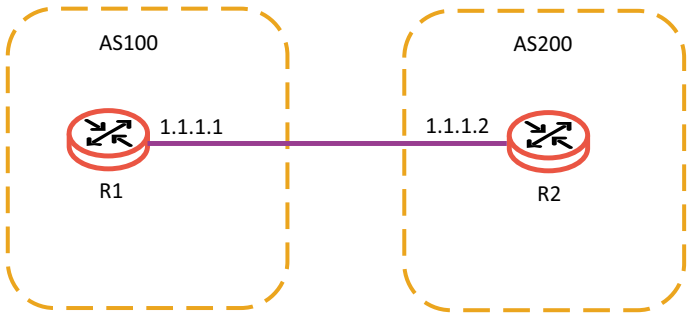
To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**
- **show ip bgp**
- **show bgp nexthop-tracking**
- **show ip bgp scan**

Configure BGP Graceful Reset

The graceful restart mechanism for BGP session reset (the BGP daemon is not restarted) is used so that any changes in network configuration do not affect packet forwarding. The **bgp graceful-restart graceful-reset** command invokes graceful restart when a configuration change forces a peer reset. Graceful restart is invoked only when these CLI configuration changes force a peer reset.

Figure 78 BGP Graceful Reset



Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the router.
`iBOS(config-router)#bgp graceful-restart`	Enable BGP graceful restart support.
`iBOS(config-router)#bgp graceful-restart graceful-reset`	Configure to invoke graceful restart when a configuration change forces a peer reset.
`iBOS(config-router)#neighbour 1.1.1.2 remote-as 200`	Specify the neighbour's IP address (1.1.1.2) and the ASN value of the neighbour (200).
`iBOS(config-router)#neighbour 1.1.1.2 capability graceful-restart`	Specify the neighbour's IP address (1.1.1.2) for which the graceful restart capability supports.

Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#bgp graceful-restart`	Enable BGP graceful restart support.
`iBOS(config-router)#bgp graceful-restart graceful-reset`	Configure to invoke graceful restart when a configuration change forces a peer reset.

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Specify the neighbour's IP address (1.1.1.1) and the ASN value of the neighbour (100).
`iBOS(config-router)#neighbour 1.1.1.1 capability graceful-restart`	Specify the neighbour's IP address (1.1.1.1) for which the graceful restart capability supports.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp graceful-restart**
- **bgp graceful-restart graceful-reset**
- **neighbour remote-as**
- **neighbour capability graceful-restart**

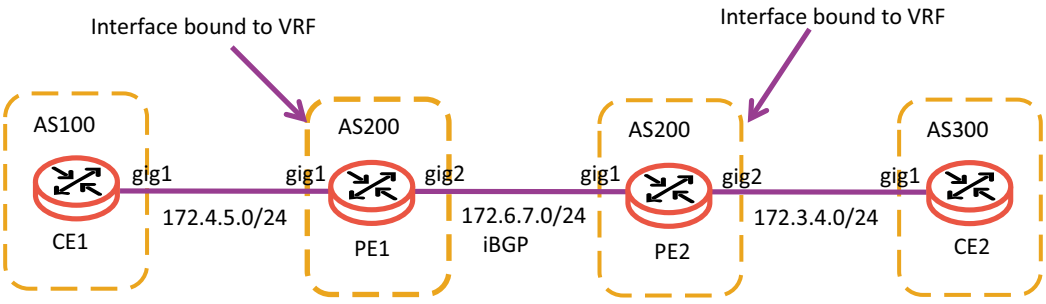
To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**
- **show ip bgp**
- **show ip route database bgp**
- **show ip route database**
- **show ip route**

Configure BGP-MPLS Graceful Restart

BGP-MPLS graceful restart provides a mechanism to save necessary MPLS forwarding-state information in NSM and synchronize with the VRF table when BGP goes offline in the control plane.

Figure 79 BGP-MPLS Graceful Restart



Configure router CE1:

`CE1# configure terminal`	Enter Configure mode.
`CE1(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`CE1(config-if)# ip address 172.4.5.115/24`	Configure the IP address on this interface
`CE1(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`CE1(config)# router bgp 100`	Assign the ASN value (100) to the router. The ASN range is <1-65535>.
`CE1(config-router)# neighbour 172.4.5.116 remote-as 200`	Specify the neighbour's IP address (172.4.5.116) and the ASN value of the neighbour (200).

Configure router PE1:

`PE1# configure terminal`	Enter Configure mode.
`PE1(config)# ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`PE1(config-vrf)# rd 100:10`	Assign a route distinguisher (RD) for the VRF, which is a unique value on the router. The RD value can be in ASN:NN or A.B.C.D:NN format.
`PE1(config-vrf)# route-target both 100:200`	Specify the 2-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`PE1(config-vrf)# exit`	Exit VRF mode, and return to Configure mode.
`PE1(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`PE1(config-if)# ip vrf forwarding VRF1`	Bind the interface (gigabitethernet1) to the VRF (VRF1).
`PE1(config-if)# ip address 172.4.5.116/24`	Configure the IP address on this interface
`PE1(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`PE1(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`PE1(config-if)# ip address 172.6.7.116/24`	Configure the IP address on this interface
`PE1(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`PE1(config)# router bgp 200`	Assign the ASN value (200) to the router.
`PE1(config-router)# bgp graceful-restart`	Enable BGP graceful restart support.
`PE1(config-router)# bgp graceful-restart restart-time 180`	Configure the maximum time (180) required for neighbour(s) to restart.
`PE1(config-router)# bgp graceful-restart stalepath-time 150`	Configure the maximum time (150) to retain stale paths from the restarting neighbour(s).
`PE1(config-router)# bgp update-delay 120`	Configure the maximum time to defer initial route selection.
`PE1(config-router)# neighbour 172.6.7.117 remote-as 200`	Specify the neighbour's (PE2) IP address (172.6.7.117) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.

`PE1# configure terminal`	Enter Configure mode.
`PE1(config)# ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`PE1(config-vrf)# rd 100:10`	Assign a route distinguisher (RD) for the VRF, which is a unique value on the router. The RD value can be in ASN:NN or A.B.C.D:NN format.
`PE1(config-vrf)# route-target both 100:200`	Specify the 2-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`PE1(config-vrf)# exit`	Exit VRF mode, and return to Configure mode.
`PE1(config)# interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`PE1(config-if)# ip vrf forwarding VRF1`	Bind the interface (gigabitethernet1) to the VRF (VRF1).
`PE1(config-if)# ip address 172.4.5.116/24`	Configure the IP address on this interface
`PE1(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`PE1(config)# interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`PE1(config-if)# ip address 172.6.7.116/24`	Configure the IP address on this interface
`PE1(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`PE1(config-router)# address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`PE1(config-router-af)# neighbour 172.6.7.117 activate`	Enable the exchange of routing information with a peer router.
`PE1(config-router-af)# neighbour 172.6.7.117 capability graceful-restart`	Specify the neighbour IP address (172.6.7.117) for which the graceful restart capability supports.
`PE1(config-router-af)# exit`	Exit Address-Family-VPNv4 mode.
`PE1(config-router)# address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`PE1(config-router-af)# neighbour 172.4.5.115 remote-as 100`	Specify the neighbour's (CE1) IP address and ASN value.

Configure router PE2:

`PE2#configure terminal`	Enter Configure mode.
`PE2(config)#ip vrf VRF1`	Specify the name of the VRF (VRF1) to be created.
`PE2(config-vrf)# rd 500:50`	Assign a route distinguisher (RD) for the VRF.
`PE2(config-vrf)#r oute-target both 100:200`	Specify the 2-Octet AS specific or IPv4 specific Transitive Route-Target extended community attribute.
`PE2(config-vrf)#exit`	Exit VRF mode, and return to Configure mode.
`PE2(config)#interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`PE2(config-if)#ip vrf forwarding VRF1`	Bind the interface (gigabitethernet2) to the VRF (VRF1).
`PE2(config-if)#ip address 172.3.4.117/24`	Configure the IP address on this interface
`PE2(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`PE2(config)#interface gigabitethernet2`	Specify the interface (gigabitethernet2)to configure, and enter Interface mode.
`PE2(config-if)#ip address 172.6.7.117/24`	Configure the IP address on this interface
`PE2(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`PE2(config)#router bgp 200`	Assign the ASN value (200) to the router.
`PE2(config-router)#bgp graceful-restart`	Enable BGP graceful restart support.
`PE2(config-router)#bgp graceful-restart restart-time 180`	Configure the maximum time (180) required for neighbour(s) to restart.
`PE2(config-router)#bgp graceful-restart stalepath-time 150`	Configure the maximum time (150) to retain stale paths from the restarting neighbour(s).
`PE2(config-router)#neighbour 172.6.7.116 remote-as 200`	Specify the neighbour's (PE1) IP address (172.6.7.116) and the ASN value of the neighbour (200). In this case, it is an iBGP connection, so both PE1 and PE2 are in the same AS.
`PE2(config-router)#address-family vpnv4 unicast`	Enable the exchange of VPNv4 routing information among ISP PE-routers, and enter Address-Family-VPNv4 mode.
`PE2(config-router-af)# neighbour 172.6.7.116 activate`	Enable the exchange of routing information with a peer router.
`PE2(config-router-af)# neighbour 172.6.7.116 capability graceful-restart`	Specify the neighbour's IP address (172.6.7.116) for which the graceful restart capability supports.
`PE2(config-router-af)# exit`	Exit Address-Family-VPNv4 mode.
`PE2(config-router)# address-family ipv4 vrf VRF1`	Enable the exchange of VRF routing information among ISP PE-routers, and enter Address-Family-VRF mode.
`PE2(config-router-af)# neighbour 172.3.4.114 remote-as 300`	Specify the neighbour's (CE2) IP address and ASN value.

Configure router CE1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure, and enter Interface mode.
`iBOS(config-if)#ip address 172.3.4.114/24`	Configure the IP address on this interface
`iBOS(config-if)#exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)#router bgp 300`	Assign the ASN value (300) to the router.
`iBOS(config-router)#neighbour 172.3.4.117 remote-as 200`	Specify the neighbour's IP address (172.3.4.117) and the ASN value of the neighbour (200).

Refer *iBOS Command Reference* for detailed description of following commands:

- **ip vrf forwarding**
- **neighbour remote-as**
- **ip vrf, route-target both**
- **address-family vpnv4 unicast**
- **neighbour activate**
- **address-family ipv4 vrf**
- **bgp-graceful restart**
- **bgp graceful-restart restart-time**
- **bgp graceful-restart stalepath-time**
- **bgp update-delay**

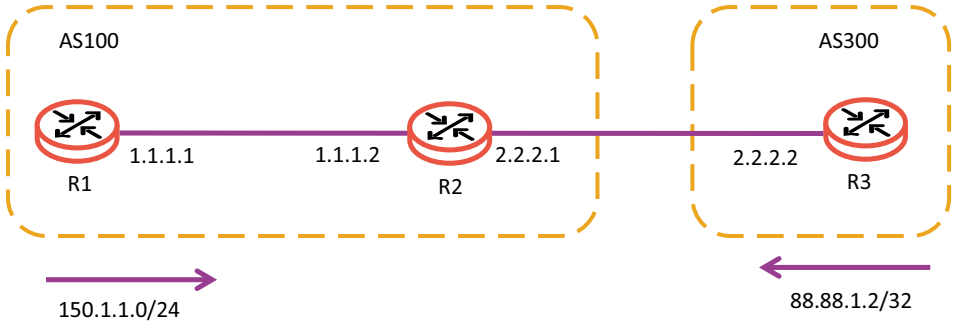
To verify the configuration, use the following commands:

- **show ip bgp summary**
- **show ip bgp neighbours**
- **show ip bgp vpnv4 all**
- **show mpls ilm-table**
- **show mpls vrf-table**
- **show ip route vrf database**
- **show ip bgp neighbours**
- **show ip route vrf**

Configure BGP Distance

The administrative distance rates the trustworthiness of a routing information source. Administrative distance in BGP can be configured for a specific address family.

Figure 80 Administrative Distance for IPv4 BGP



The default values for administrative distance are:

Route Source	Default Distance Value
External route	20
Internal route	200
Local route	200

To configure BGP Distance, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the router.
`iBOS(config-router)#network 150.1.1.0/24`	Specify the network to be advertised by the BGP routing process.
`iBOS(config-router)#neighbour 1.1.1.2 remote-as 100`	Specify the neighbour's IP address and ASN value.

2. Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the router.
`iBOS(config-router)#neighbour 2.2.2.2 remote-as 200`	Specify the neighbour's IP address and ASN value.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Specify the neighbour's IP address and the ASN value of another neighbour.

`iBOS(config-router)#distance bgp 12 13 120`	Configure the administrative distance for external, internal, and local routes received.
`iBOS(config-router)#aggregate-address 50.50.0.0/16 summary-only`	Configure a non-AS-set aggregate route on R2. The local distance is applied to this route.

3. Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#network 88.88.1.2/32`	Specify the network to be advertised by the BGP routing process.
`iBOS(config-router)#neighbour 2.2.2.1 remote-as 100`	Specify the neighbour's IP address and ASN value.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **network**
- **neighbour remote-as**
- **distance bgp**
- **aggregate-address summary-only**

To verify the configuration, use the following commands:

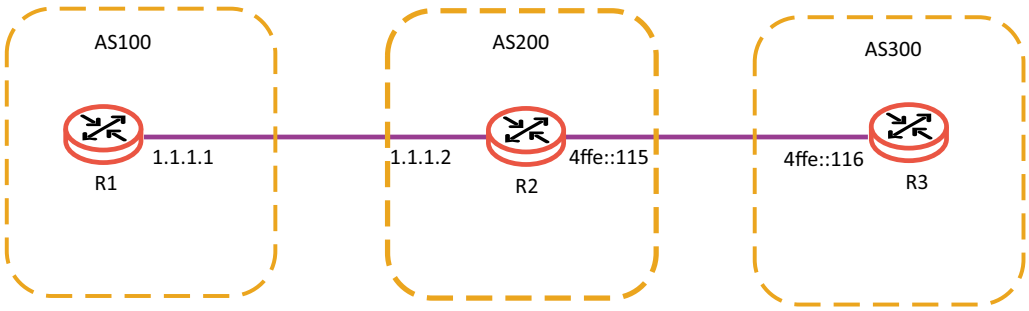
- **show ip bgp summary**
- **show ip route database bgp**
- **show ip route database**
- **show ip route**
- **show ip bgp**

Configure BGP Weight per Peer

A different weight can be assigned per address family of a peer. For example, a system can be configured to prefer VPN4 routes from peer A and IPv4 routes from peer B.

If the neighbour weight command is given under a specific address-family mode, the peer weight is set for that specific address family. If the address family is not specifically set, the weight is updated for the default address-family.

Figure 81 BGP Weight Per Peer



Configure router R1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the router.
`iBOS(config-router)#neighbour 1.1.1.2 remote-as 200`	Specify the neighbour's IP address and ASN value.

Configure router R2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the router.
`iBOS(config-router)#neighbour 1.1.1.1 remote-as 100`	Specify the neighbour's IP address and ASN value.
`iBOS(config-router)#neighbour 1.1.1.1 weight 500`	Add a weight of 500 to all the routes coming from the neighbour, 1.1.1.1 (only IPv4 routes).
`iBOS(config-router)#neighbour 4ffe:116 remote-as 300`	Specify the IPv6 neighbour's address and ASN value.
`iBOS(config-router)#address-family ipv6 unicast`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)#neighbour 4ffe::116 activate`	Activate the peer in IPv6 Address-Family mode.
`iBOS(config-router-af)#neighbour 4ffe::116 weight 700`	Add a weight of 700 to all routes coming from R3 (only IPv6 routes).
`iBOS (config-router)#address-family ipv6 labeled-unicast`	Enter 6PE address-family mode.

`iBOS (config-router-af)#neighbour 1.1.1.1 activate`	Activate the IPv4 neighbour to send IPv6 routes.
`iBOS(config-router-af)#neighbour 1.1.1.1 weight 3000`	Add a weight of 3000 to all 6PE routes coming from R1.

Configure router R3:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router bgp 300`	Assign the ASN value (300) to the router.
`iBOS(config-router)#neighbour 4ffe::115 remote-as 200`	Define the IPv6 neighbour (R2).
`iBOS(config-router)#address-family ipv6 unicast`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)#neighbour 4ffe::115 activate`	Activate the peer in Address-Family mode.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **neighbour remote-as**
- **neighbour weight**
- **address-family ipv6 unicast**
- **neighbour activate**
- **address-family ipv6 labeled unicast**

To verify the configuration, use the following commands:

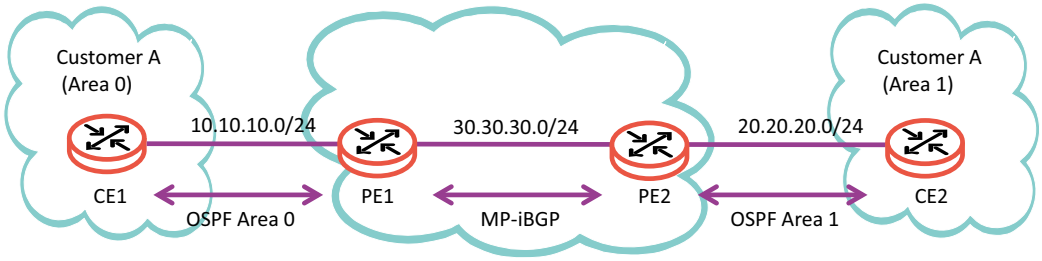
- **show ip bgp summary**
- **show ip bgp neighbours**
- **show bgp ipv6 neighbours**
- **show ip bgp**
- **show bgp ipv6**
- **show ip route**
- **show ipv6 route**

Configure BGP-OSPF as PE-CE Protocol for VPNs

In an MPLS VPN environment, customer networks are connected to an MPLS VPN-enabled provider backbone. As shown in Figure 82, Customer A areas, Areas 0 and 1, are connected to an MPLS VPN–enabled provider network. Area 0 and Area 1 have routers CE1 and CE2 running OSPF. MP-iBGP is used between PE1 and PE2 to propagate routes between Site 1 (Area 0) and Site 2 (Area 1). Traditional OSPF-BGP redistribution is performed at PE routers, PE1 and PE2. In this case, routes distributed by CE1 into the MP-iBGP cloud are sent to CE2 as external routes, even though both CE1 and CE2 belong to the same customer.

This behaviour can be changed with the additional domain ID configuration. Each VRF should be configured a domain ID on the PE routers. If a PE router gets a route through the MP-iBGP cloud and has to send to any customer site, it checks the domain ID value against the list of stored domain ID values. If the incoming domain ID matches any of the stored IDs, that route is inserted into the customer site with the same type, as it was inserted into the MP-BGP cloud; otherwise, it is inserted as external route.

Figure 82 OSPF as PE-CE Protocol



To configure OSPF as PE-CE protocol, do the following:

1. Configure router CE1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the routing process and specify the Process ID (1).
`iBOS(config-router)#network 10.10.10.0/24 area 0`	Advertise the network in OSPF.

2. Configure router PE1:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#ip vrf ws`	Specify the name of the VRF (ws) to be created.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure.
`iBOS(config-if)#ip vrf forwarding`	Remove interface gigabitethernet1 from passive mode.
`iBOS(config-if)#exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)#router ospf 1 ws`	Configure OSPF for VRF.
`iBOS(config-router)#network 10.10.10.0/24 area 0`	Advertise the network for OSPF adjacency with CE1.
`iBOS(config-router)#domain-id 1.1.1.1`	Configure the primary domain ID.
`iBOS(config-router)#domain-id 2.2.2.2 secondary`	Configure a secondary domain ID.
`iBOS(config-router)#domain-id 3.3.3.3 secondary`	Configure a secondary domain ID.
`iBOS(config-router)#exit`	Exit Router mode and return to Configure mode.

`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the BGP router.
`iBOS(config-router)#neighbour 30.30.30.2 remote-as 100`	Configure neighbour 30.30.30.2 for iBGP.
`iBOS(config-router)#address-family vpnv4 unicast`	Enter Address-Family-VPNv4 mode.
`iBOS(config-router-af)#neighbour 30.30.30.2 activate`	Activate neighbour 30.30.30.2.
`iBOS(config-router-af)#exit-address-family`	Exit Address Family mode and return to Router mode.
`iBOS(config-router)#address-family ipv4 vrf ws`	Enter Address-Family-VRF mode.
`iBOS(config-router)#redistribute ospf`	Specify redistributing routes from OSPF into BGP.
`iBOS(config-router-af)#exit-address-family`	Exit Address Family mode and return to Router mode.

3. Configure router PE2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#interface gigabitethernet1`	Specify the interface (gigabitethernet1)to configure.
`iBOS(config-if)#ip vrf forwarding`	Remove interface gigabitethernet1 from passive mode.
`iBOS(config-if)#exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)#router ospf 1 ws`	Configure OSPF for VRF.
`iBOS(config-router)#network 20.20.20.0/24 area 0`	Advertise the network for OSPF adjacency with CE1.
`iBOS(config-router)#domain-id 1.1.1.1`	Configure the primary domain ID.
`iBOS(config-router)#domain-id 2.2.2.2 secondary`	Configure a secondary domain ID.
`iBOS(config-router)#domain-id 3.3.3.3 secondary`	Configure a secondary domain ID.
`iBOS(config-router)#exit`	Exit Router mode and return to Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the BGP router.
`iBOS(config-router)#neighbour 30.30.30.1 remote-as 100`	Configure neighbour 30.30.30.1 for iBGP.
`iBOS(config-router)#address-family vpnv4 unicast`	Enter Address-Family-VPNv4 mode.

`iBOS(config-router-af)#neighbour 30.30.30.2 activate`	Activate neighbour 30.30.30.2.
`iBOS(config-router-af)#exit-address-family`	Exit Address Family mode and return to Router mode.
`iBOS(config-router)#address-family ipv4 vrf ws`	Enter Address-Family-VRF mode.
`iBOS(config-router)#redistribute ospf`	Specify redistributing routes from OSPF into BGP.
`iBOS(config-router-af)#exit-address-family`	Exit Address Family mode and return to Router mode.

4. Configure router CE2:

`iBOS#configure terminal`	Enter Configure mode.
`iBOS(config)#router ospf 1`	Configure the routing process, and specify the Process ID (1).
`iBOS(config-router)#network 20.20.20.0/24 area 0`	Advertise the network in OSPF.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **network area**
- **ip vrf**
- **ip vrf forwarding**
- **router ospf**
- **domain-id**
- **neighbour remote-as**
- **address-family vpnv4 unicast**
- **neighbour activate**
- **address-family ipv4 vrf**
- **redistribute**

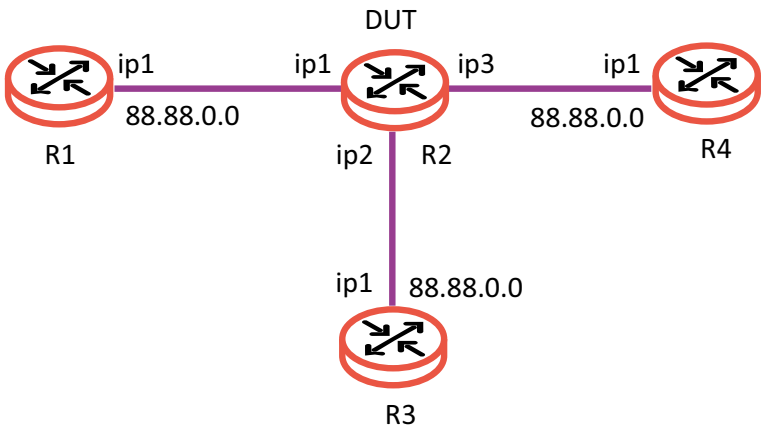
To verify the configuration, use the command **show ip bgp vpnv4 all**.

BGP Multipath for IPv4 and IPv6

The BGP Multipath feature supports both IPv4 and IPv6 prefixes and allows load-balancing traffic of multiple BGP routes. iBOS supports both iBGP and eBGP routes. For eBGP, the routes should arrive from same AS number.

Configure Multipath iBGP for IPv4

Figure 83 Multipath iBGP for IPv4



To configure Multipath iBGP for IPv4, do the following

1. Configure router R1:

`R1# configure terminal`	Enter the Configure mode.
`R1(config)# router bgp 100`	Assign the ASN value (100) to the BGP router.
`R1(config-router)# bgp router-id 2.2.2.2`	Configure a fixed Router ID (2.2.2.2).
`R1(config-router)# redistribute static`	Redistribute the static routes.
`R1(config-router)# neighbour 30.30.30.9 activate` `R1(config-router)# neighbour 30.30.30.9 remote-as 100`	Configure neighbour 30.30.30.9 for iBGP.
`R1(config-router)# exit`	Exit the BGP Router mode and return to the Configure mode.
`R1(config)# ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

2. Configure router R3:

`R3# configure terminal`	Enter the Configure mode.
`R3(config)# router bgp 100`	Assign the ASN value (100) to the BGP router.
`R3(config-router)# bgp router-id 4.4.4.4`	Configure a fixed Router ID (4.4.4.4).
`R3(config-router)# redistribute static`	Redistribute the static routes.
`R3(config-router)# neighbour 40.40.40.9 activate` `R3(config-router)# neighbour 40.40.40.9 remote-as 100`	Configure neighbour 40.40.40.9 for iBGP.

`R3(config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`R3(config)# ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

3. Configure router R4:

`R4# configure terminal`	Enter the Configure mode.
`R4(config)# router bgp 100`	Assign the ASN value (100) to the BGP router.
`R4(config-router)# bgp router-id 6.6.6.6`	Configure a fixed Router ID (6.6.6.6).
`Rtr4(config-router)# redistribute static`	Redistribute the static routes.
`R4(config-router)# neighbour 50.50.50.9 activate R4(config-router)# neighbour 50.50.50.9 remote-as 100`	Configure neighbour 50.50.50.9 for iBGP.
`R4(config)# exit`	Exit the BGP Router mode and return to the Configure mode.
`R4(config)# ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

4. Configure DUT on router R2:

`R# configure terminal`	Enter the Configure mode.
`R2(config)# router bgp 100`	Assign the ASN value (100) to the BGP router.
`R2(config-router)# max-paths ibgp 2`	Configure iBGP max-paths (2).
`R2(config)# bgp router-id 9.9.9.9`	Configure a fixed Router ID (9.9.9.9).
`R2(config-router)# neighbour 30.30.30.2 activate #R2(config-router)# neighbour 30.30.30.2 remote-as 100`	Configure neighbour 30.30.30.2 for iBGP.
`R2(config-router)# neighbour 40.40.40.4 activate R2(config-router)# neighbour 40.40.40.4 remote-as 100`	Configure neighbour 40.40.40.4 for iBGP.
`R2(config-router)# neighbour 50.50.50.6 activate tr2(config-router)# neighbour 50.50.50.6 remote-as 100`	Configure neighbour 50.50.50.6 for iBGP.
`R2(config-router)# exit`	Exit the Router mode and return to Configure mode.

To verify all available multipaths, use command show ip bgp on DUT (router R2):

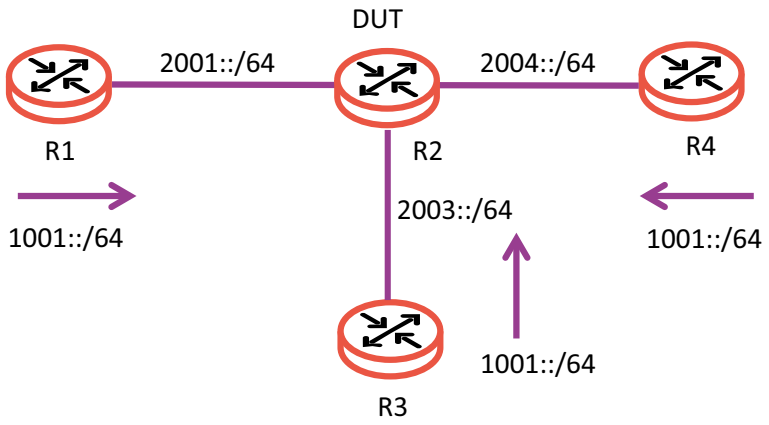
```
R2#show ip bgp 88.88.0.0
BGP routing table entry for 88.88.0.0/16
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    40.40.40.4 from 40.40.40.4 (4.4.4.4)
      Origin incomplete metric 0, localpref 100, valid, internal,
multipath-candidate, installed
      Last update: Sat Jan  3 00:50:38 1970

  Local
    50.50.50.6 from 50.50.50.6 (6.6.6.6)
      Origin incomplete metric 0, localpref 100, valid, internal,
multipath-candidate
      Last update: Sat Jan  3 00:50:28 1970

  Local
    30.30.30.2 from 30.30.30.2 (2.2.2.2)
      Origin incomplete metric 0, localpref 100, valid, internal,
multipath-candidate, installed, best
      Last update: Sat Jan  3 00:50:20 1970
```

To verify routes installed, use the command **show ip route** on router R2:

```
R2#show ip route

Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default

C       30.30.30.0/24 is directly connected, ge3
C       40.40.40.0/24 is directly connected, ge5
C       50.50.50.0/24 is directly connected, ge7
B       88.88.0.0/16 [200/0] via 30.30.30.2, ge3, 00:00:44
                     [200/0] via 40.40.40.4, ge5, 00:00:44
C       90.90.90.90/32 is directly connected, lo
C       127.0.0.0/8 is directly connected, lo
R2#
```

Configure Multipath iBGP for IPv6

Figure 84 Multipath iBGP for IPv6



To configure Multipath iBGP for IPv6, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)#bgp router-id 1.1.1.1`	Configure a fixed Router ID (1.1.1.1).
`iBOS(config-router)#neighbour 2001::2 remote-as 100`	Configure neighbour 2001::2 for iBGP.
`iBOS(config-router)#address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)#redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)#neighbour 2001::2 activate`	Activate neighbour 2001::2.
`iBOS (config-router-af)#exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS (config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS (config)#ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

2. Configure router R3

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)#bgp router-id 3.3.3.3`	Configure a fixed Router ID (3.3.3.3).
`iBOS(config-router)#neighbour 2003::2 remote-as 100`	Configure neighbour 2003::2 for iBGP.

`iBOS(config-router)#address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)#redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)#neighbour 2003::2 activate`	Activate neighbour 2003::2.
`iBOS (config-router-af)#exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS (config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS (config)#ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

3. Configure router R4:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)#bgp router-id 4.4.4.4`	Configure a fixed Router ID (4.4.4.4).
`iBOS(config-router)#neighbour 2004::2 remote-as 100`	Configure neighbour 2004::2 for iBGP.
`iBOS(config-router)#address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)#redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)#neighbour 2004::2 activate`	Activate neighbour 2004::2.
`iBOS (config-router-af)#exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS (config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS (config)#ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

4. Configure Multipath iBGP IPv6 for DUT:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)#bgp router-id 2.2.2.2`	Configure a fixed Router ID (2.2.2.2).
`iBOS (config-router)#max-paths ibgp 2`	Configure iBGP maximum paths (2).
`iBOS(config-router)#neighbour 2001::1 remote-as 100`	Configure neighbour 2001::1 for iBGP.
`iBOS(config-router)#neighbour 2003::3 remote-as 100`	Configure neighbour 2003::3 for iBGP.

`iBOS(config-router)#neighbour 2004::4 remote-as 100`	Configure neighbour 2004::4 for iBGP.
`iBOS(config-router)#address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS (config-router-af)#neighbour 2001::1 activate`	Activate neighbour 2001::1.
`iBOS(config-router-af)#redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)#neighbour 2004::2 activate`	Activate neighbour 2004::2.
`iBOS (config-router-af)#neighbour 2003::3 activate`	Activate neighbour 2003::3.
`iBOS (config-router-af)#neighbour 2004::4 activate`	Activate neighbour 2004::4.
`iBOS (config-router-af)#exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS(config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.

To verify configuration for Multipath iBGP for IPv6, use the following commands:

- **show bgp**
- **show bgp ipv6**
- **show bgp summary**
- **show ipv6 route**

To verify all available multiple paths, use command **show bgp ipv6** on router R2:

```
iBOS#show bgp ipv6 1001::/64

BGP routing table entry for 1001::/64
Paths: (3 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    2004::4(fe80::5054:ff:feae:1f85) from 2004::4 (4.4.4.4)
    (fe80::5054:ff:feae:1f85)
      Origin incomplete metric 0, localpref 100, valid, internal,
multipath-candidate
      Last update: Wed Oct 17 12:52:46 2012

  Local
    2001::1(fe80::5054:ff:fe76:f9a2) from 2001::1 (1.1.1.1)
    (fe80::5054:ff:fe76:f9a2)
      Origin incomplete metric 0, localpref 100, valid, internal,
multipath-candidate, installed, best
      Last update: Wed Oct 17 12:36:57 2012

  Local
    2003::3(fe80::5054:ff:fe62:253a) from 2003::3 (3.3.3.3)
    (fe80::5054:ff:fe62:253a)
      Origin incomplete metric 0, localpref 100, valid, internal,
```

```
multipath-candidate, installed
      Last update: Wed Oct 17 12:25:27 2012
iBOS#
```

To verify list of routes installed, use command **show ipv6 route** on router Rtr 2:

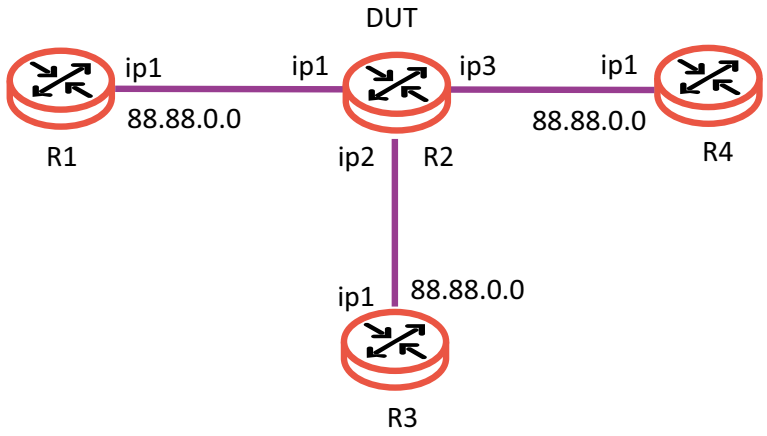
```
iBOS#show ipv6 route
IPv6 Routing Table
Codes: K - kernel, C - connected, S - static, CL - client, B - BGP
       O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       * - candidate default

C     ::1/128 via ::, lo, 01:12:53
B     1001::/64 [200/0] via fe80::5054:ff:fe76:f9a2,
gigabitethernet1, 00:19:09
            [200/0] via fe80::5054:ff:fe62:253a, gigabitethernet3,
00:19:09
C     2001::/64 via ::, gigabitethernet1, 01:12:48
C     2003::/64 via ::, gigabitethernet3, 01:12:48
C     2004::/64 via ::, gigabitethernet4, 01:12:48
C     fe80::/64 via ::, gigabitethernet0, 01:12:53
K     ff00::/8 via ::, virbr0, 01:12:53
iBOS#
```

To verify list of connected neighbours, use command **show bgp summary** on router Rtr2:

```
iBOS#show bgp summary
BGP router identifier 2.2.2.2, local AS number 100
BGP table version is 5
1 BGP AS-PATH entries
0 BGP community entries
1  Configured ebgp ECMP multipath: Currently set at 1
2  Configured ibgp ECMP multipath: Currently set at 2

neighbour       V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
2001::1       4   100     79     78       5    0    0 00:31:29       1
2003::3       4   100    169    171       5    0    0 00:32:31       1
2004::4       4   100     80     79       5    0    0 00:33:13       1

Total number of neighbours 3
iBOS#
```

Configure Multipath eBGP for IPv4

Figure 85 Multipath eBGP for IPv4



To configure Multipath eBGP for IPv4, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the BGP router.
`iBOS(config-router)#bgp router-id 2.2.2.2`	Configure a fixed Router ID (2.2.2.2).
`iBOS(config-router)#redistribute static`	Redistribute the static routes.
`iBOS(config-router)#neighbour 30.30.30.9 remote-as 100`	Configure neighbour 30.30.30.9 for iBGP.
`iBOS(config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS(config)#ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

2. Configure router R3:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the BGP router.
`iBOS(config-router)#bgp router-id 4.4.4.4`	Configure a fixed Router ID (4.4.4.4).
`iBOS(config-router)#redistribute static`	Redistribute the static routes.
`iBOS(config-router)#neighbour 40.40.40.9 remote-as 100`	Configure neighbour 40.40.40.9 for iBGP.

`iBOS(config-router)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS(config)#ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

3. Configure router R4:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the BGP router.
`iBOS(config-router)#bgp router-id 6.6.6.6`	Configure a fixed Router ID (6.6.6.6).
`iBOS(config-router)#redistribute static`	Redistribute the static routes.
`iBOS(config-router)#neighbour 50.50.50.9 remote-as 100`	Configure neighbour 50.50.50.9 for iBGP.
`iBOS(config)#exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS(config)#ip route 88.88.0.0/16 gigabitethernet3`	Configure the IP address on gigabitethernet3 interface.

4. Configure DUT on R2:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the BGP router.
`iBOS(config-router)#max-paths ebgp 2`	Configure eBGP max-paths (2).
`iBOS(config)#bgp router-id 9.9.9.9`	Configure a fixed Router ID (9.9.9.9).
`iBOS(config-router)#neighbour 30.30.30.2 remote-as 200`	Configure neighbour 30.30.30.2 for eBGP.
`iBOS(config-router)#neighbour 40.40.40.4 remote-as 200`	Configure neighbour 40.40.40.4 for eBGP.
`iBOS(config-router)#neighbour 50.50.50.6 remote-as 200`	Configure neighbour 50.50.50.6 for eBGP.
`iBOS(config-router)#exit`	Exit the Router mode and return to Configure mode.

To verify available multipaths, use the command **show ip bgp** on router R2:

```
iBOS#show ip bgp 88.88.0.0

BGP routing table entry for 88.88.0.0/16
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  30.30.30.2 50.50.50.6
  200
    30.30.30.2 from 30.30.30.2 (2.2.2.2)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate, installed
      Last update: Sat Jan  3 02:06:25 1970
```

```
  200
    50.50.50.6 from 50.50.50.6 (6.6.6.6)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate
      Last update: Sat Jan  3 02:05:39 1970

  200
    40.40.40.4 from 40.40.40.4 (4.4.4.4)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate, installed, best
      Last update: Sat Jan 3 02:05:11 1970
```

To verify routes installed, use the command **show ip route** on router R2:

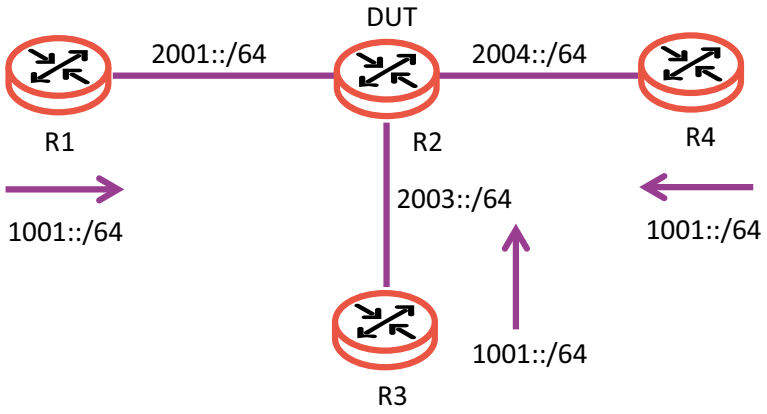
```
iBOS#show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default

C       30.30.30.0/24 is directly connected, ge3
C       40.40.40.0/24 is directly connected, ge5
C       50.50.50.0/24 is directly connected, ge7
B       88.88.0.0/16 [20/0] via 30.30.30.2, ge3, 00:00:11
                     [20/0] via 40.40.40.4, ge5, 00:00:11
C       90.90.90.90/32 is directly connected, lo
C       127.0.0.0/8 is directly connected, lo
iBOS#
```

To verify BGP configuration, use the command **show run router bgp** on router R2:

```
iBOS#show run router bgp
!
router bgp 100
 bgp router-id 9.9.9.9
 max-paths ebgp 2
 neighbour 30.30.30.2 remote-as 200
 neighbour 40.40.40.4 remote-as 200
 neighbour 50.50.50.6 remote-as 200
```

Configure Multipath eBGP for IPv6

Figure 86 Multipath eBGP for IPv6



To configure multipath eBGP for IPv6, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)# bgp router-id 1.1.1.1`	Configure a fixed Router ID (1.1.1.1).
`iBOS(config-router)# neighbour 2001::2 remote-as 200`	Configure neighbour 2001::2 for eBGP in AS: 200.
`iBOS(config-router)# address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)# redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)# neighbour 2001::2 activate`	Activate neighbour 2001::2.
`iBOS (config-router-af)# exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS (config-router)# exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS (config)# ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

2. Configure router R3:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)# bgp router-id 3.3.3.3`	Configure a fixed Router ID (3.3.3.3).
`iBOS(config-router)# neighbour 2003::2 remote-as 200`	Configure neighbour 2003::2 for eBGP in AS: 200.
`iBOS(config-router)# address-family ipv6`	Enter the IPv6 Address-Family mode.

`iBOS(config-router-af)# redistribute static`	Redistribute the IPv6 static routes.
`iBOS (config-router-af)# neighbour 2003::2 activate`	Activate neighbour 2003::2.
`iBOS (config-router-af)# exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS (config-router)# exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS (config)# ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

3. Configure router R4:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 100`	Assign the ASN value (100) to BGP router.
`iBOS(config-router)# bgp router-id 4.4.4.4`	Configure a fixed Router ID (4.4.4.4).
`iBOS(config-router)# neighbour 2004::2 remote-as 200`	Configure neighbour 2004::2 for eBGP in AS: 200.
`iBOS(config-router)# address-family ipv6`	Enter the IPv6 Address-Family mode.
`iBOS(config-router-af)# redistribute static`	Redistribute the IPv6 static routes.
`iBOS(config-router-af)# neighbour 2004::2 activate`	Activate neighbour 2004::2.
`iBOS(config-router-af)# exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS(config-router)# exit`	Exit the BGP Router mode and return to the Configure mode.
`iBOS(config)# ipv6 route 1001::/64 null`	Configure IPv6 static route with next-hop as NULL interface.

4. Configure DUT on router R2:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 200`	Assign the ASN value (200) to BGP router.
`iBOS(config-router)# bgp router-id 2.2.2.2`	Configure a fixed Router ID (2.2.2.2).
`iBOS (config-router)# max-paths ebgp 2`	Configure eBGP maximum paths (2).
`iBOS(config-router)# neighbour 2001::1 remote-as 100`	Configure neighbour 2001::1 for eBGP in AS: 100.
`iBOS(config-router)# neighbour 2003::3 remote-as 100`	Configure neighbour 2003::3 for eBGP in AS: 100.
`iBOS(config-router)# neighbour 2004::4 remote-as 100`	Configure neighbour 2004::4 for eBGP in AS: 100.
`iBOS(config-router)# address-family ipv6`	Enter the IPv6 Address-Family mode.

`iBOS(config-router-af)# neighbour 2001::1 activate`	Activate neighbour 2001::1.
`iBOS(config-router-af)# neighbour 2003::3 activate`	Activate neighbour 2003::3.
`iBOS(config-router-af)# neighbour 2004::4 activate`	Activate neighbour 2004::4.
`iBOS(config-router-af)# exit-address-family`	Exit the IPv6 Address-Family mode and return to the BGP Router mode.
`iBOS(config-router)# exit`	Exit the BGP Router mode and return to the Configure mode.

To verify available multiple paths, use the command **show bgp** on router R2:

```
iBOS#show bgp 1001::/64

BGP routing table entry for 1001::/64
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  2001::1 2003::3
  100
    2001::1(fe80::5054:ff:fe76:f9a2) from 2001::1 (1.1.1.1)
    (fe80::5054:ff:fe76:f9a2)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate
      Last update: Wed Oct 17 13:17:58 2012

  100
    2003::3(fe80::5054:ff:fe62:253a) from 2003::3 (3.3.3.3)
    (fe80::5054:ff:fe62:253a)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate, installed
      Last update: Wed Oct 17 13:17:50 2012

  100
    2004::4(fe80::5054:ff:feae:1f85) from 2004::4 (4.4.4.4)
    (fe80::5054:ff:feae:1f85)
      Origin incomplete metric 0, localpref 100, valid, external,
multipath-candidate, installed, best
      Last update: Wed Oct 17 13:16:57 2012
iBOS#
```

To verify routes installed, use the command **show ipv6 route** on router Rtr2:

```
iBOS#show ipv6 route
IPv6 Routing Table
Codes: K - kernel, C - connected, S - static, CL - client, O - OSPF,
       IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, I - IS-IS, B - BGP
Timers: Uptime

C     ::1/128 via ::, lo, 03:18:43
B     1001::/64 [20/0] via fe80::5054:ff:fe62:253a,
gigabitethernet3, 01:44:06
             [20/0] via fe80::5054:ff:feae:1f85, gigabitethernet4,
01:44:06
C     2001::/64 via ::, gigabitethernet1, 03:18:38
C     2003::/64 via ::, gigabitethernet3, 03:18:38
C     2004::/64 via ::, gigabitethernet4, 03:18:38
C     fe80::/64 via ::, gigabitethernet0, 03:18:43
K     ff00::/8 via ::, virbr0, 03:18:43
iBOS#
```

To list connected neighbours, use the command **show bgp summary** on router R2:

```
iBOS#show bgp summary
BGP router identifier 2.2.2.2, local AS number 200
BGP table version is 1
1 BGP AS-PATH entries
0 BGP community entries
2  Configured ebgp ECMP multipath: Currently set at 2
1  Configured ibgp ECMP multipath: Currently set at 1

neighbour       V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
2001::1       4  100    262    265       1   0    0 01:44:49      1
2003::3       4  100    282    284       1   0    0 01:45:20      1
2004::4       4  100    273    274       1   0    0 01:45:50      1

Total number of neighbours 3
iBOS#
```

BGP Graceful Shutdown

This section contains BGP graceful shutdown configuration examples. To see details about the commands used in these examples, or to view outputs of the validation commands, refer *iBOS Command Reference*.

BGP graceful shutdown reduces and avoid packets loss during maintenance shutdown by keeping the path undisturbed even after shutdown by lowering local preference so that routers of an AS can use this until they become aware of new path. If there is no alternate path available in the AS, the same path is used until forwarding failure occurs.

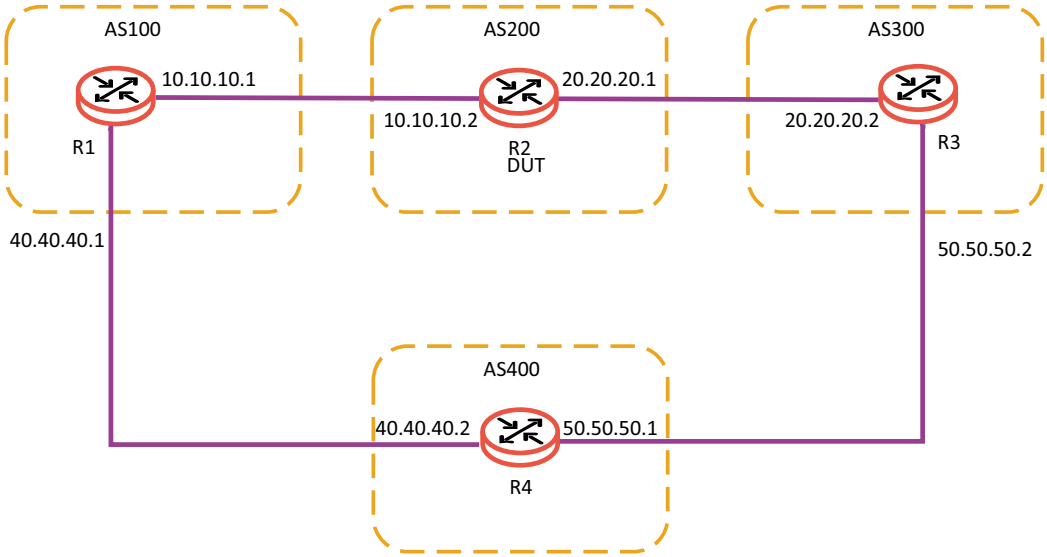
BGP graceful shutdown is supported for both BGP-IPv4 and BGP-IPv6 configurations.

Configure BGP Graceful Shutdown on Interface

Figure 87 shows the minimum configuration required to enable graceful shutdown on an interface. R1, R2, and R3 are three routers in three different ASs. R4 is another router in another AS forming an alternate path – R1 R4 R3 to the primary path.

BGP graceful shutdown is only supported for eBGP configurations. It is not supported for iBGP configurations.

Figure 87 BGP Graceful Shutdown Topology 1



To configure BGP Graceful Shutdown on an interface, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 100`	Assign the ASN value (100) to BGP router. The AS number should be a unique positive integer identifying the Autonomous System.
`iBOS(config-router)# neighbour 10.10.10.2 remote-as 200`	Establish eBGP neighbourship with R2.
`iBOS(config-router)# neighbour 40.40.40.2 remote-as 400`	Establish eBGP neighbourship with R4.

2. Configure router R2:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 200`	Assign the ASN value (200) to BGP router. T
`iBOS(config-router)# neighbour 10.10.10.1 remote-as 100`	Establish eBGP neighbourship with R1.
`iBOS(config-router)# neighbour 20.20.20.2 remote-as 300`	Establish eBGP neighbourship with R3.
`iBOS(config-router)# bgp g-shut-capable`	Enable the graceful shutdown capability on the DUT.
`iBOS(config-router)# neighbour 10.10.10.1 g-shut`	Enable graceful shutdown on the interface connected to R1.

`iBOS(config-router)# neighbour 10.10.10.1 g-shut-timer 70`	Configure the value of the graceful shutdown timer. After the timer expires, the BGP session initiated for graceful shutdown is shut down. The supported range for the graceful shutdown timer value is from 10 to 65535, in seconds. The default value is 60 seconds.
`iBOS(config-router)# network 6.6.6.0/24`	Specify the network to be advertised by the BGP routing process from DUT.

3. Configure router R3:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 300`	Assign the ASN value (300) to BGP router.
`iBOS(config-router)# neighbour 20.20.20.1 remote-as 200`	Establish eBGP neighbourship with R2
`iBOS(config-router)# neighbour 50.50.50.1 remote-as 400`	Establish eBGP neighbourship with R4.

4. Configure router R4:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value (400) to the BGP router.
`iBOSRtr4(config-router)# neighbour 40.40.40.1 remote-as 100`	Establish eBGP neighbourship with R1.
`iBOS(config-router)# neighbour 50.50.50.2 remote-as 300`	Establish eBGP neighbourship with R3.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp g-shut-capable**
- **neighbour g-shut-local-preference**
- **neighbour g-shut-timer**

To verify the BGP Graceful Shutdown Community Tag on the DUT interface, use the command **show ip bgp** on router R2:

```
iBOS# show ip bgp 6.6.6.0
BGP routing table entry for 6.6.6.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    0.0.0.0 from 0.0.0.0 (20.20.20.1)
     Origin IGP, localpref 100, weight 32768, valid, sourced, local,
best
Community: 0:65535 <<<<<<< community tagged (The community value
65535 is used for the graceful shutdown)
Last update: Wed Apr 18 15:37:18 2012
Checking for BGP Graceful Shutdown Community Tag on DUT
iBOS#
```

Configure BGP Graceful Shutdown on Router

Figure 87 shows the minimum configuration required to enable graceful shutdown at the router level. R1, R2, and R3 are three routers in three different ASs. Router R4 is another router in another AS forming an alternate path – R1, R4, R3 to the primary path.

To configure BGP Graceful Shutdown, do the following:

1. Configure router R1:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 100`	Assign the ASN value (100) to the BGP router. The AS number should be a unique positive integer identifying the Autonomous System.
`iBOS(config-router)#neighbour 10.10.10.2 remote-as 200`	Establish eBGP neighbourship with R2.
`iBOS(config-router)#neighbour 40.40.40.2 remote-as 400`	Establish eBGP neighbourship with R4.

2. Configure router R2:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 200`	Assign the ASN value (200) to the BGP router. The AS number should be a unique positive integer identifying the Autonomous System.
`iBOS(config-router)#neighbour 10.10.10.1 remote-as 100`	Establish eBGP neighbourship with R1.
`iBOS(config-router)#neighbour 20.20.20.2 remote-as 300`	Establish eBGP neighbourship with R3.
`iBOS(config-router)#bgp g-shut-capable`	Enabling the graceful shutdown capability on the DUT.
`iBOS(config-router)#bgp g-shut`	Enable BGP graceful shutdown functionality at the router level.
`iBOS(config-router)#bgp g-shut-local-preference 100`	Configure the local preference value of the routes to be used during graceful shutdown. This local reference is different from the BGP local preference value and is set in order to make the alternative path a preferred one in the case of graceful shutdown. The supported range of the local preference value is from 0 to 4294967295.
`iBOS(config-router)#network 70.70.70.0/24`	Specify the network to be advertised by the BGP routing process from DUT.

3. Configure router R3:

`iBOS#configure terminal`	Enter the Configure mode.
`iBOS(config)#router bgp 300`	Assign the ASN value (300) to the BGP router. The AS number should be a unique positive integer identifying the Autonomous System.
`iBOS(config-router)#neighbour 20.20.20.1 remote-as 200`	Establish eBGP neighbourship with R2.
`iBOS(config-router)#neighbour 50.50.50.1 remote-as 400`	Establish eBGP neighbourship with R4.

4. Configure router R4:

`iBOS# configure terminal`	Enter the Configure mode.
`iBOS(config)# router bgp 400`	Assign the ASN value (400) to the BGP router. The AS number should be a unique positive integer identifying the Autonomous System.
`iBOS(config-router)# neighbour 40.40.40.1 remote-as 100`	Establish eBGP neighbourship with R1.
`iBOS(config-router)# neighbour 50.50.50.2 remote-as 300`	Establish eBGP neighbourship with R3.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp g-shut-capable**
- **bgp g-shut**

To verify the configuration, use the command **show ip bgp** on router R2:

```
iBOS#show ip bgp 70.70.70.0
BGP routing table entry for 70.70.70.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  20.20.20.2
  Local
    0.0.0.0 from 0.0.0.0 (20.20.20.1)
     Origin IGP, localpref 100, weight 32768, valid, sourced, local,
bestBGP routing table entry for 70.70.70.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Community: 0:65535 <<<<<<< community tagged (The community value
65535 is used for the graceful shutdown)
Last update: Wed Apr 20 11:21:17 2012
iBOS#
```

Extended BGP

Extended BGP, commonly known as BGP4+, updates the original BGP-4 specification (RFC2858) with support for additional protocols, like IPv6, as defined by RFC2545.

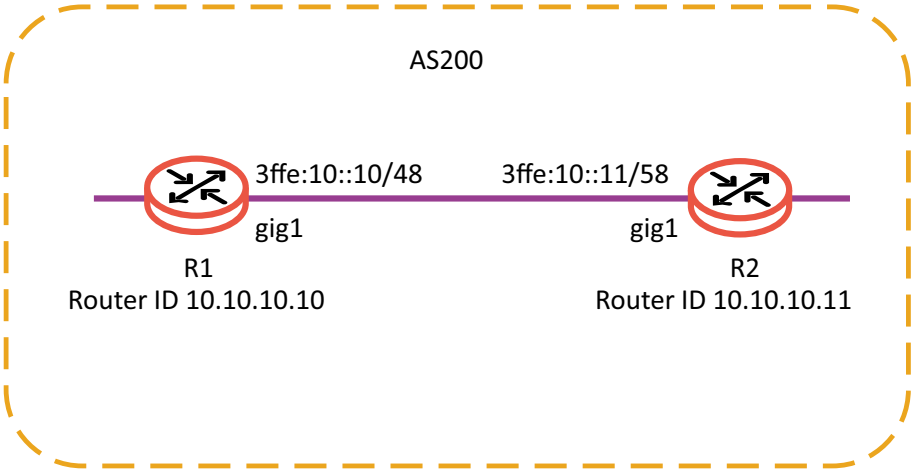
iBOS supports the following BGP4+ features:

- "Enable iBGP Peering Using Link-local Address" on page 332
- "Enable eBGP Peering Between Different Autonomous Systems" on page 334
- "Configure BGP4+ Route-Map" on page 335
- "BGP4+ Route Reflector" on page 338
- "BGP4+ Confederations" on page 340
- "Configure BGP4+ Graceful Restart" on page 342

Enable iBGP Peering Using Global Address

This example shows the minimum configuration required to enable BGP on an interface. R1 and R2 are two routers belonging to the same Autonomous System (AS), AS200, connecting to network 3ffe:10::/48.

Figure 88 iBGP Peering



First, specify the IPv6 global address, then define the routing process and AS number to which the routers belong. Configure a fixed Router ID, and then define BGP neighbours to start exchanging routing updates.

To configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# interface eth1`	Specify the interface (`eth1`) to configure.
`R1(config-if)# ipv6 address 3ffe:10::10/48`	Specify the IPv6 global address.
`R1(config-if)# exit`	Enter Configure mode.
`R1(config)# router bgp 200`	Define the routing process. The number `200` specifies the AS number of `R1`.
`R1(config-router)# bgp router-id 10.10.10.10`	Configure a fixed Router ID (`10.10.10.10`) for the BGP4+ routing process.
`R1(config-router)# neighbour 3ffe:10::11 remote-as 200`	Define BGP neighbour `R2`, and establish a TCP session by specifying the global IPv6 address (`3ffe:10::11`) and the AS number (`200`) of neighbour R2.
`R1(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R1(config-router-af)# neighbour 3ffe:10::11 activate`	Activate the neighbour `R2(3ffe:10::11)`, and enable exchange of IPv6 address prefix types with this neighbour.

To configure router R2:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# interface eth1`	Specify the interface (`eth1`) to configure.
`R2(config-if)# ipv6 address 3ffe:10::11/48`	Specify the IPv6 global address.
`R2(config-if)# exit`	Enter Configure mode.
`R2(config)# router bgp 200`	Define the routing process. The number `200` specifies the AS number of `R2`.
`R2(config-router)# bgp router-id 10.10.10.11`	Configure a fixed Router ID (`10.10.10.11`) for the BGP4+ routing process.
`R2(config-router)# neighbour 3ffe:10::10 remote-as 200`	Define the BGP neighbour (R1), and establish a TCP session by specifying the global IPv6 address (`3ffe:10::10`) and the AS number (`200`) of neighbour `R1`.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# neighbour 3ffe:10::10 activate`	Activate the neighbour `R1(3ffe:10::10)`, and enable exchange of IPv6 address prefix types with this neighbour.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

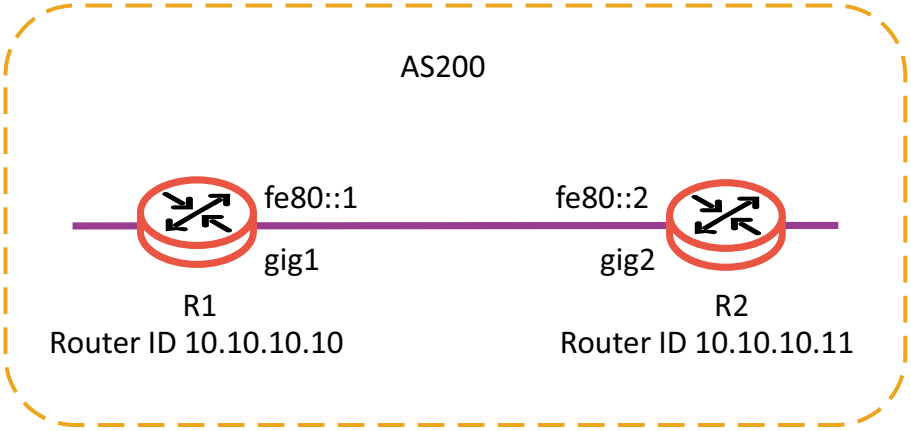
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**

Enable iBGP Peering Using Link-local Address

This example shows the minimum configuration required to enable iBGP on an interface.

Figure 89 iBGP Peering Link-Local Address



R1 and R2 are two routers belonging to the same AS, AS200, connecting to network fe80::/10. First, define the routing process and AS number to which the routers belong. Configure a fixed Router ID for the BGP4+ routing process, and then define BGP neighbours to start exchanging routing updates.

To configure the iBGP peering example, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# router bgp 200`	Define the routing process. The number `200` specifies the AS number of `R1`.
`R1(config-router)# bgp router-id 10.10.10.10`	Configure a fixed Router ID (`10.10.10.10`) for the BGP4+ routing process.
`R1(config-router)# neighbour fe80::2 remote-as 200`	Define BGP neighbour (R2), and establish a TCP session by specifying the link-local address (`fe80::2`) and the AS number (`200`) of neighbour `R2`.
`R1(config-router)# neighbour fe80::2 interface eth0`	To specify a link-local neighbour, configure the interface name of the neighbour `fe80::2`.
`R1(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R1(config-router-af)# neighbour fe80::2 activate`	Activate the neighbour `R2(fe80::2)`, and enable exchange of IPv6 address prefix types with this neighbour.

2. Configure router R2:

`R2(config)# router bgp 200`	Define the routing process. The number `200` specifies the AS number of `R2`.
`R2(config-router)# bgp router-id 10.10.10.11`	Configure a fixed Router ID (`10.10.10.11`) for the BGP4+ routing process.
`R2(config-router)# neighbour fe80::1 remote-as 200`	Define the BGP neighbour (R1), and establish a TCP session by specifying the link-local address `R1(fe80::1)` and the AS number (`200`) of neighbour `R1`.
`R2(config-router)# neighbour fe80::1 interface eth1`	To specify a link-local neighbour, configure the interface name of the neighbour `fe80::1`.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# neighbour fe80::1 activate`	Activate the neighbour `R1(fe80:1)`, and enable exchange of IPv6 address prefix types with this neighbour.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

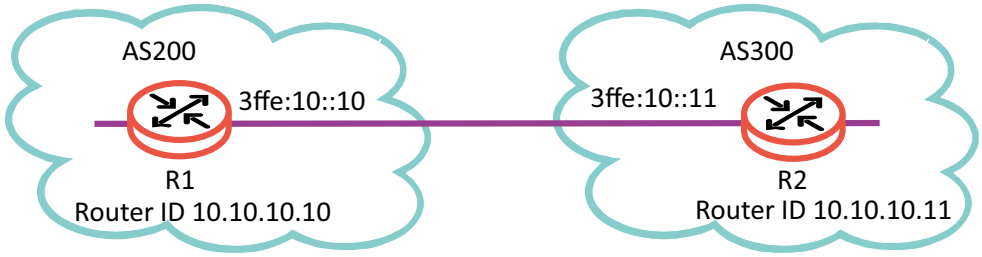
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**

Enable eBGP Peering Between Different Autonomous Systems

This example shows the minimum configuration required to enable eBGP on an interface, when the routers belong to different ASs.

Figure 90 BGP Peering - Different AS



R1 and R2 are two routers in different ASs, AS200 and AS300 connecting to network 3ffe:10::/64.

To configure eBGP peering in the example, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# router bgp 200`	Define the routing process. The number `200` specifies the AS number of `R1`.
`R1(config-router)# bgp router-id 10.10.10.10`	Configure a fixed Router ID (`10.10.10.10`) for the BGP4+ routing process.
`R1(config-router)# neighbour 3ffe:10::11 remote-as 300`	Define the BGP neighbour (R2), and establish a TCP session by specifying the IPv6 address (`3ffe:10::11`) and the AS number (`300`) of neighbour `R2`.
`R1(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R1(config-router-af)# neighbour 3ffe:10::11 activate`	Activate the neighbour `R2(3ffe:10::11)`, and enable exchange of IPv6 address prefix types with this neighbour.

2. Configure router R2:

`R2(config)# router bgp 300`	Define the routing process. The number `300` specifies the AS number of `R2`.
`R2(config-router)# bgp router-id 10.10.10.11`	Configure a fixed Router ID (`10.10.10.11`) for the BGP4+ routing process.
`R2(config-router)# neighbour 3ffe:10::10 remote-as 200`	Define the BGP neighbour (R1), and establish a TCP session by specifying the IPv6 address (`3ffe:10::10`) and the AS number (`200`) of neighbour `R1`.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# neighbour 3ffe:10::10 activate`	Activate the neighbour `R1(3ffe:10::10)` and enable exchange of IPv6 address prefix types with this neighbour.

To verify the configuration, use the following commands:

- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

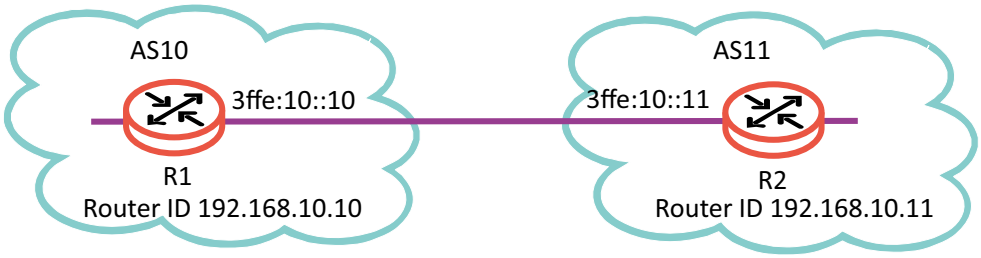
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**

Configure BGP4+ Route-Map

Route-maps are used to filter incoming updates from a BGP peer.

Figure 91 Configure Route-Maps (BGP4+)



In this example, a prefix-list ws on R1 is configured to deny entry of any routes with the IP address 3ffe:12::/32.

To test the route filter, R2 is configured to generate network prefixes 3ffe:11::/48 and 3ffe:12::/48.

To verify that the filter is working, the show ipv6 bgp command on R1 should show that R1 is only receiving the 3ffe:11::/48 network prefix.

To configure route map in the example, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# ipv6 prefix-list ws seq 5 deny 3ffe:12::/32 ge 48 le 64`	Create an entry in the prefix-list. `ws` is the name of the map that is created. `5` and `10` specify the sequence number or position of this specific route map. `deny` specifies the packets are to be rejected. `permit` specifies the packets are to be allowed. `48` and `64` are the minimum and maximum prefix lengths, respectively, to be matched.
`R1(config)# ipv6 prefix-list ws seq 10 permit any`	Create another entry in the `ws` map. `10` specifies the sequence number or position of this specific route map. `permit any` specifies accept all packets of any length.
`R1(config)# route-map ipi permit 1`	Enter Route-map mode.
`R1(config-route-map)# match ipv6 address prefix-list ws`	Set the match criteria. In this case, if the route-map name matches `ws`, the packets from the first sequence will be denied.
`R1(config-route-map)# exit`	Exit Route-map mode, and return to Configure mode.
`R1(config)# router bgp 10`	Define the routing process. The number `10` specifies the AS number of `R1`.
`R1(config-router)# bgp router-id 192.168.10.10`	Configure a fixed Router ID (`192.168.10.10`) for the BGP4+ routing process.
`R1(config-router)# neighbour 3ffe:10:11 remote-as 11`	Define the BGP neighbour (R2), and establish a TCP session by specifying the IPv6 address (`3ffe:10::11`) and the AS number (`11`) of neighbour `R2`.
`R1(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R1(config-router-af)# neighbour 3ffe:10::11 route-map ws in`	Apply the route-map `ws` to all incoming routes.
`R1(config-router-af)# neighbour 3ffe:10::11 activate`	Activate the neighbour `R2(3ffe:10::11)`, and enable exchange of IPv6 address prefix types with this neighbour.
`R1(config-router-af)# exit-address-family`	Exit Address Family mode, and return to Router mode.
`R1(config-router)# exit`	Exit Router mode, and return to Configure mode.

2. Configure router R2:

`R2(config)# router bgp 11`	Define the routing process. The number `11` specifies the AS number of `R2`.
`R2(config-router)# bgp router-id 192.168.10.11`	Configure a fixed Router ID (`192.168.10.11`) for the BGP4+ routing process.
`R2(config-router)# neighbour 3ffe:10::10 remote-as 10`	Define the BGP neighbour (R1), and establish a TCP session by specifying the IPv6 address (`3ffe:10::10`) and the AS number (`10`) of neighbour `R1`.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# network 3ffe:11::/48`	Announce the IPv6 network prefix (`3ffe:11::/48`).
`R2(config-router-af)# network 3ffe:12::/48`	Announce the IPv6 network prefix (`3ffe:12::/48`).

`R2(config-router-af)# neighbour 3ffe:10::10 activate`	Activate the neighbour `R1(3ffe:10::10)`, and enable exchange of IPv6 address prefix types with this neighbour.
`R2(config-router-af)# exit-address-family`	Exit Address Family mode, and return to `Router` mode.

To verify the configuration, use the following commands:

- **neighbour route-map**
- **ipv6 prefix-list**
- **neighbour remote-as**
- **route-map**
- **match ipv6 address**
- **network**
- **network backdoor**
- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

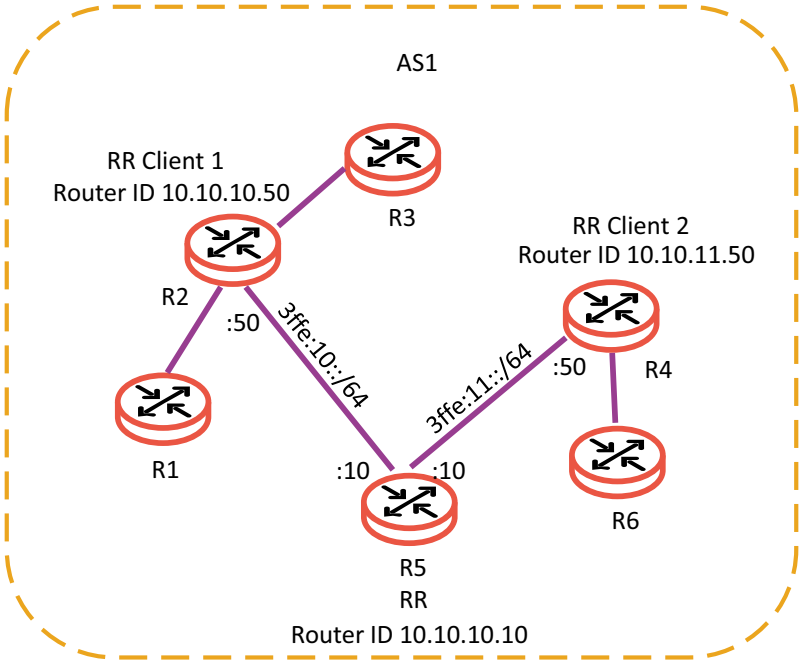
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**
- **show ipv6 bgp**
- **show ipv6 bgp prefix-list**

BGP4+ Route Reflector

Route Reflectors are used to reduce the iBGP mesh inside an AS.

Figure 92 BGP4+ Route Reflector



In this example, R2, R5, and R4 would have to maintain a full mesh among themselves, but by making R5 the Route Reflector, R2 (Client1) has an iBGP session with RR only, and not with R4 (Client 2). The routes learned from R2 are advertised to the other clients, and to iBGP peers outside the cluster; the iBGP routes learned from iBGP peers outside the cluster are advertised to the R2. This reduces the iBGP peer connections in AS1.

To configure Route Reflector in the example, do the following:

1. Configure route reflector RR (router R5):

`R5# configure terminal`	Enter Configure mode.
`R5(config)# router bgp 1`	Define the routing process. The number `1` specifies the AS number of R5 (RR).
`R5(config-router)# bgp router-id 10.10.10.10`	Configure a fixed Router ID (`10.10.10.10`) for the BGP4+ routing process.
`R5(config-router)# neighbour 3ffe:10::50 remote-as 1`	Define the BGP neighbour (R2), and establish a TCP session by specifying the IPv6 address (`3ffe:10::50`) and the AS number (`1`) of neighbour R2.
`R5(config-router)# neighbour 3ffe:11::50 remote-as 1`	Define the BGP neighbour (R4), and establish a TCP session by specifying the IPv6 address (`3ffe:11::50`) and the AS number (`1`) of neighbour R4.
`R5(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R5(config-router-af)# neighbour 3ffe:10::50 activate`	Activate the neighbour R2 (`3ffe:10::50`), and enable exchange of IPv6 address prefix types with this neighbour.

`R5(config-router-af)# neighbour 3ffe:10::50 route-reflector-client`	Configure R5 as the Route-Reflector (RR) and neighbour R2 as its client.
`R5(config-router-af)# neighbour 3ffe:11::50 activate`	Activate the neighbour R4(3ffe:11::50), and enable exchange of IPv6 address prefix types with this neighbour.
`R5(config-router-af)# neighbour 3ffe:11::50 route-reflector-client`	Configure R5 as the Route-Reflector (RR) and neighbour R4 as its client.

2. Configure RR Client 1 (R2):

`R2(config)# router bgp 1`	Define the routing process. The number 1 specifies the AS number of R2 (RR Client 1).
`R2(config-router)# bgp router-id 10.10.10.50`	Configure a fixed Router ID (10.10.10.50) for the BGP4+ routing process.
`R2(config-router)# neighbour 3ffe:10::10 remote-as 1`	Define the BGP neighbour (R5), and establish a TCP session by specifying the IPv6 address (3ffe:10::10) and the AS number(1) of neighbour R5.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# neighbour 3ffe:10::10 activate`	Activate the neighbour (3ffe:10::10), and enable exchange of IPv6 address prefix types with this neighbour.

7. Configure RR Client 2 (R4):

`R4(config)# router bgp 1`	Define the routing process. The number 1 specifies the AS number of R4 (RR Client 2).
`R4(config-router)# bgp router-id 10.10.11.50`	Configure a fixed Router ID (10.10.11.50) for the BGP4+ routing process.
`R4(config-router)# neighbour 3ffe:11::10 remote-as 1`	Define the BGP neighbour (R5), and establish a TCP session by specifying the IPv6 address (3ffe:11::10) and the AS number(1) of the neighbour.
`R4(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R4(config-router-af)# neighbour 3ffe:11::10 activate`	Activate the neighbour (3ffe:11::10), and enable exchange of IPv6 address prefix types with this neighbour.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **neighbour route-reflector-client**
- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

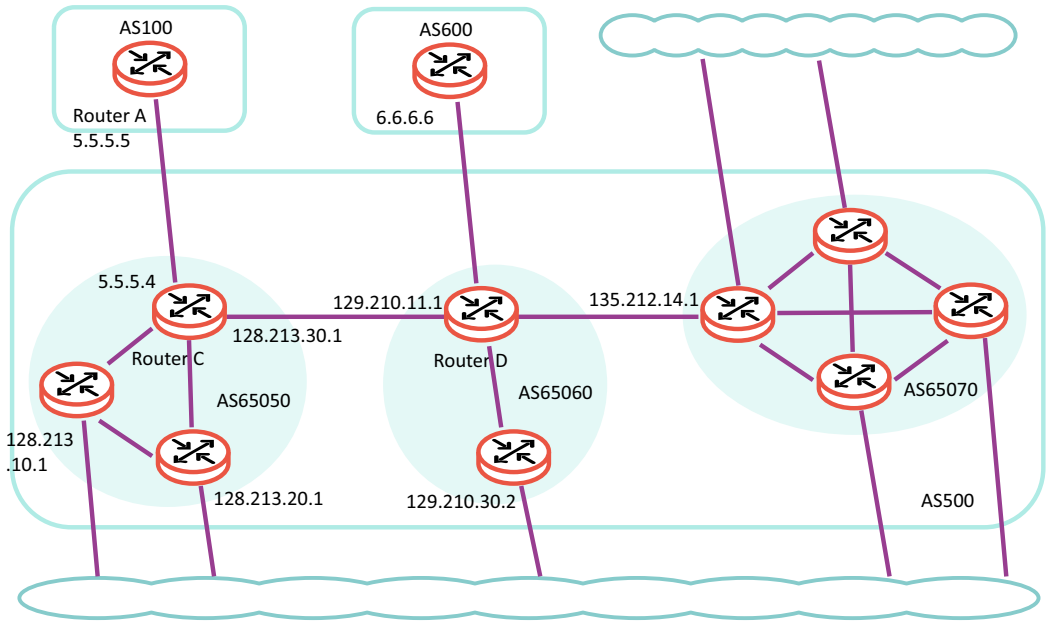
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**

BGP4+ Confederations

In this example, AS1 contains three Confederated Autonomous Systems--AS 1000, AS 1001 and AS 1002. To any outside AS, the overall Confederation is a single AS, AS1. Confederation eBGP is run between R2 and R5, and between R5 and R7. R2 is configured so that its local AS is 1000, and the peer connection to R5 is set up like any other eBGP session. The bgp confederation identifier command informs the router that it is a member of a Confederation and passes the Confederation ID. The bgp confederation peers command lists the member AS to which R2 is connected. This command also tells the BGP process that the eBGP connection is a Confederation eBGP, rather than a normal eBGP.

Figure 93 BGP4+ Confederations



To configure BGP4+ Confederation in the example, do the following:

1. Configure router R2:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# router bgp 1000`	Define the routing process. The number 1000 specifies the AS number of R2.
`R2(config-router)# bgp router-id 10.10.10.2`	Configure a fixed Router ID (10.10.10.2) for the BGP4+ routing process.
`R2(config-router)# bgp confederation identifier 1`	Specify the BGP Confederation Identifier (1). To others, the group will appear as a single AS, and the identifier as its AS number.
`R2(config-router)# bgp confederation peers 1001 1002`	Specify ASs 1001 and 1002 as confederation peers, making them members of the Confederation.
`R2(config-router)# neighbour 3ffe:10::5 remote-as 1001`	Define the BGP neighbour (R5), and establish a TCP session by specifying the IPv6 address (3ffe:10::5) and the AS number(1001)of neighbour R5.
`R2(config-router)# neighbour 3ffe:9::1 remote-as 1000`	Define the BGP neighbour (R1), and establish a TCP session by specifying the IPv6 address (3ffe:9::1) and the AS number(1000)of neighbour R1.

`R2(config-router)# neighbour 3ffe:7::3 remote-as 1000`	Define the BGP neighbour (R3), and establish a TCP session by specifying the IPv6 address (3ffe:7::3)and the AS number(1000)of neighbour R3.
`R2(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R2(config-router-af)# neighbour 3ffe:10::5 activate`	Activate neighbour R5(3ffe:10::5), and enable exchange of IPv6 address prefix types with this neighbour.
`R2(config-router-af)# neighbour 3ffe:9::1 activate`	Activate neighbour R1(3ffe:9::1), and enable exchange of IPv6 address prefix types with this neighbour.
`R2(config-router-af)# neighbour 3ffe:7::3 activate`	Activate neighbour R3(3ffe:7::3), and enable exchange of IPv6 address prefix types with this neighbour.

2. Configure router R5:

`R5(config)# router bgp 1001`	Define the routing process. The number 1001 specifies the AS number of R5.
`R5(config-router)# bgp router-id 10.10.10.5`	Configure a fixed Router ID (10.10.10.5) for the BGP4+ routing process.
`R5(config-router)# bgp confederation identifier 1`	Specify the BGP Confederation Identifier (1). To others, the group will appear as a single AS, and the identifier as its AS number.
`R5(config-router)# bgp confederation peers 1000 1002`	Specify ASs 1001 and 1002 as confederation peers, making them members of the Confederation.
`R5(config-router)# neighbour 3ffe:10::2 remote-as 1000`	Define the BGP neighbour (R2), and establish a TCP session by specifying the IPv6 address (3ffe:10::2)and the AS number(1000)of neighbour R2.
`R5(config-router)# neighbour 3ffe:11::7 remote-as 1002`	Define the BGP neighbour (R7), and establish a TCP session by specifying the IPv6 address (3ffe:11::7)and the AS number(1002)of neighbour R7.
`R5(config-router)# neighbour 3ffe:12::4 remote-as 1001`	Define the BGP neighbour (R4), and establish a TCP session by specifying the IPv6 address (3ffe:12::4)and the AS number(1001)of neighbour R4.
`R5(config-router)# neighbour 3ffe:13::6 remote-as 1001`	Define the BGP neighbour (R6), and establish a TCP session by specifying the IPv6 address (3ffe:13::6)and the AS number(1001)of neighbour R6.
`R5(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R5(config-router-af)# neighbour 3ffe:10::2 activate`	Activate the neighbour R2(3ffe:10::2), and enable exchange of IPv6 address prefix types with this neighbour.
`R5(config-router-af)# neighbour 3ffe:11::7 activate`	Activate the neighbour R7(3ffe:11::7), and enable exchange of IPv6 address prefix types with this neighbour.
`R5(config-router-af)# neighbour 3ffe:12::4 activate`	Activate the neighbour R4(3ffe:12::4), and enable exchange of IPv6 address prefix types with this neighbour.
`R5(config-router-af)# neighbour 3ffe:13::6 activate`	Activate the neighbour R6(3ffe:13::6), and enable exchange of IPv6 address prefix types with this neighbour.

3. Configure router R7:

`R7(config)# router bgp 1002`	Define the routing process. The number 1002 specifies the AS number of R7.
`R7(config-router)# bgp router-id 10.10.11.7`	Configure a fixed Router ID (10.10.11.7) for the BGP4+ routing process.
`R7(config-router)# bgp confederation identifier 1`	Specify BGP Confederation Identifier (1). To others, the group will appear as a single AS, and the identifier as its AS number.
`R7(config-router)# bgp confederation peers 1000 1001`	Specify ASs 1000 and 1001 as confederation peers, making them members of the Confederation.
`R7(config-router)# neighbour 3ffe:11::5 remote-as 1001`	Define the BGP neighbour (R5), and establish a TCP session by specifying the IPv6 address (3ffe:11::5)and the AS number(1001)of neighbour R5.
`R7(config-router)# neighbour 3ffe:15::8 remote-as 1002`	Define the BGP neighbour (R8), and establish a TCP session by specifying the IPv6 address (3ffe:15::8)and the AS number(1002)of neighbour R8.
`R7(config-router)# address-family ipv6`	Enter Address Family mode for configuring routing sessions that use IPv6 address prefixes.
`R7(config-router-af)# neighbour 3ffe:11::5 activate`	Activate the neighbour R5(3ffe:11::5), and enable exchange of IPv6 address prefix types with this neighbour.
`R7(config-router-af)# neighbour 3ffe:15::8 activate`	Activate the neighbour R8(3ffe:15::8), and enable exchange of IPv6 address prefix types with this neighbour.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **bgp confederation identifier**
- **bgp confederation peer**
- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family**
- **neighbour activate**

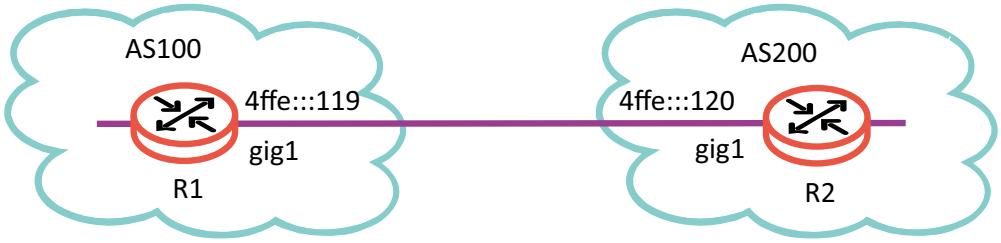
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**

Configure BGP4+ Graceful Restart

Using BGP4+ graceful restart, the data-forwarding plane of a router can continue to process and forward packets, even if the control plane (which is responsible for determining best paths) fails.

Figure 94 BGP4+ Graceful Restart



To configure BGP4+ Graceful Restart in the example, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# router bgp 100`	Assign the ASN value (100) to the router.
`R1(config-router)# bgp graceful-restart`	Enable BGP graceful restart support.
`R1(config-router)# neighbour 4ffe::120 remote-as 200`	Specify the neighbour's IP address (4ffe::120) and the ASN value of the neighbour (200).
`R1(config-router)# address-family ipv6 unicast`	Exchange the IPv6 capabilities, and switch the mode to the IPv6 address family.
`R1(config-router-af)# neighbour 4ffe::120 activate`	Specify the neighbour's IPv6 address(4ffe::120), and activate the neighbour.
`R1(config-router-af)# neighbour 4ffe::120 capability graceful-restart`	Specify the neighbour's IPv6 address (4ffe::120) for which the graceful restart capability is supported.

2. Configure router R2:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# router bgp 200`	Assign the ASN value (200) to the router.
`R2(config-router)# bgp graceful-restart`	Enable BGP graceful restart support.
`R2(config-router)# bgp graceful-restart restart-time 120`	Configure the maximum time (120) required for neighbour(s) to restart.
`R2(config-router)# bgp graceful-restart stalepath-time 120`	Configure the maximum time (120) to retain stale paths from the restarting neighbour(s).
`R2(config-router)# neighbour 4ffe::119 remote-as 100`	Specify the neighbour's IP address (4ffe::119) and the ASN value of the neighbour (100).
`R2(config-router)# address-family ipv6 unicast`	Exchange the IPv6 capabilities, and switch the mode to the IPv6 address family.

`R2# configure terminal`	Enter Configure mode.
`R2(config-router-af)# neighbour 4ffe::119 activate`	Specify the neighbour's IPv6 address(4ffe::119), and activate the neighbour.
`R2(config-router-af)# neighbour 4ffe::119 capability graceful-restart`	Specify the neighbour's IPv6 address (4ffe::119) for which the graceful restart capability is supported.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router bgp**
- **bgp graceful-restart**
- **bgp graceful-restart restart-time**
- **bgp graceful-restart stalepath-time**
- **neighbour remote-as**
- **neighbour capability graceful-restart**
- **address-family ipv6 unicast**
- **neighbour activate**

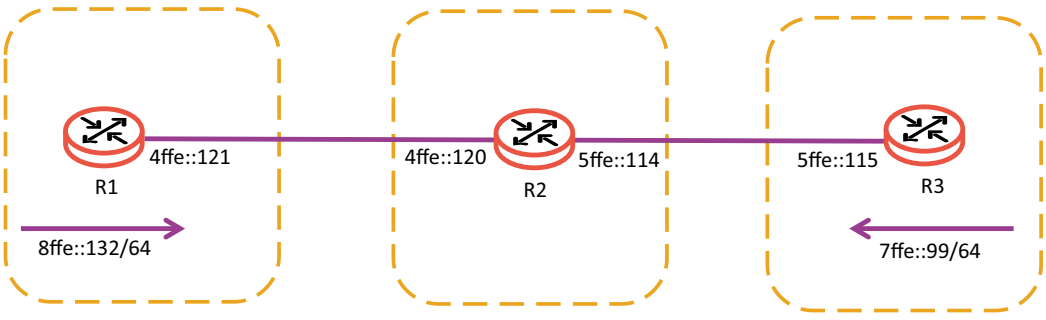
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**
- **show bgp ipv6**
- **show ipv6 route database bgp**
- **show ipv6 route database**
- **show ipv6 route**

Configure BGP4+ Distance

Administrative distance in BGP+ can be configured for a specific address family. The following example shows how to configure the BGP administrative distance for the IPv6 address family.

Figure 95 BGP4+ Distance



To configure BGP4+ distance, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# router bgp 100`	Assign the ASN value (100) to the router.
`R1(config-router)# network 8ffe::132/64`	Specify the network to be advertised by the BGP routing process.
`R1(config-router)# neighbour 4ffe::120 remote-as 100`	Specify the neighbour's IP address and ASN value.
`R1(config-router)# address-family ipv6 unicast`	Enter IPv6 Address-Family mode.
`R1(config-router-af)# neighbour 4ffe::120 activate`	Activate the IPv6 neighbour.

2. Configure router R2:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# router bgp 100`	Assign the ASN value (100) to the router.
`R2(config-router)# neighbour 4ffe::121 remote-as 100`	Specify the neighbour's IP address and ASN value.
`R2(config-router)# neighbour 5ffe::115 remote-as 200`	Specify the neighbour's IP address and the ASN value of another neighbour.
`R2(config-router)# address-family ipv6 unicast`	Enter IPv6 Address-Family mode.
`R2(config-router-af)# aggregate-address 2ffe::102/64 summary-only`	Configure an IPv6 non-AS-set aggregate route on R2. The local distance will be applied to this route.
`R2(config-router-af)# neighbour 4ffe::121 activate`	Activate the IPv6 neighbour.
`R2(config-router-af)# neighbour 5ffe::115 activate`	Activate the IPv6 neighbour.
`R2(config-router-af)# distance 12 13 11`	Configure the administrative distance for external, internal, and local routes received in IPv6 Address-Family mode.

3. Configure router R3:

`R3# configure terminal`	Enter Configure mode.
`R3(config)# router bgp 100`	Assign the ASN value (100) to the router.
`R3(config-router)# network 7ffe::99/64`	Specify the network to be advertised by the BGP routing process.
`R3(config-router)# neighbour 5ffe::114 remote-as 100`	Specify the neighbour's IP address and ASN value.

`R3(config-router)# address-family ipv6 unicast`	Enter IPv6 Address-Family mode.
`R3(config-router-af)# neighbour 5ffe::114 activate`	Activate the IPv6 neighbour.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **aggregate-address summary-only**
- **distance**
- **router bgp**
- **bgp router-id**
- **neighbour remote-as**
- **address-family ipv6 unicast**
- **neighbour activate**
- **network**

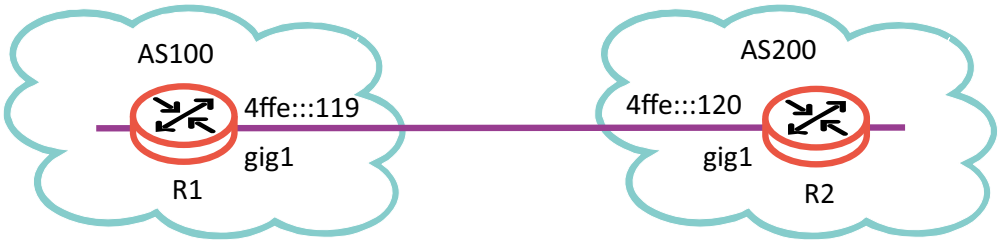
To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**
- **show ipv6 route database bgp**
- **show ipv6 route database**
- **show ipv6 route**
- **show bgp ipv6**

Configure BGP4+ Graceful Reset

The graceful reset feature for BGP4+ sessions is used to prevent changes in network configuration from disrupting packet forwarding. The **bgp graceful-restart graceful-reset** command invokes graceful restart when a configuration change forces a peer reset. Graceful restart can only be invoked when such changes force a peer reset.

Figure 96 BGP4+ Graceful Reset



To configure BGP4+ Graceful Reset, do the following:

1. Configure router R1:

`R1# configure terminal`	Enter Configure mode.
`R1(config)# router bgp 100`	Assign the ASN value (100) to the router.
`R1(config-router)# bgp graceful-restart`	Enable BGP graceful restart support.
`R1(config-router)# bgp graceful-restart graceful-reset`	Configure to invoke graceful restart when a configuration change forces a peer reset.
`R1(config-router)# neighbour 4ffe::120 remote-as 200`	Specify the neighbour's IP address (4ffe::120) and the ASN value of the neighbour (200).
`R1(config-router)# address-family ipv6 unicast`	Exchange the IPv6 capabilities, and switch the mode to the IPv6 address family.
`R1(config-router-af)# neighbour 4ffe::120 activate`	Specify the neighbour's IPv6 address(4ffe::120), and activate the neighbour.
`R1(config-router-af)# neighbour 4ffe::120 capability graceful-restart`	Specify the neighbour's IPv6 address (4ffe::120) for which the graceful restart capability is supported.

2. Configure router R2:

`R2# configure terminal`	Enter Configure mode.
`R2(config)# router bgp 200`	Assign the ASN value (200) to the router.
`R2(config-router)# bgp graceful-restart`	Enable BGP graceful restart support.
`R2(config-router)# bgp graceful-restart graceful-reset`	Configure to invoke graceful restart when a configuration change forces a peer reset.
`R2(config-router)# neighbour 4ffe::119 remote-as 100`	Specify the neighbour's IP address (4ffe::119) and the ASN value of the neighbour (100).
`R2(config-router)# address-family ipv6 unicast`	Exchange the IPv6 capabilities, and switch the mode to the IPv6 address family.
`ZebOS(config-router-af)#neighbour 4ffe::119 activate`	Specify the neighbour's IPv6 address(4ffe::119), and activate the neighbour.
`ZebOS(config-router-af)#neighbour 4ffe::119 capability graceful-restart`	Specify the neighbour's IPv6 address (4ffe::119) for which the graceful restart capability is supported.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **bgp graceful-restart**
- **bgp graceful-restart graceful-reset**
- **router bgp**
- **neighbour capability graceful-restart**
- **neighbour remote-as**
- **address-family ipv6 unicast**
- **neighbour activate**
- **network**

To verify the configuration, use the following commands:

- **show ipv6 bgp summary**
- **show ipv6 bgp neighbours**
- **show ipv6 route database**
- **show ipv6 route**
- **show ipv6 route**
- **show bgp ipv6**

Configuring VRF

Virtual Routing and Forwarding (VRF) is a technology for Layer 3 networks that allows multiple instances of a routing table to co-exist and run simultaneously on a single physical device.

VRF segregates network paths without using multiple devices. As a result, the VRF separation of traffic also increases network security and can eliminate the need for encryption and authentication.

Internet service providers (ISPs) often take advantage of VRF to create separate virtual private networks (VPNs) for customers; thus the technology is often referred to as VPN routing and forwarding.

VRF acts like a logical router, but while a logical router may include many routing tables, a VRF instance uses only a single routing table. In addition, VRF requires a forwarding table that designates the next hop for each data packet, a list of devices that may be called upon to forward the packet, and a set of rules and routing protocols that govern how the packet is forwarded. These tables prevent traffic from being forwarded outside a specific VRF path and also keep out traffic that should remain outside the VRF path.

VRF configuration in iBOS is described in the following sections:

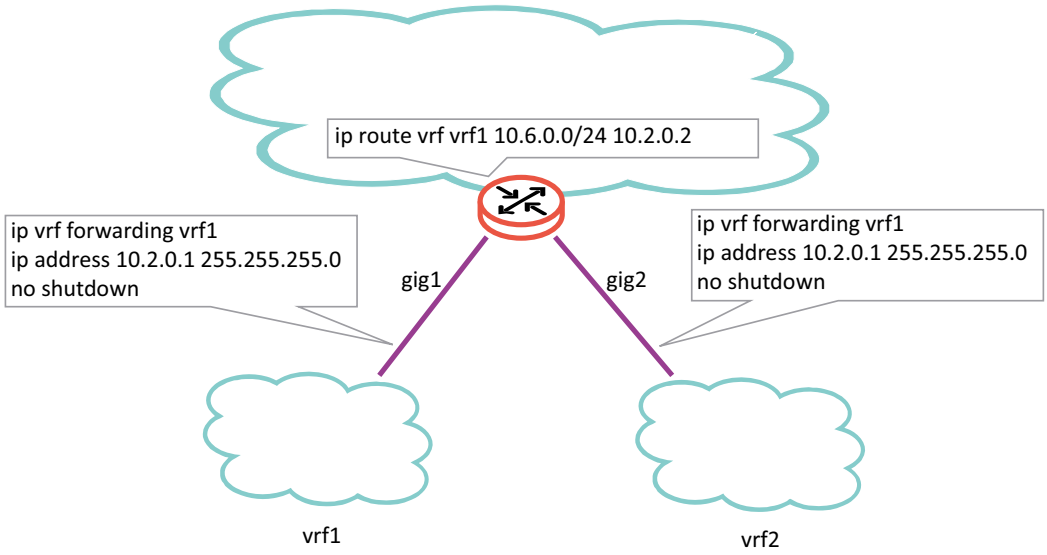
- "Simple VRF Implementation" on page 350
- "Full VRF Implementation" on page 354

Simple VRF Implementation

The simplest form of VRF implementation is commonly called VRF Lite. In this implementation, each router within the network participates in the virtual routing environment in a peer-based fashion. While simple to deploy and appropriate for small-to-medium enterprises and shared data centres, VRF Lite does not scale to the size required by global enterprises or large carriers because each VRF instance has to be implemented on every router, even intermediate routers.

VRFs were initially introduced in combination with MPLS but on local installations, it was easier to use VRF without MPLS. This is the historical explanation of the term VRF Lite: VRFs without MPLS.

Figure 97 Simple VRF configuration



First we must create the VRF and configure it on each interface:

```
ibos(config)# ip vrf vrf1

ibos(config)# interface gigabitethernet1
ibos(config)# ip vrf forwarding vrf1
ibos(config-vrf)# ip address 10.1.0.1 255.255.255.0
ibos(config-vrf)# no shutdown

ibos(config)# interface gigabitethernet2
ibos(config)# ip vrf forwarding vrf1
ibos(config-vrf)# ip address 10.2.0.1 255.255.255.0
ibos(config-vrf)# no shutdown
```

Now vrf1 is bound to the interfaces, we configure a static route for the VRF:

```
ibos(config)#ip route vrf vrf1 10.6.0.0/24 10.2.0.2
```

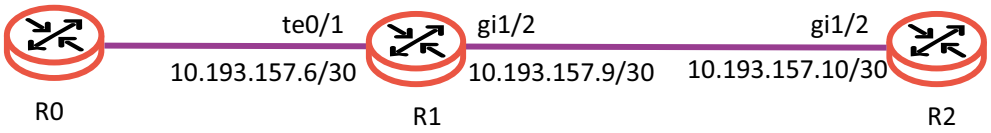
To set up a VRF configuration with OSPF, do the following:

- Bind interfaces to the VRF and enable OSPF
- Configure OSPF on the VRF

VRF Configuration Example

In this example, the VRF is configured and uses OSPF for forwarding.

Figure 98 VRF with OSPF



1. To bind the interface to a VRF and enable OSPF, configure R1 and R2.

Table 38 Configure VRF on R1

iBOS Command	Description

`R1(config)# ip vrf TEST1`
`R1(config-ip-vrf)# interface loopback0`
`R1(config-if-lo)# ip vrf forwarding TEST1`
`R1(config-if-lo)# ip address 10.193.157.0/32`
`R1(config-if-lo)# no shutdown`
	Create VRF

`R1(config-if-lo)# interface gigabitethernet1/2`
`R1(config-if-gi)# ip vrf forwarding TEST1`
`R1(config-if-gi)# ip address 10.193.157.9/30`
`R1(config-if-gi)# ip ospf hello-interval 1`
`R1(config-if-gi)# ip ospf dead-interval 4`
`R1(config-if-gi)# no shutdown`
	Bind interface gigabitethernet1/2 and enable OSPF

`R1(config-if-gi)# interface tengigabitethernet0/1`
`R1(config-if-te)# ip address 10.193.157.6/30`
`R1(config-if-te)# ip vrf forwarding TEST1`
`R1(config-if-te)# ip ospf hello-interval 1`
`R1(config-if-te)# ip ospf dead-interval 4`
`R1(config-if-te)# no shutdown`
	Bind uplink interface tengigabitethernet0/1 and enable OSPF

Table 39 Configure VRF on R2

iBOS Command	Description
R2(config)# ip vrf TEST2 R2(config-ip-vrf)# interface loopback0 R2(config-if-lo)# ip vrf forwarding TEST2 R2(config-if-lo)# ip address 10.193.157.1/32 R2(config-if-lo)# no shutdown	Create VRF
R2(config-if-lo)# interface gigabitethernet1/2 R2(config-if-gi)# ip vrf forwarding TEST2 R2(config-if-gi)# ip address 10.193.157.10/30 R2(config-if-gi)# ip ospf hello-interval 1 R2(config-if-gi)# ip ospf dead-interval 4 R2(config-if-gi)# no shutdown	Bind interface gigabitethernet1/2 and enable OSPF

2. Configure OSPF on VRFs for routers R1 and R2

Table 40 Configure OSPF for VRF on R1

iBOS command	Description
R1(config-if-te)# router ospf 1 TEST1 R1(config-router-ospf)# network 10.193.157.0 32 area 10.193.157.0 R1(config-router-ospf)# network 10.193.157.4 30 area 10.193.157.0 R1(config-router-ospf)# network 10.193.157.8 30 area 10.193.157.0 R1(config-router-ospf)# end	Enable OSPF on VRF

Table 41 Configure OSPF for VRF on R2

iBOS command	Description
R2(config-if-gi)# router ospf 2 TEST2 R2(config-router-ospf)# network 10.193.157.1 32 area 10.193.157.0 R1(config-router-ospf)# network 10.193.157.8 30 area 10.193.157.0 R1(config-router-ospf)# end	Enable OSPF on VRF

To check the OSPF neighbour and status:

```
R1# show ip ospf 1 neighbour

OSPF process 1:
neighbour ID     Pri   State           Dead Time   Address
Interface
-----------------------------------------------------------------------
--------------
10.193.157.5      1   Full/DR         00:00:04    10.193.157.5
```

```
tengigabitethernet0/1
10.193.157.1      1   Full/DR          00:00:04    10.193.157.10
gigabitethernet1/2
R1#
```

To check the interface for OSPF process 1 and the VRF TEST1:

```
R1# show ip ospf interface gigabitethernet1/2
gigabitethernet1/2 is up, line protocol is up
  Internet Address 10.193.157.9/30, Area 10.193.157.0, MTU 1500
  Process ID 1, VRF TEST1 (1)
  Router ID 10.193.157.0, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.193.157.0, Interface Address
10.193.157.9
  Backup Designated Router (ID) 10.193.157.1, Interface Address
10.193.157.10
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
    Hello due in 00:00:01
  neighbour Count is 1, Adjacent neighbour count is 1
  Crypt Sequence Number is 145
  Hello received 256 sent 257, DD received 6 sent 8
  LS-Req received 2 sent 1, LS-Upd received 5 sent 17
  LS-Ack received 14 sent 2, Discarded 0
R1#
```

To check the OSPF neighbour and status:

```
R1# show ip ospf 1 neighbour

OSPF process 1:
neighbour ID     Pri   State           Dead Time   Address
Interface
-----------------------------------------------------------------
--------------
10.193.157.5      1   Full/DR          00:00:04    10.193.157.5
tengigabitethernet0/1
10.193.157.1      1   Full/DR          00:00:04    10.193.157.10
gigabitethernet1/2
R1#

R2# show ip ospf 2 neighbour

OSPF process 2:
neighbour ID     Pri   State           Dead Time   Address
Interface
-----------------------------------------------------------------
--------------
10.193.157.0      1   Full/Backup      00:00:04    10.193.157.9
gigabitethernet1/2
R2#
```

Full VRF Implementation

The scaling limitations of VRF Lite are resolved by the implementation of full VRF, also known as IP-VPNs.

In this implementation, a core backbone network is responsible for the transmission of data across the wide area between VRF instances at each edge location. IP-VPNs have been traditionally deployed by carriers to provide a shared wide-area backbone network for multiple customers. They are also appropriate in large enterprises, multi-tenant and shared data centre environments.

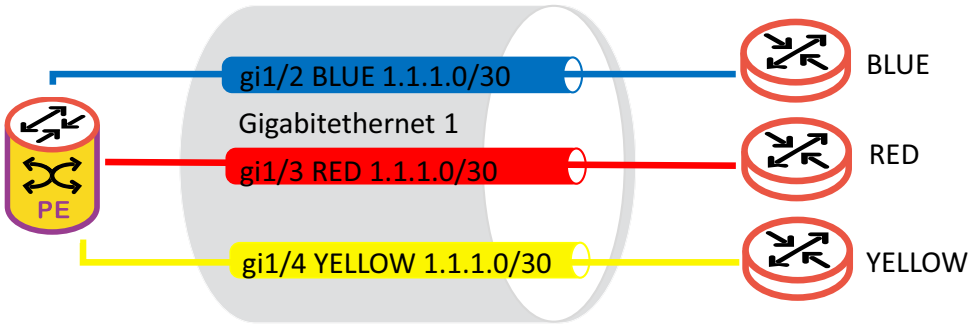
IP-VPN is most commonly deployed across a Multi Protocol Label Switching (MPLS) backbone as the inherent labeling of packets in MPLS lends itself to the identification of the customer VRF. See “LDP Configuration Examples” on page 213 for more details how VRF is used in combination with MPLS.

Other IP-VPN implementations utilize a simpler IP-in-IP encapsulation over a pure IP backbone, eliminating the need to maintain and support an MPLS environment.

In a typical deployment, Customer Edge (CE) routers handle local routing in a traditional fashion and distribute routing information to a Provider Edge (PE) router where the routing tables are virtualized. The PE router then encapsulates the traffic, marks it to identify the VRF instance, and transmits it across the provider backbone network to the destination PE router. The destination PE router then decapsulates the traffic and forwards it to the CE router at the destination. The backbone network is completely transparent to the customer equipment, allowing multiple customers or user communities to utilize the common backbone network while maintaining end-to-end traffic separation.

Routes across the provider backbone network are maintained using an Interior Gateway Protocol - typically IBGP. IBGP uses extended community attributes in a common routing table to differentiate the customers' routes with overlapping IP addresses.

Figure 99 Full VRF Implementation



First the VRFs are configured on the PE router:

```
mpc480(config)#host PE
PE(config)#ip vrf blue
PE(config-vrf)#rd 1:1
PE(config-vrf)#ip vrf red
PE(config-vrf)#rd 2:2
PE(config-vrf)#ip vrf yellow
PE(config-vrf)#rd 3:3
```

Once the VRFs are created, the VRFs are applied to the virtual interfaces configured on the PE router:

```
PE(config)#int gi1/vid2
PE(config-int-vid)#encapsulation dot1q 2
PE(config-int-vid)#ip vrf forwarding blue
PE(config-int-vid)#ip address 1.1.1.1 255.255.255.252
PE(config-int-vid)#int gi1/3
PE(config-int-vid#encapsulation dot1q 3
PE(config-int-vid)#ip vrf forwarding red
PE(config-int-vid)#ip address 1.1.1.1 255.255.255.252
PE(config-int-vid)#int gi1/4
PE(config-int-vid)#encapsulation dot1q 4
PE(config-int-vid)#ip vrf forwarding yellow
PE(config-int-vid)#ip address 1.1.1.1 255.255.255.252
```

Note that all interfaces have the same IP address which is 1.1.1.1. With no VRF, the router will warn that overlapping IP addresses are not allowed. The command **ip vrf forwarding** binds an interface to a specific VRF.

To configure the Blue, Red and Yellow routers with 1.1.1.2/30 on their Gigabitethernet interfaces, you can ping 1.1.1.1 from each of the routers to test reachability to the PE router.

```
Blue#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/35/80
ms

Red#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/48/
156 ms

Yellow#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/60/
136 ms
```

Now that the PE connectivity from the CE routers is confirmed, we also need to test the connectivity to each of the VRFs. Since all the VRFs have the same IP, we need to specify the VRF name using the ping vrf command

```
PE#ping vrf blue 1.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/28/68
ms

PE#ping vrf red 1.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/28/88
ms

PE#ping vrf yellow 1.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/31/68
ms
```

Note: *Note that duplicate IP addresses are possible with VRF. VRFs are usually configured on PE routers. CE routers normally do not make use of VRFs.*

Multicast VRF

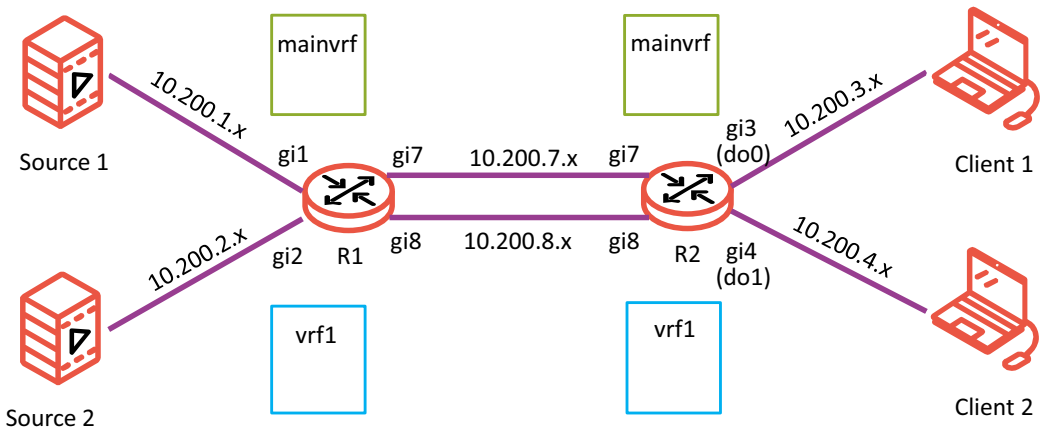
This section describes the configuration of multicast traffic in VRFs. The following use cases are described:

- Configure PIM-SM in VRF
- Configure IGMP-Querier in VRF
- Configure S,G route in VRF

Configure PIM-SM in VRF

To configure PIM-SM for multicast forwarding within multiple VRFs, the following topology is used.

Figure 100 Multicast Forwarding with PIM-SM in VRF



The multicast stream is forwarded in two separate VRFs; mainvrf and vrf1.

- source1 sends multicast group 230.0.0.1 to be forwarded by VRF mainvrf
- source2 sends multicast group 230.0.0.1 to be forwarded by VRF vrf1
- client1 joins group (*, 230.0.0.1)
- client2 sends multicast group 230.0.0.1 to be forwarded by VRF mainvrf

The procedure for the configuring a multicast VRF is as follows:

1. Bind interfaces to VRF and enable PIM
2. Enable OSPF for VRF
3. Configure PIM for VRF

To configure router R1, do the following:

1. Bind interfaces to VRF mainvrf and enable PIM on interfaces

```
R1# configure terminal
R1(config)# interface gigabitethernet1
R1(config-if-gi)# ip address 10.200.1.10/24
R1(config-if-gi)# ip pim sparse-mode
R1(config-if-gi)# no shutdown
```

```
R1(config-if-gi)# interface gigabitethernet7
R1(config-if-gi)# ip address 10.200.7.1/24
R1(config-if-gi)# ip pim sparse-mode
R1(config-if-gi)# no shutdown

R1(config-if-gi)# interface loopback0
R1(config-if-lo)# ip address 10.0.0.11/32
R1(config-if-lo)# ip pim sparse-mode
R1(config-if-lo)# no shutdown
```

2. Enable OSPF for the VRF mainvrf:

```
R1(config)# router ospf
R1(config-router-ospf)# router-id 10.0.0.11
R1(config-router-ospf)# network 10.0.0.11/32 area 0.0.0.11
R1(config-router-ospf)# network 10.200.1.0/24 area 0.0.0.11
R1(config-router-ospf)# network 10.200.7.0/24 area 0.0.0.11
R1(config-router-ospf)# end
R1#
```

3. Configure PIM for the VRF mainvrf:

```
R1# configure terminal
R1(config)# ip pim rp-address 10.0.0.11
R1(config)# ip pim spt-threshold
```

To configure router R2, do the following:

1. Bind interfaces to VRF mainvrf and enable PIM

```
R2# configure terminal
R2(config-if-gi)# interface downstream0
R2(config-if-downstream)# ip address 10.200.3.10 255.255.255.0
R2(config-if-downstream)# ip pim sparse-mode
R2(config-if-downstream)# no shutdown
R2(config-if-downstream)# interface gigabitethernet7
R2(config-if-gi)# ip address 10.200.7.2/24
R2(config-if-gi)# ip pim sparse-mode
R2(config-if-gi)# no shutdown
R2(config-if-gi)# interface loopback0
R2(config-if-lo)# ip address 10.0.0.12/32
R2(config-if-lo)# ip pim sparse-mode
R2(config-if-lo)# no shutdown
```

2. Map client interface to interface downstream of VRF mainvrf:

```
R2(config-if-lo)# interface gigabitethernet3
R2(config-if-gi)# arp proxy
R2(config-if-gi)# ip igmp downstream-map
R2(config-if-gi)# no shutdown
```

3. Enable OSPF for VRF mainvrf:

```
R2(config)# router ospf
R2(config-router-ospf)# router-id 10.0.0.12
R2(config-router-ospf)# network 10.0.0.12/32 area 0.0.0.11
R2(config-router-ospf)# network 10.200.3.0/24 area 0.0.0.11
R2(config-router-ospf)# network 10.200.7.0/24 area 0.0.0.11
R2(config-router-ospf)# end
R2#
```

4. Configure PIM for VRF mainvrf

```
R2# configure terminal
R2(config)# ip pim rp-address 10.0.0.11
R2(config)# ip pim spt-threshold
```

To setup PIM in VRF vrf on router R1:

1. Bind interfaces to VRF vrf and enable PIM

```
R1# configure terminal
R1(config)# interface gigabitethernet2
R1(config-if-gi)# ip vrf forwarding vrf1
R1(config-if-gi)# ip address 10.200.2.10/24
R1(config-if-gi)# ip pim sparse-mode
R1(config-if-gi)# no shutdown
R1(config-if-gi)# interface gigabitethernet8
R1(config-if-gi)# ip vrf forwarding vrf1
R1(config-if-gi)# ip address 10.200.8.1/24
R1(config-if-gi)# ip pim sparse-mode
R1(config-if-gi)# no shutdown
R1(config-if-gi)# interface loopback0
R1(config-if-lo)# ip vrf forwarding vrf1
R1(config-if-lo)# ip address 10.0.0.21/32
R1(config-if-lo)# ip pim sparse-mode
R1(config-if-lo)# no shutdown
```

2. Enable OSPF for VRF vrf:

```
R1(config)# router ospf 1 vrf1
R1(config-router-ospf)# router-id 10.0.0.21
R1(config-router-ospf)# network 10.0.0.21/32 area 0.0.0.21
R1(config-router-ospf)# network 10.200.2.0/24 area 0.0.0.21
R1(config-router-ospf)# network 10.200.8.0/24 area 0.0.0.21
R1(config-router-ospf)# end
R1#
```

3. Configure PIM for VRF vrf:

```
R1# configure terminal
R1(config)# ip pim vrf vrf1 rp-address 10.0.0.21
R1(config)# ip pim vrf vrf1 spt-threshold
```

To configure router R2, do the following:

1. Bind interfaces to VRF vrf and enable PIM

```
R2# configure terminal
R2(config-if-gi)# interface downstream1
R2(config-if-downstream)# ip vrf forwarding vrf1
R2(config-if-downstream)# ip address 10.200.4.10 255.255.255.0
R2(config-if-downstream)# ip pim sparse-mode
R2(config-if-downstream)# no shutdown
R2(config-if-downstream)# interface gigabitethernet8
R1(config-if-gi)# ip vrf forwarding vrf1
R2(config-if-gi)# ip address 10.200.8.2/24
R2(config-if-gi)# ip pim sparse-mode
R2(config-if-gi)# no shutdown
R2(config-if-gi)# interface loopback1
R2(config-if-lo)# ip vrf forwarding vrf1
R2(config-if-lo)# ip address 10.0.0.22/32
R2(config-if-lo)# ip pim sparse-mode
R2(config-if-lo)# no shutdown
```

2. Map client interface to downstream interface of VRF vrf:

```
R2(config-if-lo)# interface gigabitethernet4
R1(config-if-gi)# ip vrf forwarding vrf1
R2(config-if-gi)# arp proxy
```

```
R2(config-if-gi)# ip igmp downstream-map
R2(config-if-gi)# no shutdown
```

3. Enable OSPF for VRF vrf:

```
R2(config)# router ospf 1 vrf1
R2(config-router-ospf)# router-id 10.0.0.22
R2(config-router-ospf)# network 10.0.0.22/32 area 0.0.0.21
R2(config-router-ospf)# network 10.200.4.0/24 area 0.0.0.21
R2(config-router-ospf)# network 10.200.8.0/24 area 0.0.0.21
R2(config-router-ospf)# end
R1#
```

4. Configure PIM for VRF vrf:

```
R2# configure terminal
R2(config)# ip pim vrf vrf1 rp-address 10.0.0.21
R2(config)# ip pim vrf vrf1 spt-threshold
```

The detailed configuration for router R1 will then be:

```
R1# show running-config
! version ibos-ibos-mpc-7.0.1-ED-R (ibos/ibos-ibos-mpc-7.0.1-ED-
R.bz2)
ip vrf vrf1

interface loopback0
 ip address 10.0.0.11 255.255.255.255
 pim sparse-mode
 shutdown

interface loopback1
 ip vrf forwarding vrf1
 ip address 10.0.0.21 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface gigabitethernet1
 ip address 10.200.1.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet2
 ip vrf forwarding vrf1
 ip address 10.200.2.10 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
interface gigabitethernet7
 ip address 10.200.7.1 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 ip address 10.200.8.1 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
```

```
interface gigabitethernet23
 no shutdown

interface gigabitethernet24
 description Management interface
 ip address 10.192.0.189 255.255.255.0
 no shutdown

interface tengigabitethernet1
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
 no shutdown


router ospf
 router-id 10.0.0.11
 network 10.0.0.11/32 area 0.0.0.11
 network 10.200.1.0/24 area 0.0.0.11
 network 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.21
 network 10.0.0.21/32 area 0.0.0.21
 network 10.200.2.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

hostname R1

exception dump 172.19.36.4 10000

logging console debugging
logging monitor debugging
service timestamp debug datetime
service timestamp log datetime

ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4

ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username test password B0GhWKgQdx3fo encrypted
username automat password B0/oaq4.CM7hE encrypted
username t password SzoyzMXs5XCWQ encrypted
enable secret uflBpDArZbO22 encrypted
!end
R1#
```

The detailed configuration for router R2 is now:

```
R2# show running-config
! version ibos-ibos-mpc-7.0.0-ED-R (ibos/ibos-ibos-mpc-7.0.0-ED-
R.bz2)
ip vrf vrf1

interface loopback0
 address 10.0.0.12 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface loopback1
 ip vrf forwarding vrf1
 ip address 10.0.0.22 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface gigabitethernet1

interface gigabitethernet2

interface gigabitethernet3
 arp proxy
 ip igmp downstream-map
 no shutdown

interface gigabitethernet4
 ip vrf forwarding vrf1
 arp proxy
 igmp downstream-map
 shutdown
:
 gigabitethernet7
 ip address 10.200.7.2 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 ip address 10.200.8.2 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
interface gigabitethernet24
 description Management interface
 ip address 10.192.0.183 255.255.255.0
 no shutdown

interface tengigabitethernet1
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
```

```
 ip address 10.200.3.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface downstream1
 ip vrf forwarding vrf1
 ip address 10.200.4.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

 ospf
 capability vrf-lite
 router-id 10.0.0.12
 network 10.0.0.12/32 area 0.0.0.11
 network 10.200.3.0/24 area 0.0.0.11
 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.22
 network 10.0.0.22/32 area 0.0.0.21
 network 10.200.4.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

hostname R2

exception dump 172.19.36.4 10001

logging console debugging

ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4

ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username t password TgSZeGBCCdFaY encrypted
!end
R2#
```

Check PIM-SM VRF Configuration

The following examples demonstrate how to check that the PIM configuration is active and is correctly configured.

Check PIM Interface and Status

To check the PIM interfaces in VRF mainvrf on router R1, do the following:

```
R1# show ip pim interface
Address          Ver/ Nbr  DR              DR  Interface
                 Mode Cnt Pri          Address
-----------------------------------------------------------------
```

```
-------------
10.200.1.10      v2/S   0   1      10.200.1.10  gigabitethernet1
10.200.7.1       v2/S   1   1       10.200.7.2  gigabitethernet7
10.0.0.11        v2/S   0   1        10.0.0.11  loopback0

# pim interface in vrf vrf1:

R1# show ip pim vrf vrf1 interface
Address          Ver/ Nbr  DR              DR  Interface
                 Mode Cnt Pri          Address
-------------------------------------------------------------------
-------------
10.200.2.10      v2/S   0   1      10.200.2.10  gigabitethernet2
10.200.8.1       v2/S   1   1       10.200.8.2  gigabitethernet8
10.0.0.21        v2/S   0   1        10.0.0.21  loopback1
```

To check the PIM interfaces in VRF mainvrf on router 2, do the following:

```
R2# show ip pim interface
Address          Ver/ Nbr  DR              DR  Interface
                 Mode Cnt Pri          Address
-------------------------------------------------------------------
-------------
10.200.7.2       v2/S   1   1       10.200.7.2  gigabitethernet7
10.0.0.12        v2/S   0   1        10.0.0.12  loopback0
10.200.3.10      v2/S   0   1      10.200.3.10  downstream0
```

To check the PIM interfaces in VRF vrf1 on router 2:

```
R2# show ip pim vrf vrf1 interface
Address          Ver/ Nbr  DR              DR  Interface
                 Mode Cnt Pri          Address
-------------------------------------------------------------------
-------------
10.200.8.2       v2/S   1   1       10.200.8.2  gigabitethernet8
10.0.0.22        v2/S   0   1        10.0.0.22  loopback1
10.200.4.10      v2/S   0   1      10.200.4.10  downstream1
```

Check PIM Neighbour and Status

To check neighbour for VRF mainvrf on router R1, do the following:

```
R1# show ip pim neighbour
neighbour        Interface            Uptime/Expires    Ver   DR
Address                                              Priority/Mode
-------------------------------------------------------------------
-------------
10.200.7.2     gigabitethernet7     00:19:13/00:01:32       v2
1 / DR
```

To check the neighbour for VRF vrf1, do the following:

```
R1# show ip pim vrf vrf1 neighbour
neighbour        Interface            Uptime/Expires    Ver   DR
Address                                              Priority/Mode
-------------------------------------------------------------------
-------------
10.200.8.2     gigabitethernet8     00:19:17/00:01:29       v2
1 / DR
```

To check neighbours for VRF mainvrf on router R2:

```
R2# show ip pim neighbour
neighbour        Interface             Uptime/Expires    Ver   DR
Address                                                 Priority/Mode
--------------------------------------------------------------------
-------------
10.200.7.1     gigabitethernet7     00:19:29/00:01:17       v2
1 /
```

To check neighbours for VRF vrf1 on router R2:

```
R2# show ip pim vrf vrf1 neighbour
neighbour        Interface             Uptime/Expires    Ver   DR
Address                                                 Priority/Mode
--------------------------------------------------------------------
-------------
10.200.8.1     gigabitethernet8     00:19:35/00:01:40       v2
1 /
```

Check PIM RP Mapping Status

To check RP mapping for VRF mainvrf on router R1:

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
Override RP cnt: 0
Group(s): 224.0.0.0/4, Static
    RP: 10.0.0.11
         Uptime: 00:28:14
```

To check RP mapping for VRF vrf1 on router 1:

```
R1# show ip pim vrf vrf1 rp mapping
PIM Group-to-RP Mappings
Override RP cnt: 0
Group(s): 224.0.0.0/4, Static
    RP: 10.0.0.21
         Uptime: 00:29:06
```

To check RP mapping for VRF mainvrf on router 2:

```
R2# show ip pim rp mapping
PIM Group-to-RP Mappings
Override RP cnt: 0
Group(s): 224.0.0.0/4, Static
    RP: 10.0.0.11
         Uptime: 00:31:14
```

To check RP mapping for VRF vrf1 on router 2:

```
R2# show ip pim vrf vrf1 rp mapping
PIM Group-to-RP Mappings
Override RP cnt: 0
Group(s): 224.0.0.0/4, Static
    RP: 10.0.0.21
         Uptime: 00:31:21
```

Check PIM Multicast Route

To check the multicast route for VRF mainvrf on router 1:

```
R1# sh ip mroute
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
```

```
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(*, 230.0.1.1)
 RP: 10.0.0.11
 No upstream neighbour
 Upstream State: JOINED, SPT-Switch
 Macro State: JD(*,*,RP)
 Downstream: gigabitethernet7 (Vif3) Joined
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m16s)

(10.200.1.1, 230.0.1.1)
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 KAT Expiry: 1m16s
 Downstream: gigabitethernet7 (Vif3) Joined Outgoing
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m16s)

(10.200.1.1, 230.0.1.1, RPT)
 RP: 10.0.0.11
 No upstream neighbour
 Upstream State: NO INFO
 Macro State: RPTJ(G),RPF'(S,G,RPT)==RPF'(*,G)
 Downstream: gigabitethernet7 (Vif3) Outgoing
```

To check multicast route for VRF vrf1 on router 1:

```
R1# show ip mroute vrf vrf1
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(*, 230.0.1.1)
 RP: 10.0.0.21
 No upstream neighbour
 Upstream State: JOINED, SPT-Switch
 Macro State: JD(*,*,RP)
 Downstream: gigabitethernet8 (Vif3) Joined
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m16s)

(10.200.2.1, 230.0.1.1)
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 KAT Expiry: 1m16s
 Downstream: gigabitethernet8 (Vif3) Joined Outgoing
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m16s)

(10.200.2.1, 230.0.1.1, RPT)
 RP: 10.0.0.21
 No upstream neighbour
```

```
 Upstream State: NO INFO
 Macro State: RPTJ(G),RPF'(S,G,RPT)==RPF'(*,G)
 Downstream: gigabitethernet8 (Vif3) Outgoing
```

To check multicast route for VRF mainvrf on router 2:

```
R2# sh ip mroute

Macrostate legend:
    JD(*,*,RP)    JoinDesired(*,*,RP)
              CR(S,G)       CouldRegister(S,G)
              RPTJ(G)       RPTJoinDesired(G)
              PD(S,G,RPT)   PruneDesired(S,G,RPT)
              RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream   CA(*,G,I)     CouldAssert(*,G,I)
              ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(*, 230.0.1.1)
 RP: 10.0.0.11
 Upstream neighbour: 10.200.7.1 on gigabitethernet7 (Vif3)
 Upstream State: JOINED, SPT-Switch
 Macro State: JD(*,*,RP)
 JT Expiry: 18s
 Downstream: downstream0 (Vif2) Local
MacroState:CA(*,G,I),ATD(*,G,I)

(10.200.1.1, 230.0.1.1)
 Upstream neighbour: 10.200.7.1 on gigabitethernet7 (Vif3)
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 JT Expiry: 18s
 KAT Expiry: 2m48s
 Downstream: downstream0 (Vif2) Outgoing
             gigabitethernet3, Outgoing Snooped

(10.200.1.1, 230.0.1.1, RPT)
 RP: 10.0.0.11
 Upstream neighbour: 10.200.7.1 on gigabitethernet7 (Vif3)
 Upstream State: NO INFO
 Macro State: RPTJ(G),RPF'(S,G,RPT)==RPF'(*,G)
 Downstream:
```

To check multicast route for VRF vrf1 on router 2:

```
R2# show ip mroute vrf vrf1

Macrostate legend:
    JD(*,*,RP)    JoinDesired(*,*,RP)
              CR(S,G)       CouldRegister(S,G)
              RPTJ(G)       RPTJoinDesired(G)
              PD(S,G,RPT)   PruneDesired(S,G,RPT)
              RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream   CA(*,G,I)     CouldAssert(*,G,I)
              ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(*, 230.0.1.1)
 RP: 10.0.0.21
 Upstream neighbour: 10.200.8.1 on gigabitethernet8 (Vif3)
 Upstream State: JOINED, SPT-Switch
 Macro State: JD(*,*,RP)
 JT Expiry: 30s
```

```
 Downstream: downstream1 (Vif2) Local
MacroState:CA(*,G,I),ATD(*,G,I)

(10.200.2.1, 230.0.1.1)
 Upstream neighbour: 10.200.8.1 on gigabitethernet8 (Vif3)
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 JT Expiry: 30s
 KAT Expiry: 2m
 Downstream: downstream1 (Vif2) Outgoing
             gigabitethernet4, Outgoing Snooped

(10.200.2.1, 230.0.1.1, RPT)
 RP: 10.0.0.21
 Upstream neighbour: 10.200.8.1 on gigabitethernet8 (Vif3)
 Upstream State: NO INFO
 Macro State: RPTJ(G),RPF'(S,G,RPT)==RPF'(*,G)
 Downstream:
```

Check Multicast Forwarding Status

To check multicast forwarding for VRF mainvrf on router 1:

```
R1# show ip multicast forwarding
(10.200.1.1, 230.0.1.1) upstream gigabitethernet1, ttl: 151, cos: 0
   33999 packets, 46170642 bytes, 0 wrong parent, 0 output errors
   1086400 bits/sec (1.09 Mbps), 100 packets/sec
    Downstream: gigabitethernet7 (Hardware switched)
```

To check multicast forwarding for VRF vrf1 on router 1:

```
R1# show ip multicast vrf vrf1 forwarding
(10.200.2.1, 230.0.1.1) upstream gigabitethernet2, ttl: 151, cos: 0
   40599 packets, 55133428 bytes, 0 wrong parent, 0 output errors
   1085312 bits/sec (1.09 Mbps), 99 packets/sec
    Downstream: gigabitethernet8 (Hardware switched)
```

To check multicast forwarding for VRF mainvrf on router 2:

```
R2# show ip multicast forwarding
(10.200.1.1, 230.0.1.1) upstream gigabitethernet7, ttl: 150, cos: 0
   17299 packets, 23492042 bytes, 0 wrong parent, 0 output errors
   1086400 bits/sec (1.09 Mbps), 100 packets/sec
    Downstream: downstream0 (IGMP snooped) (Hardware switched)
        Joined: gigabitethernet3
```

To check multicast forwarding for VRF vrf1 on router 2:

```
R2# show ip multicast vrf vrf1 forwarding
(10.200.2.1, 230.0.1.1) upstream gigabitethernet8, ttl: 150, cos: 0
   17499 packets, 23763642 bytes, 0 wrong parent, 0 output errors
   1086400 bits/sec (1.09 Mbps), 100 packets/sec
    Downstream: downstream1 (IGMP snooped) (Hardware switched)
        Joined: gigabitethernet4
```

Check PIM Local Member Status

To check that a local member is only joined on R2 and that the local member joined in VRF mainvrf:

```
R2# show ip pim local-members
PIM Local membership information

gigabitethernet7:
```

```
loopback0:

downstream0:
  (*, 230.0.1.1) : Include
```

To check local members that joined in VRF vrf1:

```
R2# show ip pim vrf vrf1 local-members
PIM Local membership information

gigabitethernet8:

loopback1:

downstream1:
  (*, 230.0.1.1) : Include
```

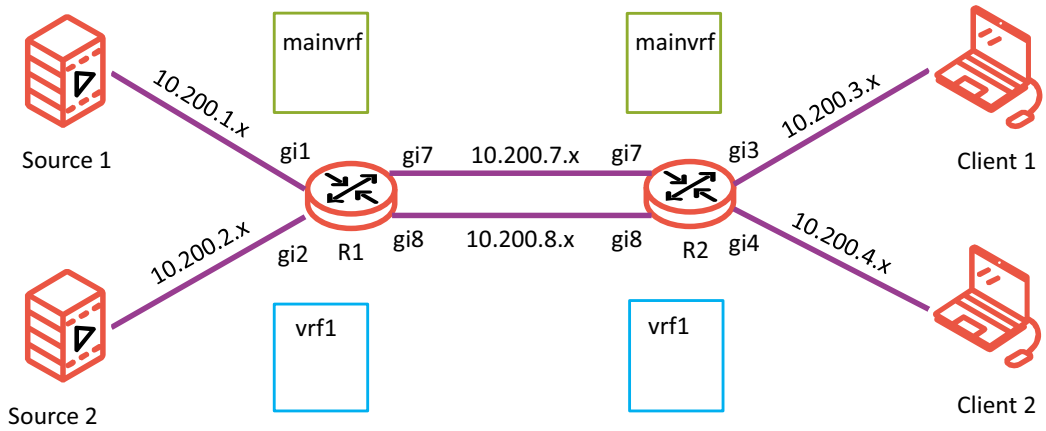
To check the snooped group status on each interface:

```
R2# show ip igmp snooper groups
Snooped groups on all interfaces
 V  Group          Snooped Interface                    Age  MisRprt
                        Reporter/Source
 ------------------------------------------------------------------
----------
  230.0.1.1       gigabitethernet3                      6m5s       0
                    10.200.3.1                          6m5s       0
 2  230.0.1.1       gigabitethernet4                     6m3s       0
                    10.200.4.1                          6m3s       0
```

Configure PIM-SSM in VRF with Downstream Map

To configure PIM-SSM on one or more interfaces within each VRF, the following topology is used.

Figure 101 PIM-SSM in VRF



Source1 sends multicast group 232.0.0.1 to be forwarded by VRF mainvrf

Source2 sends multicast group 232.0.0.1 to be forwarded by VRF vrf1

The multicast streams are forwarded in VRFs mainvrf and vrf1 separately.

Client1 joins (*, 232.0.0.1), which is ssm-mapped to the IGMPv3 join (10.200.1.1, 232.0.0.1).

Client2 joins (*, 232.0.0.1), which is ssm-mapped to IGMPv3 join (10.200.2.1, 32.0.0.1).

Configure PIM-SSM for VRF

The procedure to configure the above topology is identical to "Configure PIM-SM in VRF" on page 357.

In addition, the IGMP SSM group need to be configured for the client facing router R2.

Configure SSM Group Access-list for VRF

To setup igmp ssm-map group access list for VRF mainvrf on router 2, do the following:

```
R2# configure terminal
R2(config)# access-list ssm-acl-main
R2(config-acl)# permit ip 232.0.0.0/24
R2(config-acl)#end
```

To setup igmp ssm-map for VRF mainvrf on router 2, do the following:

```
R2# configure terminal
R2(config)# ip igmp ssm-map static ssm-acl-main 10.200.1.1
R2(config)# ip igmp ssm-map enable
R2(config-acl)#end
```

To setup igmp ssm-map group access-list for VRF vrf1 on router 2:

```
R2# configure terminal
R2(config)# access-list ssm-acl-vrf
R2(config-acl)# permit ip 232.0.0.0/24
R2(config-acl)#end
```

To setup igmp ssm-map for VRF vrf1 on router 2:

```
R2# configure terminal
R2(config)# ip igmp vrf vrf1 ssm-map static ssm-acl-vrf 10.200.2.1
R2(config)# ip igmp vrf vrf1 ssm-map enable
R2(config-acl)#end
```

Check PIM-SSM for VRF Configuration

After performing the configuration, check the current configuration on router R1:

```
R1# show running-config
! version ibos-mpc-7.0.0-ED-R (ibos/ibos-mpc-7.0.0-ED-R.bz2)
ip vrf vrf1

interface loopback0
 ip address 10.0.0.11 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface loopback1
 ip vrf forwarding vrf1
 ip address 10.0.0.21 255.255.255.255
 ip pim sparse-mode
 no shutdown
```

```
interface gigabitethernet1
 ip address 10.200.1.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet2
 ip vrf forwarding vrf1
 ip address 10.200.2.10 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
interface gigabitethernet7
 ip address 10.200.7.1 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 ip address 10.200.8.1 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:i
interface gigabitethernet23
 no shutdown

interface gigabitethernet24
 description Management interface
 ip address 10.192.0.189 255.255.255.0
 no shutdown

interface tengigabitethernet1
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
 no shutdown

router ospf
 router-id 10.0.0.11
 network 10.0.0.11/32 area 0.0.0.11
 network 10.200.1.0/24 area 0.0.0.11
 network 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.21
 network 10.0.0.21/32 area 0.0.0.21
 network 10.200.2.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

hostname R1

exception dump 172.19.36.4 10000
```

```
logging console debugging
logging monitor debugging
service timestamp debug datetime
service timestamp log datetime

ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4

ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username test password B0GhWKgQdx3fo encrypted
username automat password B0/oaq4.CM7hE encrypted
username t password SzoyzMXs5XCWQ encrypted
enable secret uflBpDArZbO22 encrypted
!end
R1#
```

To check the current configuration on router 2:

```
R2# sh run
! version ibos-ibos-mpc-7.0.0-ED-R (ibos/ibos-ibos-mpc-7.0.0-ED-
R.bz2)
ip vrf vrf1

interface loopback0
 ip address 10.0.0.12 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface loopback1
 ip vrf forwarding vrf1
 ip address 10.0.0.22 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface gigabitethernet1

interface gigabitethernet2

interface gigabitethernet3
 ip address 10.200.3.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet4
 ip vrf forwarding vrf1
 ip address 10.200.4.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet5
```

```
interface gigabitethernet6

interface gigabitethernet7
 ip address 10.200.7.2 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 ip address 10.200.8.2 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
interface gigabitethernet24
 description Management interface
 ip address 10.192.0.183 255.255.255.0
 no shutdown

interface tengigabitethernet1
 ip pim sparse-mode
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
 no shutdown

router ospf
 capability vrf-lite
 router-id 10.0.0.12
 network 10.0.0.12/32 area 0.0.0.11
 network 10.200.3.0/24 area 0.0.0.11
 network 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.22
 network 10.0.0.22/32 area 0.0.0.21
 network 10.200.4.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

ip multicast shaping default 10000 450

hostname R2

exception dump 172.19.36.4 10001

ip igmp ssm-map static ssm-acl-main 10.200.1.1
ip igmp vrf vrf1 ssm-map static ssm-acl-vrf 10.200.2.1

logging console debugging

ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4
```

```
ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username t password TgSZeGBCCdFaY encrypted

access-list ssm-acl-main
 seq 10 permit ip 232.0.0.0/24

access-list ssm-acl-vrf
 seq 10 permit ip 232.0.0.0/24
!end
```

Check IGMP SSM Map

The ssm-map feature only needs to be configured for the client facing router 2.

To check ssm-map for VRF mainvrf on router 2:

```
R2# show ip igmp ssm-map
SSM Mapping : Enabled
Database    : Static mappings configured

R2# show ip igmp ssm-map 232.0.0.1
Group address: 232.0.0.1
Database     : Static
Source list  : 10.200.1.1
```

To check ssm-map for VRF vrf1 on router 2:

```
R2# show ip igmp vrf vrf1 ssm-map
SSM Mapping : Enabled
Database    : Static mappings configured
R2# show ip igmp vrf vrf1 ssm-map 232.0.0.1
Group address: 232.0.0.1
Database     : Static
Source list  : 10.200.2.1
```

To check the multicast route for VRF mainvrf on router R1:

```
sr1# show ip mroute
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)

 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.1.1, 232.0.0.1)
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 KAT Expiry: 1m19s
 Downstream: gigabitethernet7 (Vif3) Joined Outgoing
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m19s)
```

To check multicast route for VRF vrf1 on router R1:

```
R1# show ip mroute vrf vrf1
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.2.1, 232.0.0.1)
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 KAT Expiry: 4s
 Downstream: gigabitethernet8 (Vif3) Joined Outgoing
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:2m34s)
```

To check multicast route for VRF mainvrf on router R2:

```
R2# show ip mroute
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.1.1, 232.0.0.1)
 Upstream neighbour: 10.200.7.1 on gigabitethernet7 (Vif3)
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 Expiry: 37s
 KAT Expiry: 7s
 Downstream: gigabitethernet3 (Vif2) Local Outgoing
 MacroState:CA(*,G,I),ATD(*,G,I)
```

To check multicast route for VRF vrf1:

```
R2# show ip mroute vrf vrf1
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.2.1, 232.0.0.1)
 Upstream neighbour: 10.200.8.1 on gigabitethernet8 (Vif3)
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 JT Expiry: 2s
 KAT Expiry: 2m32s
 Downstream: gigabitethernet4 (Vif2) Local Outgoing
MacroState:CA(*,G,I),ATD(*,G,I)
```

Check PIM Multicast Route Status

To check PIM local member status, verify that:

- local member is only joined on client facing router
- local member joined in mainvrf

```
R2# show ip pim local-members
PIM Local membership information

gigabitethernet3:
  (10.200.1.1, 232.0.0.1) : Include

gigabitethernet7:

loopback0:

# local member joined in vrf vrf1

R2# show ip pim vrf vrf1 local-members
PIM Local membership information

gigabitethernet4:
  (10.200.2.1, 232.0.0.1) : Include
:

loopback1:
```

Check Status of iGMP Groups

To check the status of the iGMP groups, verify the following:

- local groups are only joined on client facing router
- local groups exist for VRF mainvrf

```
R2# show ip igmp groups detail

Interface:      gigabitethernet3
Group:          232.0.0.1
Flags:          R
Uptime:         00:31:13
Group mode:     Include (Expires: 00:02:29)
Last reporter:  10.200.3.1
Group source list: (R - Remote, M - SSM Mapping, S - Static, L -
Local)

Include Source List :
  Source Address  Uptime    v3 Exp    Fwd  Flags
  10.200.1.1      00:01:51  00:02:29  Yes  M
```

To view the local groups for the VRF “vrf1”, do the following:

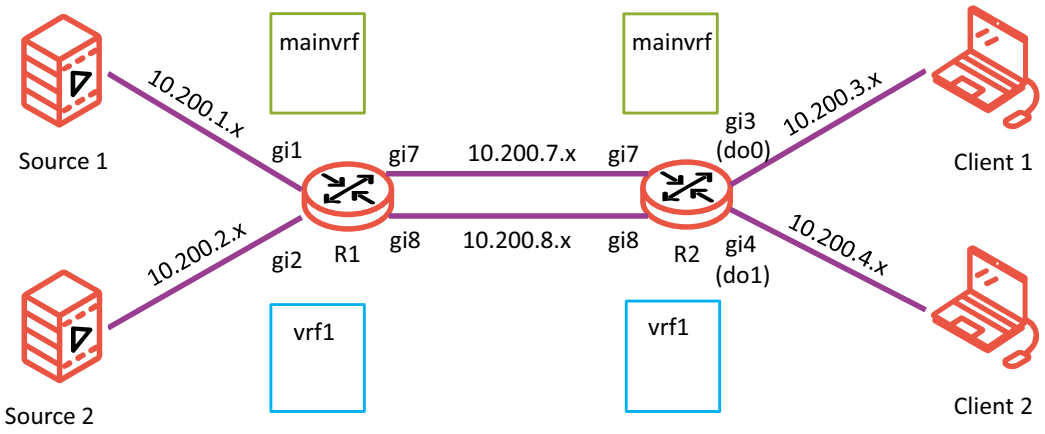
```
R2# show ip igmp vrf vrf1 groups detail

Interface:      gigabitethernet4
Group:          232.0.0.1
Flags:          R
Uptime:         00:31:15
Group mode:     Include (Expires: 00:02:32)
Last reporter:  10.200.4.1
Group source list: (R - Remote, M - SSM Mapping, S - Static, L -
```

```
Local)

Include Source List :
  Source Address  Uptime    v3 Exp    Fwd  Flags
  10.200.2.1      00:01:57  00:02:55  Yes  M
```

Configure PIM-SM in VRF with SSM Map

It is possible to have the same S,G to exist in separate VRFs with different actual S and G. For this use case, the topology is as follows:

Figure 102 PIM-SM in VRF



In the above example, the following applies:

- source1 sends multicast group 232.0.0.1 to be forwarded by VRF mainvrf
- source2 sends multicast group:232.0.0.1 to be forwarded by VRF vrf1

The multicast streams are separatelyforwarded in VRFs mainvrf and vrf1.

Client1 joins (*, 232.0.0.1), which is ssm-mapped to IGMPv3 join (10.200.1.1, 232.0.0.1)

Client2 joins (*, 232.0.0.1), which is ssm-mapped to IGMPv3 join (10.200.2.1, 232.0.0.1)

To setup PIM in the example is identical to “Configure PIM-SM in VRF” on page 357.

Configure PIM-SSM on VRFs with Same S,G

Once PIM is configured, then the igmp ssm-map needs to be setup in the VRFs on the client-facing router. I

To set up PIM-SSM on the VRFs with the same S,G IP address, do the following:

1. Setup igmp ssm-map group acl for main vrf on router 2:

```
R2# configure terminal
R2(config)# access-list ssm-acl-main
R2(config-acl)# permit ip 232.0.0.0/24
R2(config-acl)#end
```

2. Setup igmp ssm-map for main vrf

```
R2# configure terminal
R2(config)# ip igmp ssm-map static ssm-acl-main 10.200.1.1
R2(config)# ip igmp ssm-map enable
R2(config-acl)#end
```

3. Setup igmp ssm-map group acl for vrf vrf1

```
R2# configure terminal
R2(config)# access-list ssm-acl-vrf
R2(config-acl)# permit ip 232.0.0.0/24
R2(config-acl)#end
```

4. Setup igmp ssm-map for vrf vrf1

```
R2# configure terminal
R2(config)# ip igmp vrf vrf1 ssm-map static ssm-acl-vrf 10.200.2.1
R2(config)# ip igmp vrf vrf1 ssm-map enable
R2(config-acl)#end
```

The detailed configuration for router R1 will be:

```
R1# show running-config
! version ibos-ibos-mpc-7.0.0-ED-R (ibos/ibos-ibos-mpc-7.0.0-ED-
R.bz2)
ip vrf vrf1

interface loopback0
 ip address 10.0.0.11 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface loopback1
 ip vrf forwarding vrf1
 ip address 10.0.0.21 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface gigabitethernet1
 ip address 10.200.1.10 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet2
 ip vrf forwarding vrf1
 ip address 10.200.2.10 255.255.255.0
 ip pim sparse-mode
 no shutdown
:
:
interface gigabitethernet7
 ip address 10.200.7.1 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 ip address 10.200.8.1 255.255.255.0
 ip pim sparse-mode
 shutdown
:
:
```

```
interface gigabitethernet23
 no shutdown
 gigabitethernet24
 description Management interface
 ip address 10.192.0.189 255.255.255.0
 no shutdown

interface tengigabitethernet1
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
 no shutdown

router ospf
 router-id 10.0.0.11
 network 10.0.0.11/32 area 0.0.0.11
 network 10.200.1.0/24 area 0.0.0.11
 network 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.21
 network 10.0.0.21/32 area 0.0.0.21
 network 10.200.2.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

hostname R1

exception dump 172.19.36.4 10000

logging console debugging
logging monitor debugging
service timestamp debug datetime
service timestamp log datetime

ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4

ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username test password B0GhWKgQdx3fo encrypted
username automat password B0/oaq4.CM7hE encrypted
username t password SzoyzMXs5XCWQ encrypted
enable secret uflBpDArZbO22 encrypted
!end
R1#
```

The detailed configuration for router R2 will be:

```
R2# sh run
! version ibos-ibos-mpc-7.0.0-ED-R (ibos/ibos-ibos-mpc-7.0.0-ED-
R.bz2)
ip vrf vrf1

interface loopback0
 ip address 10.0.0.12 255.255.255.255
 ip pim sparse-mode
 no shutdown

interface loopback1
 vrf forwarding vrf1
 ip address 10.0.0.22 255.255.255.255
 ip pim sparse-mode
 no shutdown
 gigabitethernet1

interface gigabitethernet2


interface gigabitethernet3
 ip address 10.200.3.10 255.255.255.0
 ip pim sparse-mode
 shutdown

 gigabitethernet4
 ip vrf forwarding vrf1
 ip address 10.200.4.10 255.255.255.0
 ip pim sparse-mode
 shutdown

interface gigabitethernet5

interface gigabitethernet6

interface gigabitethernet7
 ip address 10.200.7.2 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet8
 ip vrf forwarding vrf1
 address 10.200.8.2 255.255.255.0
 ip pim sparse-mode
 no shutdown

interface gigabitethernet9

interface gigabitethernet10

interface gigabitethernet11

interface gigabitethernet12

interface gigabitethernet13

interface gigabitethernet14
```

```
interface gigabitethernet15

interface gigabitethernet16

interface gigabitethernet17

interface gigabitethernet18

interface gigabitethernet19

interface gigabitethernet20

interface gigabitethernet21

interface gigabitethernet22

interface gigabitethernet23

interface gigabitethernet24
 description Management interface
 ip address 10.192.0.183 255.255.255.0
 no shutdown

interface tengigabitethernet1
 ip pim sparse-mode
 no shutdown
 port uplink

interface tengigabitethernet2
 no shutdown
 port uplink

interface downstream0
 no shutdown

router ospf
 capability vrf-lite
 router-id 10.0.0.12
 network 10.0.0.12/32 area 0.0.0.11
 network 10.200.3.0/24 area 0.0.0.11
 10.200.7.0/24 area 0.0.0.11

router ospf 1 vrf1
 router-id 10.0.0.22
 network 10.0.0.22/32 area 0.0.0.21
 network 10.200.4.0/24 area 0.0.0.21
 network 10.200.8.0/24 area 0.0.0.21

ip multicast shaping default 10000 450

hostname R2

exception dump 172.19.36.4 10001

ip igmp ssm-map static ssm-acl-main 10.200.1.1
ip igmp vrf vrf1 ssm-map static ssm-acl-vrf 10.200.2.1

logging console debugging
```

```
ip route 10.192.0.0/10 10.192.0.1
ip route 172.19.0.0/16 10.192.0.1

ntp server 172.19.36.4

ip pim rp-address 10.0.0.11
ip pim spt-threshold
ip pim vrf vrf1 rp-address 10.0.0.21
ip pim vrf vrf1 spt-threshold

clock timezone Asia/Shanghai

username t password TgSZeGBCCdFaY encrypted

access-list ssm-acl-main
 seq 10 permit ip 232.0.0.0/24

access-list ssm-acl-vrf
 seq 10 permit ip 232.0.0.0/24
!end
```

Check igmp ssm-Map Status

The ssm-map only needs to be configured on the client-facing router.

Check the ssm-map for VRF mainvrf on router R2:

```
R2# show ip igmp ssm-map
SSM Mapping : Enabled
Database    : Static mappings configured

R2# show ip igmp ssm-map 232.0.0.1
Group address: 232.0.0.1
Database     : Static
Source list  : 10.200.1.1
```

Check the ssm-map for VRF vrf1 on router R2:

```
R2# show ip igmp vrf vrf1 ssm-map
SSM Mapping : Enabled
Database    : Static mappings configured
R2# show ip igmp vrf vrf1 ssm-map 232.0.0.1
Group address: 232.0.0.1
Database     : Static
Source list  : 10.200.2.1
```

Check PIM Multicast Route Status:

Check the multicast route for VRF mainvrf on router R1:

```
R1# show ip mroute
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.1.1, 232.0.0.1)
```

```
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 KAT Expiry: 1m19s
 Downstream: gigabitethernet7 (Vif3) Joined Outgoing
 MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:3m19s)
```

Check the multicast route for VRF vrf1 on router R1:

```
R1# show ip mroute vrf vrf1
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.2.1, 232.0.0.1)
 No upstream neighbour
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 Expiry: 4s
 Downstream: gigabitethernet8 (Vif3) Joined Outgoing
MacroState:CA(*,G,I),ATD(*,G,I) (ET Expiry:2m34s)
```

Check the multicast route for VRF mainvrf on router R2:

```
R2# show ip mroute
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.1.1, 232.0.0.1)
 Upstream neighbour: 10.200.7.1 on gigabitethernet7 (Vif3)
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 JT Expiry: 37s
 KAT Expiry: 7s
 Downstream: gigabitethernet3 (Vif2) Local Outgoing
MacroState:CA(*,G,I),ATD(*,G,I)
```

Check the multicast route for VRF vrf 1 on router R2:

```
R2# show ip mroute vrf vrf1
Macrostate legend:
 Upstream    JD(*,*,RP)    JoinDesired(*,*,RP)
             CR(S,G)       CouldRegister(S,G)
             RPTJ(G)       RPTJoinDesired(G)
             PD(S,G,RPT)   PruneDesired(S,G,RPT)
             RPF'(S,G,RPT) RPF'(S,G,RPT)
 Downstream  CA(*,G,I)     CouldAssert(*,G,I)
             ATD(*,G,I)    AssertTrackingDesired(*,G,I)

(10.200.2.1, 232.0.0.1)
 Upstream neighbour: 10.200.8.1 on gigabitethernet8 (Vif3)
```

```
 Upstream State: JOINED, Forwarding, SPT bit: 1
 Macro State: JD(*,*,RP)
 JT Expiry: 2s
 KAT Expiry: 2m32s
 Downstream: gigabitethernet4 (Vif2) Local Outgoing
MacroState:CA(*,G,I),ATD(*,G,I)
```

Check PIM Local Member Status:

Verify that local member is only joined on client-facing router and that the local member joined in VRF mainvrf:

```
R2# show ip pim local-members
PIM Local membership information

gigabitethernet3:
  (10.200.1.1, 232.0.0.1) : Include

gigabitethernet7:

loopback0:
```

Verify that the local member joined in VRF vrf1 on router R2:

```
R2# show ip pim vrf vrf1 local-members
PIM Local membership information

gigabitethernet4:
  (10.200.2.1, 232.0.0.1) : Include

gigabitethernet8:

loopback1:
```

Check IGMP Group Status

Verify that local groups are only joined on client-facing router.

Check local groups for VRF mainvrf on router R2:

```
R2# show ip igmp groups detail

Interface:      gigabitethernet3
Group:          232.0.0.1
Flags:          R
Uptime:         00:31:13
Group mode:     Include (Expires: 00:02:29)
Last reporter:  10.200.3.1
Group source list: (R - Remote, M - SSM Mapping, S - Static, L -
Local)

Include Source List :
  Source Address  Uptime    v3 Exp    Fwd  Flags
  10.200.1.1      00:01:51  00:02:29  Yes  M
```

Check local groups for VRF vrf1on router R2:

```
R2# show ip igmp vrf vrf1 groups detail

Interface:      gigabitethernet4
Group:          232.0.0.1
Flags:          R
Uptime:         00:31:15
```

```
Group mode:     Include (Expires: 00:02:32)
Last reporter:  10.200.4.1
Group source list: (R - Remote, M - SSM Mapping, S - Static, L -
Local)

Include Source List :
  Source Address  Uptime    v3 Exp    Fwd  Flags
  10.200.2.1      00:01:57  00:02:55  Yes  M
```

VRF Commands used by iBOS

The following VRF commands are available in iBOS 7:

- **ip vrf**
- **show ip vrf**
- **ip route vrf**
- **ping vrf**
- **show ip route vrf**
- **show arp vrf**
- **show ip forwarding vrf**

Refer *iBOS Command Reference* for more details on each command.

Configuring IS-IS

The IS-IS protocol is an interior gateway protocol (IGP) that uses link-state information to make routing decisions.

IS-IS is a link-state IGP that uses the shortest-path-first (SPF) algorithm to determine routes. IS-IS evaluates the topology changes and determines whether to perform a full SPF recalculation or a partial route calculation (PRC). This protocol originally was developed for routing International Organization for Standardization (ISO) Connectionless Network Protocol (CLNP) packets.

Like OSPF routing, IS-IS uses hello packets to allow quick network convergence when changes are detected. Using SPF, IS-IS evaluates network topology changes and determines if a full or partial route calculation is required.

Note: *Because IS-IS uses ISO addresses, the configuration of IP version 6 (IPv6) and IP version 4 (IPv4) implementations of IS-IS is identical.*

This chapter contains generic IS-IS configuration examples. To see details about the commands used in these examples, refer *iBOS Command Reference*.

IS-IS configuration in iBOS is described in the following sections:

- "Introduction to IS-IS" on page 388
- "Basis IS-IS Configuration" on page 389
- "IS-ISv6 Configuration" on page 413

Introduction to IS-IS

IS-IS is a link-state interior gateway routing protocol. Like OSPF, IS-IS runs the shortest path first (SPF) algorithm to create a database of the network topology and then determine the best (or shortest) path to a destination.

Unlike OSPF and other IP routing protocols, IS-IS runs directly on the data link layer (Layer 2) and uses a different terminology for naming packets. Packets that IS-IS routers send to each other are called link-state protocol units (link-state PDUs or LSPs). The LSP address, or Net Entitiy Title (NET), can be 8 to 20 bytes in length and consists of three parts:

- *Area identifier:* This field is variable and can be 1 to 13 bytes in length. The area can be any number. The first byte used is often 49 to indicate private use. Also, one area is usually sufficient for many networks so a common area identifier is 49.0001.
- *System identifier:* These six bytes identify the node, or router, in the network and is equivalent to the host address portion of the router's IP address.

  You can choose any value for the system identifier, a commonly used method involves using the router's IP address from the loopback interface and padding leading spaces with zeros. For example, a router with a loopback interface IP address 192.168.1.2 will be converted to the IS-IS system identifier 1921.6800.1002.

  Another method is to convert the MAC address to three two byte numbers. For example, the IS-IS system identifier for a router with the MAC address 00:1B:63:21:86:BE is 001b.6331.86be.
- *NSEL:* The final byte is the NET selector (NSEL) and is always "00" to indicate "this system".

An example of a complete NET for a router is

```
49.0001.1921.6800.1002.00.
```

IS-IS has a two types of router:

- Level-1 (L1) for the network core or backbone
- Level-2 (L2) for the network edge

A router can be a type L1 and/or L2 router. For small- to medium-sized networks (max. 400 routers), the use of L2 type routers is sufficient.

Basis IS-IS Configuration

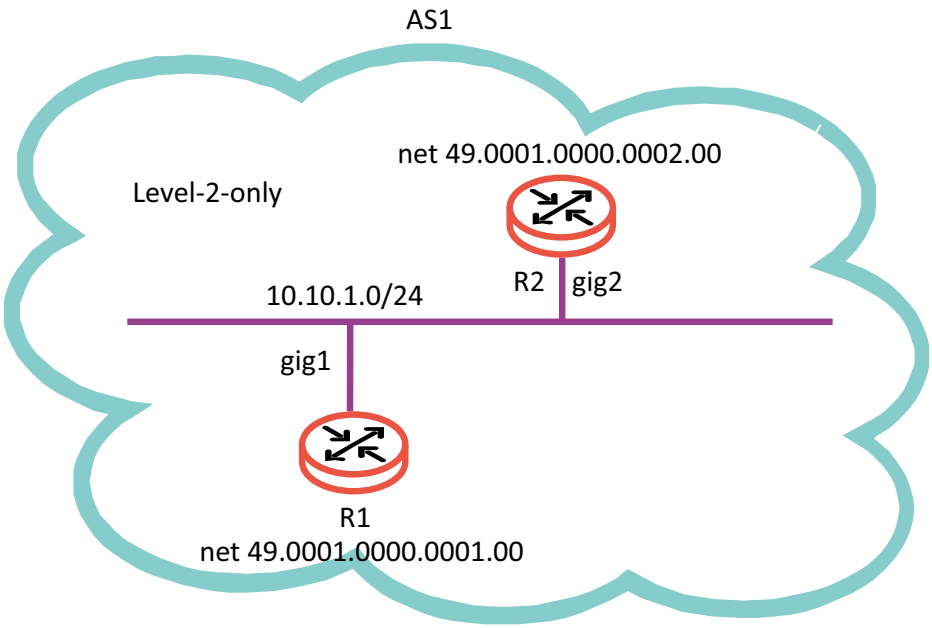
The following basic IS-IS configurations are described:

- “Enable IS-IS on an Interface” on page 390
- “Set Priority” on page 391
- “Redistribute Routes into IS-IS” on page 393
- “Configure Metric” on page 394
- “L1-L2 Area Routing with Single Instance” on page 397
- “Route Leaking” on page 402
- “Route Summarization” on page 403
- “Configure IS-IS Distance” on page 404
- “Configure Overload Bit” on page 407
- “Configure Passive Interface” on page 410

Enable IS-IS on an Interface

This example shows the minimum configuration required for enabling IS-IS on an interface. R1 and R2 are two routers in the ws instance connecting to the network 10.10.10.0/24. After enabling IS-IS on an interface, create a routing instance, and specify the Network Entity Title (NET). IS-IS explicitly specifies a NET to begin routing. NET is comprised of the area address and the system ID of the router.

Figure 103 Basic IS-IS Topology



To configure IS-IS, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws for area 49.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface ws.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **ip router isis**
- **net**
- **router isis**

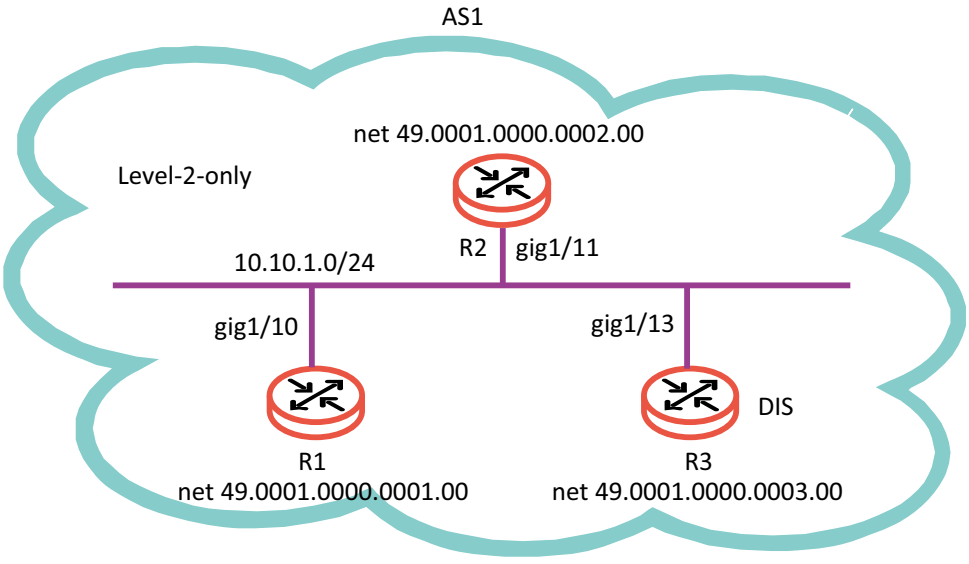
To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show isis topology**

Set Priority

This example describes how to set the priority for an interface. Set a high priority for a router to make it the Designated IS (DIS). Router R3 is configured to have a priority of 70, this is higher than the default priority (64) of R1 and R2. This makes R3 the DIS.

Figure 104 Set IS-IS Priority



To set IS-IS priority, do the following:

1. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.

`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# isis priority 70`	Specify the router priority to a higher priority (70) to make R3 the designated IS (DIS).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R1:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R2:

`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **ip router isis**
- **net**
- **router isis**
- **isis priority**

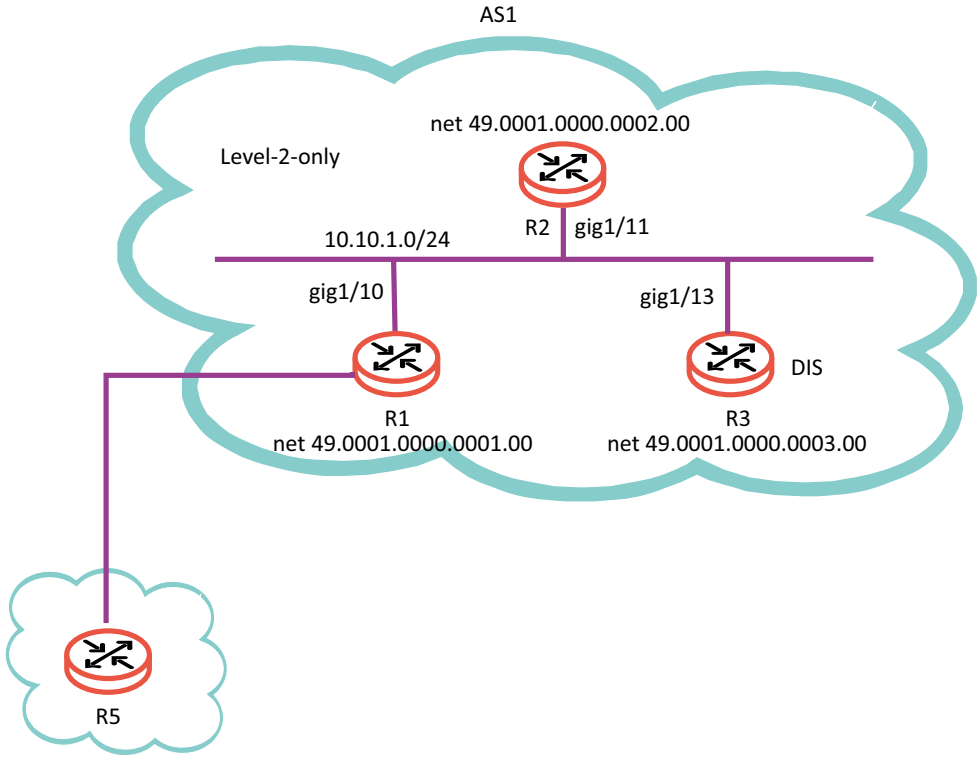
To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show isis topology**

Redistribute Routes into IS-IS

In this example, the configuration causes BGP routes to be imported into the IS-IS routing table, and advertised into the ws instance.

Figure 105 Redistribute Routes Into IS-IS



To redistribute the routes into IS-IS, configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this area (ws) as level-2-only.
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config-router)# redistribute bgp`	Specify redistributing routes from other routing protocol (BGP) into IS-IS.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **ip router isis**
- **redistribute**
- **is-type**
- **router isis**

To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show isis topology**
- **show ip isis route**
- **show ip route**

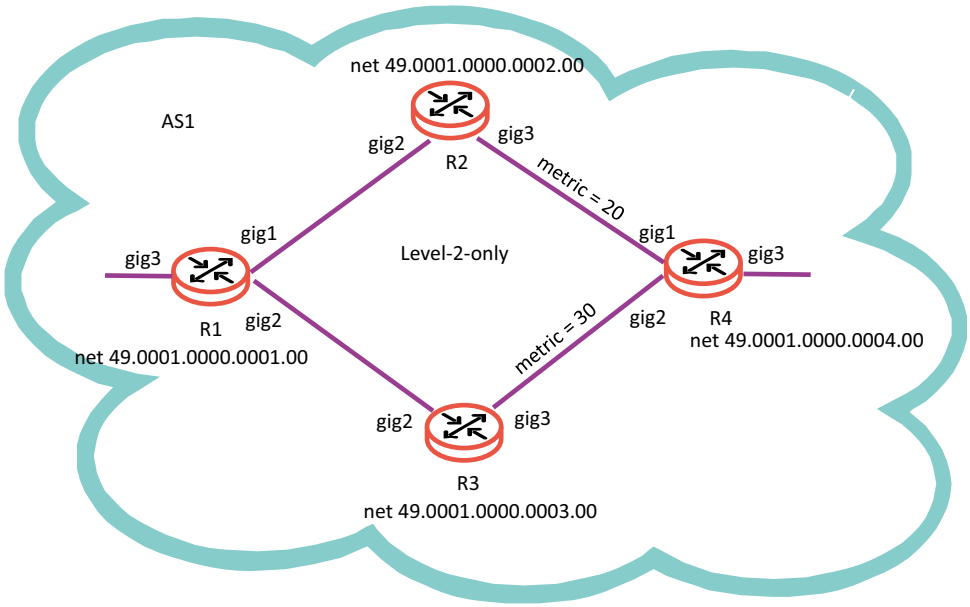
Configure Metric

You can make a route the preferred route by changing its metric. In this example, the cost has been configured to make R2 the next hop for R1 to reach R4.

The default metric for each interface is 10. Interface eth2 on R2 has a metric of 20, and Interface eth2 on R3 has a metric of 30. The total cost to reach R4 via R2 and R3 is computed as follows:

- R2: 10+20 = 30
- R3: 10+30 = 40

Figure 106 Configure IS-IS Metric



In this topology, R1 chooses R2 as its next hop for destination 10.10.14.0/24.

To configure the IS-IS metric, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig3`	Specify the interface (gig3) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig3`	Specify the interface (gig3) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# isis metric 20`	Set the value of IS-IS metric (on gig3) to 20.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig3`	Specify the interface (gig3) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# isis metric 30`	Set the value of IS-IS metric (on gig3) to 30.

`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

4. Configure router R4:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0001.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **ip router isis**
- **net**
- **isis metric**
- **router isis**

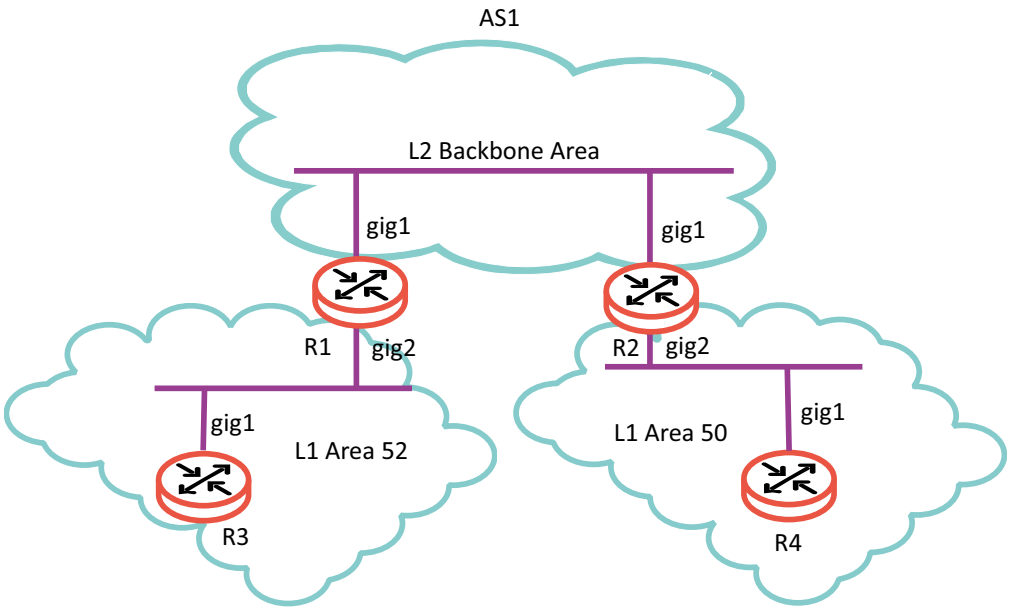
To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show isis topology**
- **show ip isis route**
- **show ip route**

L1-L2 Area Routing with Single Instance

IS-IS supports a two-level hierarchy for handling and scaling the functionality of large networks. The Level-1 (L1) area is mainly for Leaf networks, and the Level-2 (L2) area is the backbone area connecting Level-1 areas.

Figure 107 Single-Instance L1-L2 Area Routing



In this example, R3 and R4 are configured as Level-1 routers, and reside in the Level-1 area. R1 and R2 are configured as Level-1-2 routers, and connect these two Level-1 areas with a backbone Level-2 area. You can configure Level-1-2 routers with single or multiple instances: This configuration shows the single-instance version of the Level-1-2 router.

To configure the L1-L2 routing, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis abc`	Enable IS-IS routing on the interface gig1 for instance abc.
`iBOS(config-if)# isis circuit-type level-2-only`	Set the circuit type for the interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis abc`	Enable IS-IS routing on the interface gig2 for instance abc.
`iBOS(config-if)# isis circuit-type level-1`	Set the circuit type for interface gig2 to level 1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance for instance abc.
`iBOS(config-router)# net 40.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis bb`	Enable IS-IS routing on the interface gig1 for instance bb.
`iBOS(config-if)# isis circuit-type level-2-only`	Set the circuit type for the interface gig1 to level-2 only.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis bb`	Enable IS-IS routing on interface gig2 for instance bb.
`iBOS(config-if)# isis circuit-type level-1`	Set the circuit type for interface gig2 to level 1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis bb`	Create an IS-IS routing instance for instance bb.
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis xyz`	Enable IS-IS routing on the interface gig1 for area xyz.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis xyz`	Create an IS-IS routing instance for instance xyz.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (xyz) as level-1.
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

4. Configure router R4:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis aa`	Enable IS-IS routing on the interface gig1 for instance aa.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis aa`	Create an IS-IS routing instance for instance aa.

`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (aa) as level-1.
`iBOS(config-router)# net 49.0001.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **isis circuit-type**
- **is-type**
- **ip router isis**
- **net**

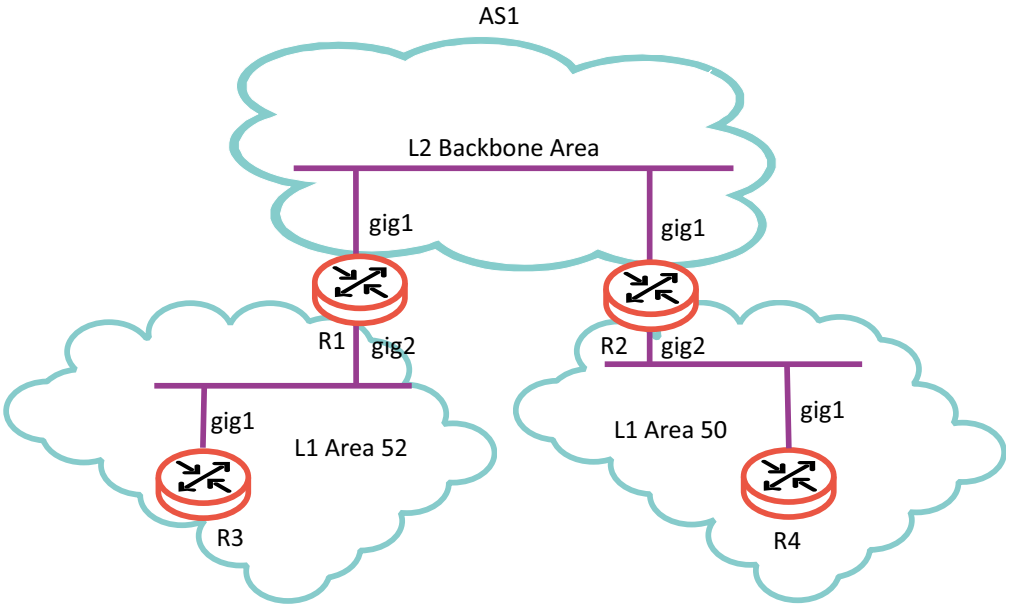
To verify the configuration, use the following commands:

- **show ip isis route**
- **show ip route**

L1-L2 Area Routing with Multiple Instances

IS-IS supports a two-level hierarchy for handling and scaling the functionality of large networks. The Level-1 (L1) area is mainly for Leaf networks, and the Level-2 (L2) area is the backbone area connecting Level-1 areas. In this example, R3 and R4 are configured as Level-1 routers, and reside in the Level-1 area. R1 and R2 are configured as Level-1-2 routers, and connect these two Level-1 areas with a backbone Level-2 area. You can configure Level-1-2 routers with single or multiple instances: This configuration shows the multiple-instance version of the Level-1-2 router.

Figure 108 Multiple-Instance L1-L2 Area Routing



To configure L1-L2 area routing with multiple distances, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis abc`	Enable IS-IS routing instance abc on interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance abc.
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this instance (abc) as level-2-only.
`iBOS(config-router)# net 49.0002.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis ccc`	Enable IS-IS routing on interface gig2 for instance ccc.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

2. Configure router R2:

`iBOS(config)# router isis ccc`	Create an IS-IS routing instance ccc.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (ccc) as level-1.
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis bb`	Enable IS-IS routing instance bb on interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis bb`	Create an IS-IS routing instance bb.
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this instance (bb) as level-2-only.
`iBOS(config-router)# net 49.0002.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis abc`	Enable IS-IS routing instance abc on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance abc.

`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (abc) as level-1.
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis xyz`	Enable IS-IS routing instance xyz on interface gig.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis xyz`	Create an IS-IS routing instance xyz.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (xyz) as level-1.
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Establish a Network Entity Title for this area, specifying the area address and the system ID.

4. Configure router R4:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ip router isis def`	Enable IS-IS routing instance def on interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis def`	Create an IS-IS routing instance def.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this instance (def) as level-1.
`iBOS(config-router)# net 49.0002.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **isis circuit-type**
- **is-type**
- **ip router isis**
- **net**

To verify the configuration, use the following commands:

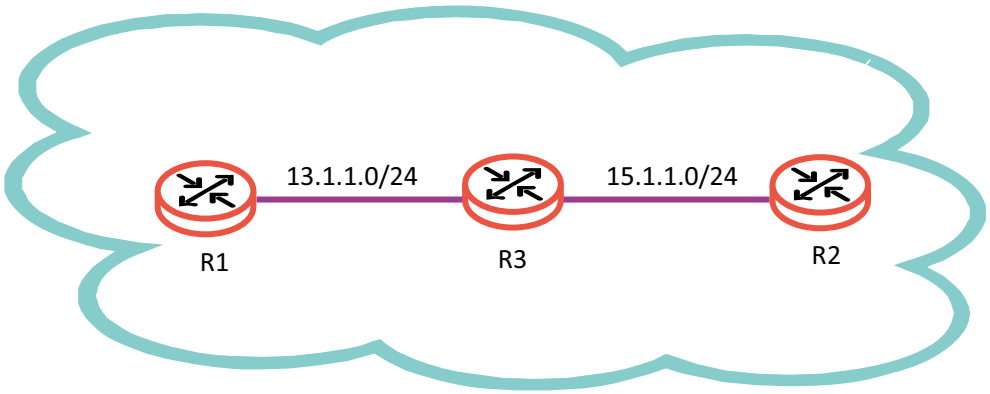
- **show ip isis route**
- **show ip route**

Route Leaking

For Level-1 (L1) routers, only level-1 routes are populated in the routing table. The L1 router has a default route to the nearest Level-1/Level-2 (L1/L2) router: This could result in sub-optimal routing in certain scenarios. Route leaking causes an L1/L2 router to advertise the level-2 routes in its database to the L1 router, thus allowing the L1 router to acknowledge the prefixes advertised by the Level-2 (L2) router. In this way, the L1 router has the ability to learn the true cost to reach other areas.

In the following example, R1 is the L1 router, R2 is the L1/L2 router doing the route leaking, and R3 is the L2 router. The following configuration is given only for R2, assuming that the adjacency with R1 and R3 are already up, and the route tables with appropriate routes are already populated.

Figure 109 Route Leaking Topology



Configure Route Leaking on router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance 1.
`iBOS(config-router)# redistribute isis level-2 into level-1`	Redistribute routes learned from level 2 into level 1 (route leaking).
`iBOS(config-router)# end`	Exit Router mode.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **router isis**
- **redistribute isis level-2 into level-1**

To verify the configuration, use the **show ip route** command.

In the example, route i ia 3.3.3.0/24 [115/30] via 15.1.1.32, eth2, 00:00:02, is the L2 route leaked by the L1/L2 router into the L1 router.

```
R1# show ip route
Codes: K - kernel, C - connected, S - static, CL - client, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2,
       ia - IS-IS inter area
       * - candidate default
```

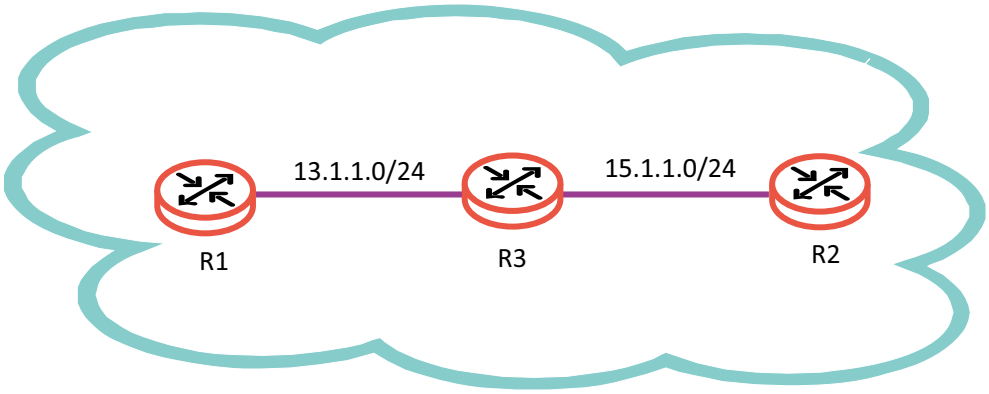
```
Gateway of last resort is 15.1.1.32 to network 0.0.0.0

i*L1    0.0.0.0/0 [115/10] via 15.1.1.32, eth2, 00:00:59
i ia    3.3.3.0/24 [115/30] via 15.1.1.32, eth2, 00:00:02
C       10.1.2.0/24 is directly connected, gig1
K       10.10.0.0/16 via 10.1.2.1, gig1
K       10.10.3.0/24 is directly connected, gig1
C       13.1.1.0/24 is directly connected, gig2
C       15.1.1.0/24 is directly connected, eth2
C       31.31.31.31/32 is directly connected, lo
C       127.0.0.0/8 is directly connected, lo
K       169.254.0.0/16 is directly connected, gig1
```

Route Summarization

Route summarization makes the routing table smaller, but still allows complete IP connectivity, if everything is configured properly.

The following example consists of a three-router topology, in which R2 is doing the summarization. In this example, R1 is the L1 router, R2 is the L1/L2 router doing the summarization, and R3 is the L2 router. The following configuration is given only for R2, assuming that the adjacencies with R1 and R3 are already up, and the route tables with the appropriate routes are already populated.

Figure 110 Route Summarization Topology



To configure route summarization on router R2, do the following:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance for area 49 (1).
`iBOS(config-router)# summary-address 20.10.3.0/6`	The prefixes, 20.10.3.0/8, 21.10.3.0/8, 22.10.3.0/8, and 23.10.3.0/8 are summarized to 20.10.3.0/6.
`iBOS(config-router)# end`	Exit Router mode.

Refer *iBOS Command Reference* for detailed descriptions of the commands **router isis** and **summary-address.**

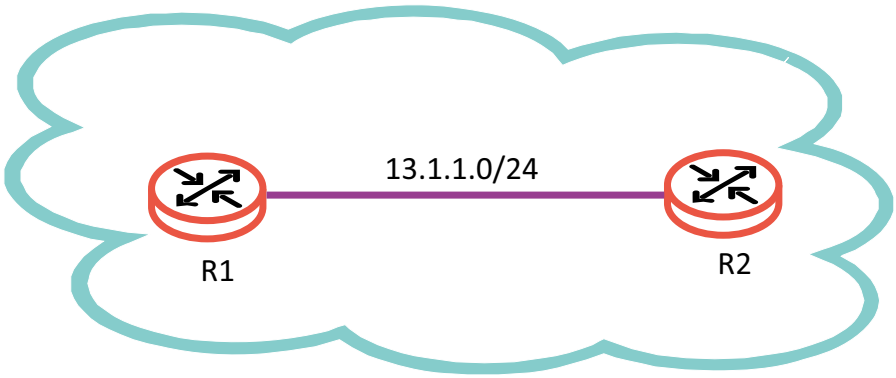
To verify the configuration, use the command **show ip route**.

Graceful Restart

ISIS provides graceful restart, in which the adjacency and routes are maintained in the routing table for the grace period. In this way, the data flow is not affected, and there is no packet loss during the restart phase.

In the following example, R1 is the L1/L2 router, and R2 is the L1/L2 restart-helper router. The following configuration is given only for R2, assuming that the adjacency with R1 is already up, and the route tables with the appropriate routes are already populated.

Figure 111 IS-IS Graceful Restart



To configure graceful restart on router R2, do the following:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router restart helper`	Configure this router as a restart helper.
`iBOS(config)# isis restart grace-period 100`	Set the grace period to 100 seconds. The restarting router should come up before 100 seconds, otherwise, the adjacency and routes will be deleted.

The IS-IS daemon in the restarting router must be manually restarted: It does not restart automatically.

Refer *iBOS Command Reference* for detailed descriptions of the commands **router restart helper** and **isis restart grace-period**.

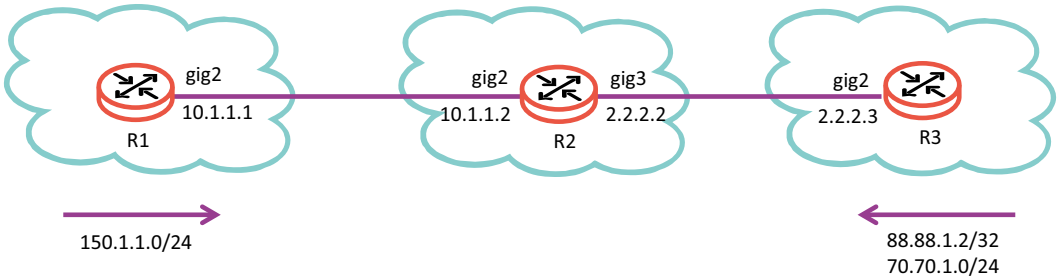
To verify the configuration, use the command **show ip route**.

Configure IS-IS Distance

Administrative distance in IS-IS can be configured for a specified source ID or for all routes.

This example shows configuring the IS-IS administrative distance for the IPv4 address family.

Figure 112 IS-IS Distance Topology



To configure IS-IS distance, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip address 10.1.1.2/24`	Assign the IP address on this interface (gig2).
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Establish a Network Entity Title (NET) for this instance, specifying the address and the system ID.
`iBOS(config-router)# redistribute static`	Redistribute the static routes.

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# access-list DIST permit 88.88.1.2/32`	Create an access list to permit the 88.88.1.2/32 route from R3.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip address 10.1.1.1/24`	Assign the IP address on this interface (gig2).
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface eth2`	Specify the interface (eth2) to configure, and enter Interface mode.
`iBOS(config-if)# ip address 2.2.2.2/24`	Assign the IP address on this interface (eth2).
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Specify the NET address.
`iBOS(config-router)# distance 100`	Configure the administrative distance for all routes received from R1 and R2.
`iBOS(config-router)# distance 20 0000.0000.0001`	Configure the administrative distance for all routes received from R1. This command overwrites the applied distance, 100, and will apply distance 20 for all routes received from R1.
`iBOS(config-router)# distance 30 0000.0000.0003 DIST`	Configure the distance, 30, to the route, 88.88.1.2/32, received from R3. All other routes from R3 (for example, 70.70.1.0/24) will have the distance applied as 100. If the distance, 100, is not configured, all other routes will have a default distance of 115.

3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ip address 2.2.2.3/24`	Assign the IP address on this interface (gig2).
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Specify the NET address.
`iBOS(config-router)# redistribute static`	Redistribute the static routes.

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **net**
- **router-isis**
- **ip router isis**
- **redistribute static**
- **distance**

To verify the configuration, use the following commands:

- **show ip isis route**
- **show ip route**
- **show ip route database**
- **show ip protocols isis**
- **show clns neighbours**

Configure Overload Bit

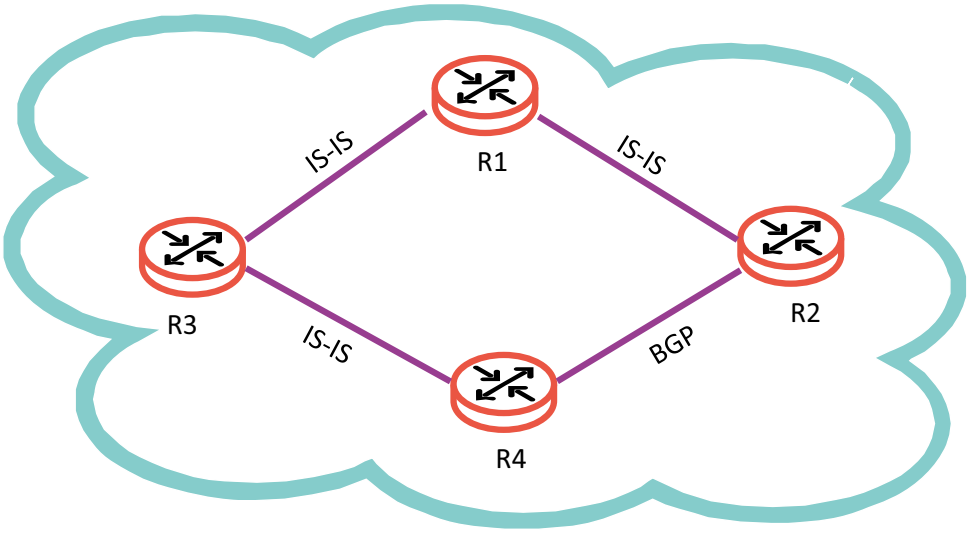
The expanded use of the overload bit in IS-IS is useful to Internet Service Providers (ISPs) who run both BGP and IS-IS to avoid certain black-hole scenarios. The overload bit feature allows configuration of a router to disable the overload bit automatically when BGP has converged.

IS-IS disables the overload bit if BGP is not converged in ten minutes, and the wait-for-bgp option is configured.

When the set overload-bit wait-for-bgp is configured after restart (for example, using the set-overload-bit on-startup wait-for-bgp or set-overload-bit on-startup <5-86400> commands), the IS-IS router advertises the overload bit to all its neighbours, informing them not to route traffic through this router, except for the traffic destined to directly-connected networks of this router.

Whenever an interface is configured as no set-overload-bit (using the no set-overload-bit command), the router does not advertise or set the overload bit on startup.

Figure 113 Configure Overload Bit



To configure the Overload Bit, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R2).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.

`iBOS(config)# interface eth2`	Specify the interface (eth2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2 (connected to R3).
`iBOS(config-if)# isis metric 50`	Increase the value of IS-IS metric from R1-R3.

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 100`	Create a BGP routing instance (100).
`iBOS(config-router)# neighbour 10.1.1.2 remote-as 2`	Specify the neighbour's IP address (10.1.1.2) and the ASN value of the neighbour (2).
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R1).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2 (connected to R4).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance 1.
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Define the NET address.
`iBOS(config-router)# set-overload-bit on-startup wait-for-bgp`	Configure IS-IS to set the overload bit until BGP is completely converged (maximum 600 seconds) only on restart.
`iBOS(config-router)# set-overload-bit on-startup wait-for-bgp suppress external`	Configure IS-IS to suppress external routes until the configured value (in seconds) only on startup.
`iBOS(config-router)# set-overload-bit on-startup wait-for-bgp suppress interlevel`	Configure IS-IS to suppress inter-level routes until the configured value (in seconds) only on startup.

3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.

`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R4).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2 (connected to R1).

4. Configure router R4:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router bgp 100`	Create a BGP routing instance (100).
`iBOS(config-router)# neighbour 10.1.1.1 remote-as 1`	Specify the neighbour's IP address (10.1.1.1) and the ASN value of the neighbour (1).
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0004.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R3).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R2).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **net**
- **router-isis**
- **ip router isis**
- **isis metric**
- **neighbour remote-as**
- **set-overload-bit on-startup wait-for-bgp**
- **set-overload-bit on-startup wait-for-bgp suppress external**
- **set-overload-bit on-startup wait-for-bgp suppress interlevel**

To verify adjacency and other protocol-related details, use the following commands:

- **show isis database**
- **show ip bgp summary**
- **show clns neighbours**

Configure Passive Interface

In ISP and large enterprise networks, many of the distribution routers have more than 200 interfaces. Before the Default Passive-Interface feature, there were two possibilities for obtaining routing information from all of these interfaces:

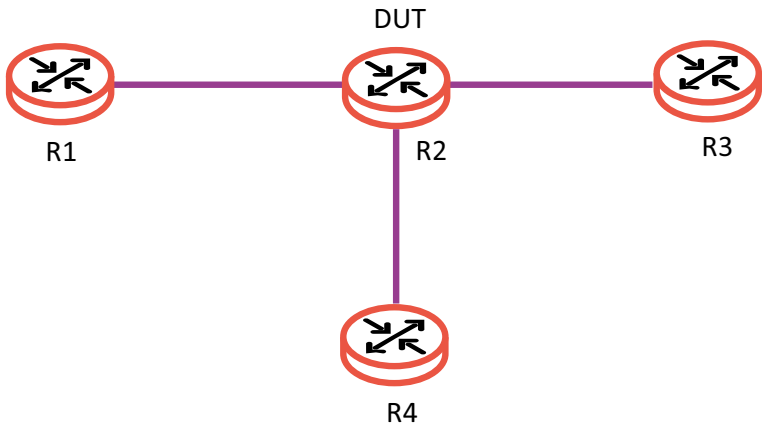
- Configure a routing protocol on the backbone interfaces and redistribute connected interfaces.
- Configure the routing protocol on all interfaces and manually set most of them as passive, which was time consuming.

The solution to this problem was to configure the routing protocol on all interfaces and manually set the passive-interface command on the interfaces where adjacency was not desired. In certain networks, this meant coding 200 or more passive-interface statements. With the Default Passive Interface feature, this problem is solved by allowing all interfaces to be set as passive by default using a single passive-interface default command, then configuring individual interfaces in which adjacencies are desired using the no passive-interface command.

When this feature is used, the following caveats apply:

- When a specific interface is configured as passive using the **passive-interface** command:
  - the interface loses its adjacency on that interface, for example, gig2.
  - the interface (gig2) is still advertised by other IS-IS speaking interfaces to their neighbours.
- When a specific interface is configured as not passive using the **no passive-interface** command:
  - the interface is IS-IS disabled and must be enabled using the ip router isis command (for example, ip router isis 1).
  - the interface (for example, gig2) is not advertised by other IS-IS speaking interfaces to their neighbours.
- When an interface is configured with the **passive-interface** command:
  - all interfaces lose their adjacency, except the interface with the higher index number. (For example: If gig1, gig2, gig3, and gig4 are the router interfaces, gig4 has the highest index number.)
  - all interfaces are advertised by the active IS-IS speaking interface to its neighbours.
- When an interface is configured with the **no passive-interface** command:
  - all interfaces are IS-IS disabled, except the interface that was active, and all interfaces must enable IS-IS on these interfaces using the ip router isis command (for example, ip router isis 1).
  - all interfaces are not advertised by the active IS-IS speaking interface to its neighbours. Figure 114 shows a passive-interface configuration example.

Figure 114 IS-IS Passive Interface



To configure the passive-interface, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0001.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R2).

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R1).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# interface eth2`	Specify the interface (eth2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2 (connected to R3).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# interface eth3`	Specify the interface (eth3)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface eth2 (connected to R4).
`iBOS(config-if)# exit`	Exit Interface mode and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0002.00`	Define the NET address.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config-router)# passive-interface gig2`	Configure the gig2 interface as passive.
`iBOS(config-router)# passive-interface`	Configure all interfaces as passive, except one (interface with the higher index number).

3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0003.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R2).

4. Configure router R4:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0000.0004.00`	Define the NET address.
`iBOS(config-router)# exit`	Exit Router mode and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure and enter Interface mode.
`iBOS(config-if)# ip router isis 1`	Enable IS-IS routing on interface gig2 (connected to R2).

Refer *iBOS Command Reference* for detailed descriptions of the following commands:

- **net**
- **router-isis**
- **ip router isis**
- **net**
- **passive-interface**

To verify adjacency and other protocol-related details, use the following commands:

- **show isis database details**
- **show clns neighbours**

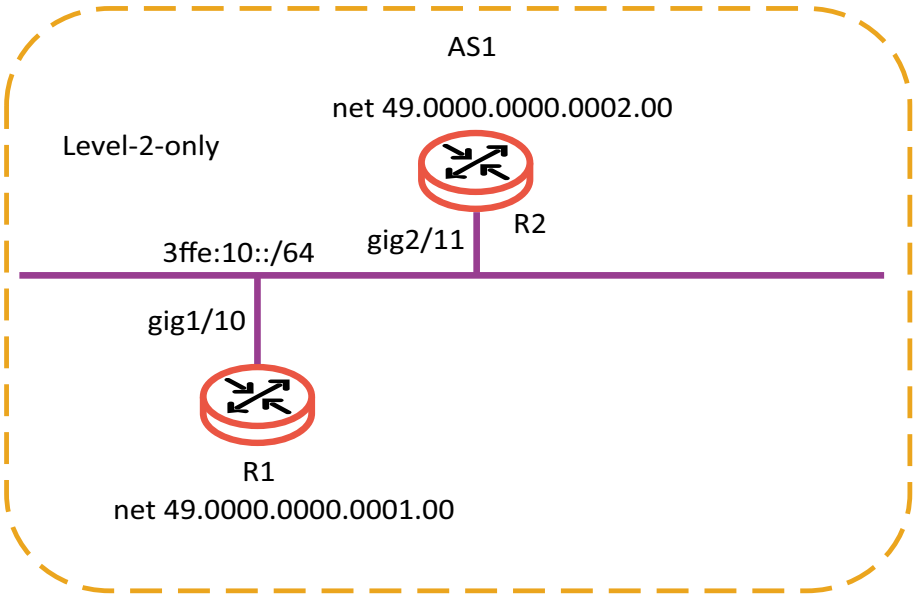
IS-ISv6 Configuration

This section contains configuration examples for IS-IS using IPv6 routing. To avoid repetition, some common commands, such as **configure terminal**, have not been listed in the examples. Common commands are defined in “About the iBOS Command Line Interface” on page 18.

Basic IS-ISv6 Configuration Example

This example shows the minimum configuration required for enabling IS-IS (IPv6) on an interface. R1 and R2 are two routers in the ws instance connecting to the network 3ffe:10::/64. After enabling IS-IS on an interface, create a routing instance, and specify the Network Entity Title (NET). IS-IS explicitly specifies a NET to begin routing. NET is comprised of the area address and the system ID of the router.

Figure 115 Enable IS-IS IPv6



To enable IS-IS IPv6, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and system ID.

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS IPv6 routing instance ws.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **net**
- **router isis**

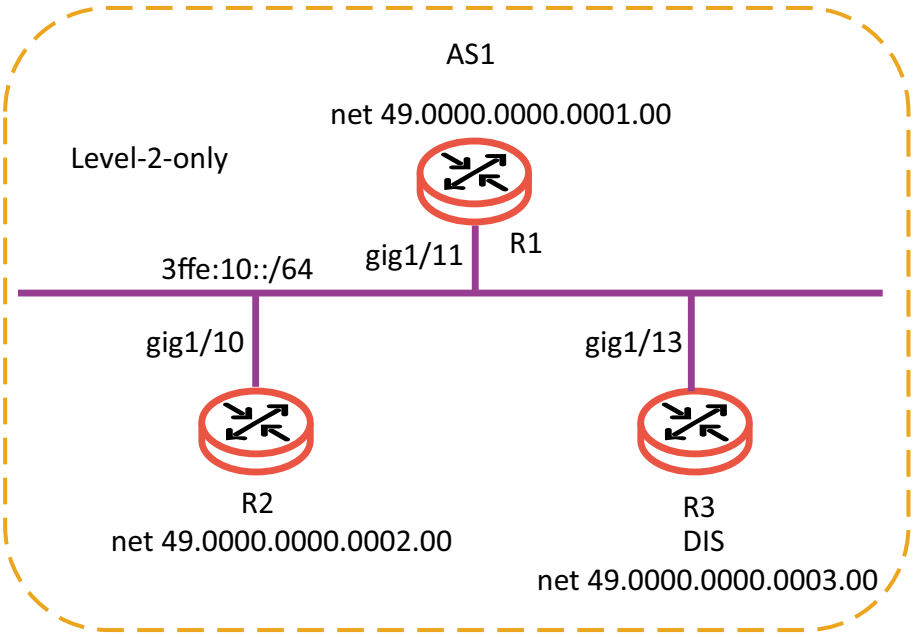
To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show ipv6 isis topology**

Set IS-ISv6 Priority

This example shows setting the priority for an interface. Set a high priority for a router to make it the Designated IS (DIS). Router R3 is configured to have a priority of 70; this is higher than the default priority (60) of R1 and R2. This makes R3 the DIS.

Figure 116 Set IS-ISv6 Priority



To configure IS-ISv6 priority, do the following:

1. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# isis priority 70`	Specify the router priority to a higher priority (70) to make R3 the designated IS (DIS).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R1:

`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0000.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R2:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **net**
- **router isis**
- **isis priority**

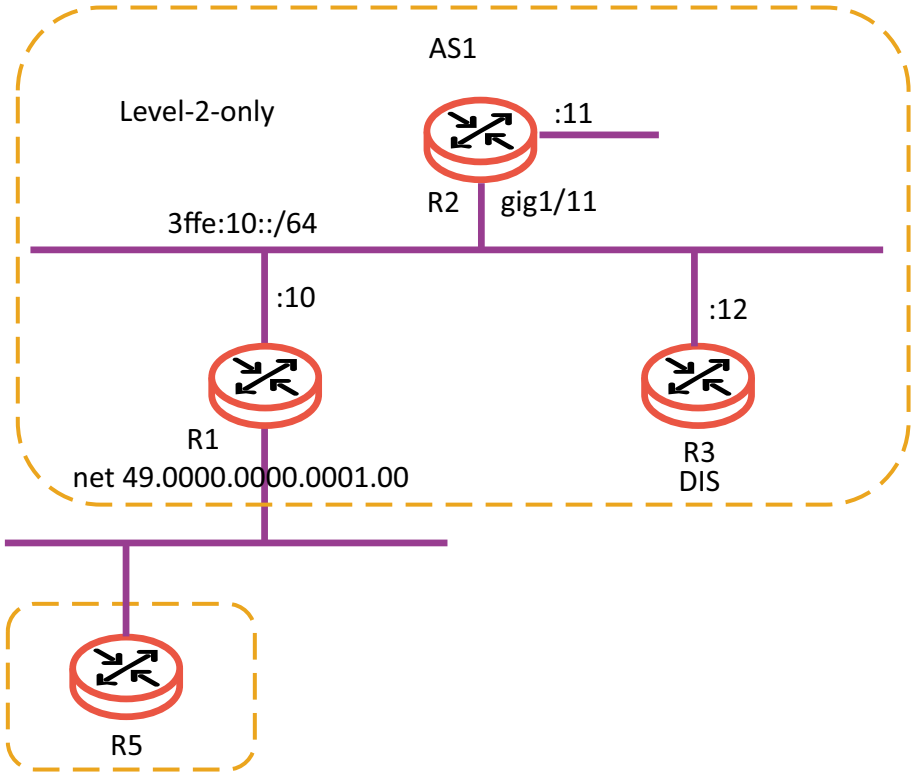
To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show ipv6 isis topology**

Redistribute Routes into IS-ISv6

In this example, the configuration causes BGP routes to be imported into the IS-IS routing table, and advertised into the ws instance.

Figure 117 Redistribute Routes Into IS-ISv6



To redistribute routes into IS-ISv6, configure the router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance for area 49 (ws).
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this area (ws) as level-2-only.
`iBOS(config-router)# net 49.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and system ID.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config-router)# address-family ipv6`	Enter Address-Family mode.
`iBOS(config-router-af)# redistribute bgp`	Specify redistributing routes from the other routing protocol (BGP) into IS-IS.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **redistribute**
- **is-type**
- **router isis**
- **address-family**

To verify the configuration, use the following commands:

- **show clns neighbours**
- **show isis database**
- **show ipv6 isis topology**
- **show ipv6 isis route**
- **show ipv6 route**

Configure Metric for IS-ISv6

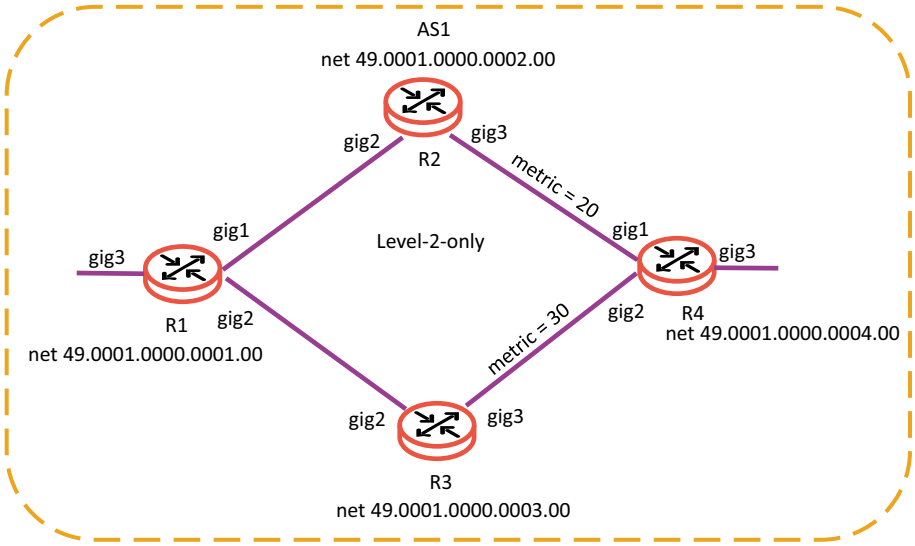
Make a route the preferred route by changing its metric. In this example, cost has been configured to make R2 the next hop for R1.

The default metric on each interface is 10. Interface eth2 on R2 has a metric of 20, and Interface eth2 on R3 has a metric of 30. The total cost to reach 3ffe:10::/64 (R4) through R2 and R3:

- R2: 10+20 = 30
- R3: 10+30 = 40

In this topology, R1 chooses R2 as its next hop for destination 10.10.14.0/24.

Figure 118 IS-ISv6 Metric



To configure metric for IS-ISv6, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing on an interface for area 49 (ws).
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS(config)# interface gig3`	Specify the interface (gig3)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# isis metric 20`	Set the value of IS-IS metric (on gig2) to 20.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws on an interface.
`iBOS(config-if)# isis metric 30`	Set the value of IS-IS metric (on gig2) to 30.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.
`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

4. Configure router R4:

`iBOS(config)# interface gig1`	Specify the interface (gig1) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis ws`	Enable IS-IS routing instance ws.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ws`	Create an IS-IS routing instance ws.

`iBOS(config-router)# is-type level-2-only`	Configure instance ws as level-2-only routing.
`iBOS(config-router)# net 49.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **net**
- **isis metric**
- **router isis**

To verify the configuration, use the following commands:

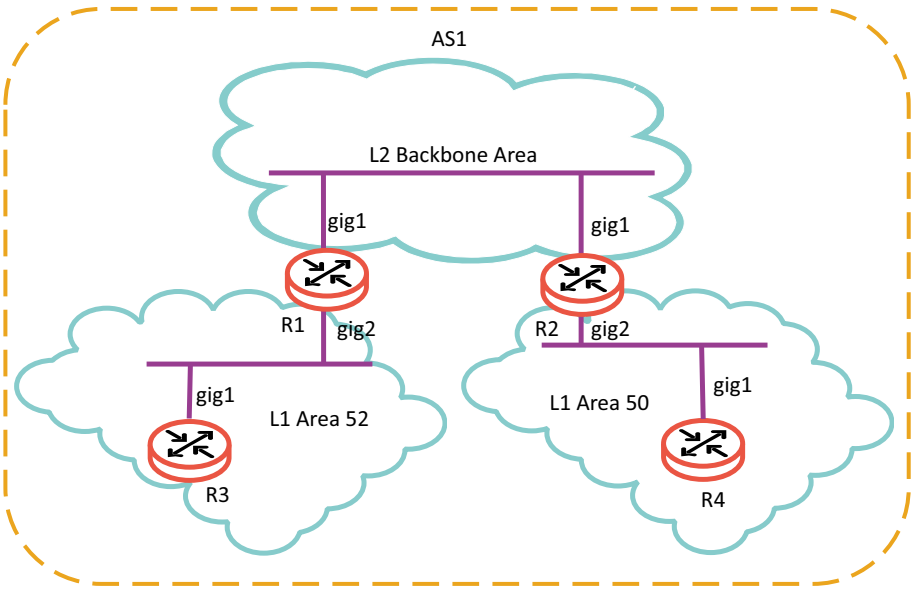
- **show clns neighbours**
- **show isis database**
- **show ipv6 isis topology**
- **show ipv6 isis route**
- **show ipv6 route**

L1-L2 Area IPv6 Routing

IS-ISv6 supports a two-level hierarchy for handling and scaling the functionality of large networks. The Level-1 (L1) area is mainly for Leaf networks, and the Level-2 (L2) area is the backbone area connecting Level-1 areas. In this example, R3 and R4 are configured as L1 routers, and reside in the Level-1 area. R1 and R2 are configured as L1-L2 routers, and connect these two Level-1 areas with a backbone Level-2 area.

You can configure L1-L2 routers with single or multiple instances.

Figure 119 L1-L2 Area Routing IPv6



L1-L2 IPv6 Routing with Single Instance

This configuration example shows the single-instance L1-L2 router for Figure 119.

To configure single instance IS-ISv6 area routing, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis abc`	Enable IS-IS routing on the interface gig1 for area abc.
`iBOS(config-if)# isis circuit-type level-2-only`	Set the circuit type for the interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis abc`	Enable IS-IS routing on the interface gig2 for area abc.
`iBOS(config-if)# isis circuit-type level-1`	Set the circuit type for the interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance for area abc.
`iBOS(config-router)# net 52.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis bb`	Enable IS-ISv6 routing on the interface gig1 for area bb.
`iBOS(config-if)# isis circuit-type level-2-only`	Set the circuit type for the interface gig1.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis bb`	Enable IS-ISv6 routing on the interface gig2 for area bb.
`iBOS(config-if)# isis circuit-type level-1`	Set the circuit type for the interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis bb`	Create an IS-IS routing instance for area bb.
`iBOS(config-router)# net 50.0000.0000.0002.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis xyz`	Enable IS-ISv6 routing on the interface gig1 for area xyz.

`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis xyz`	Create an IS-IS routing instance for area xyz.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (xyz) as level-1.
`iBOS(config-router)# net 52.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

4. Configure router R4:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis aa`	Enable IS-ISv6 routing on the interface gig1 for area aa.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis aa`	Create an IS-IS routing instance for area aa.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (aa) as level-1.
`iBOS(config-router)# net 50.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **isis circuit-type**
- **isis-type**
- **net**

To verify the configuration, use the commands **show ipv6 isis route** and **show ipv6 route.**

L1-L2 Area IPv6 Routing with Multiple Instances

This configuration example shows the multiple-instance L1-L2 router for Figure 119.

To configure multiple instance L1-L2 area IPv6 routing, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis abc`	Enable IS-IS routing on the interface gig1 for area abc.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance for area abc.
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this area (abc) as level-2-only.
`iBOS(config-router)# net bb.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2)to configure, and enter Interface mode.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config-if)# ipv6 router isis ccc`	Enable IS-IS routing on the interface gig2 for area ccc.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis ccc`	Create an IS-IS routing instance for area ccc.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (ccc) as level-1.
`iBOS(config-router)# net 52.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

2. Configure router R2:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis bb`	Enable IS-IS routing on the interface gig1 for area bb.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis bb`	Create an IS-IS routing instance for area bb.
`iBOS(config-router)# is-type level-2-only`	Establish the IS level for this area (bb) as level-2-only.
`iBOS(config-router)# net bb.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.
`iBOS(config-router)# exit`	Exit Router mode, and return to Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis abc`	Enable IS-IS routing on the interface gig2 for area abc.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis abc`	Create an IS-IS routing instance for area abc.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (abc) as level-1.
`iBOS(config-router)# net 52.0000.0000.0001.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

3. Configure router R3:

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis xyz`	Enable IS-IS routing on the interface gig1 for area xyz.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis xyz`	Create an IS-IS routing instance for area xyz.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (xyz) as level-1.
`iBOS(config-router)# net 52.0000.0000.0003.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

4. Configure router R4

`iBOS(config)# interface gig1`	Specify the interface (gig1)to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 router isis aa`	Enable IS-IS routing on the interface gig1 for area aa.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis aa`	Create an IS-IS routing instance for area aa.
`iBOS(config-router)# is-type level-1`	Establish the IS level for this area (aa) as level-1.
`iBOS(config-router)# net 50.0000.0000.0004.00`	Establish a Network Entity Title for this instance, specifying the area address and the system ID.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **isis circuit-type**
- **isis-type**
- **net**

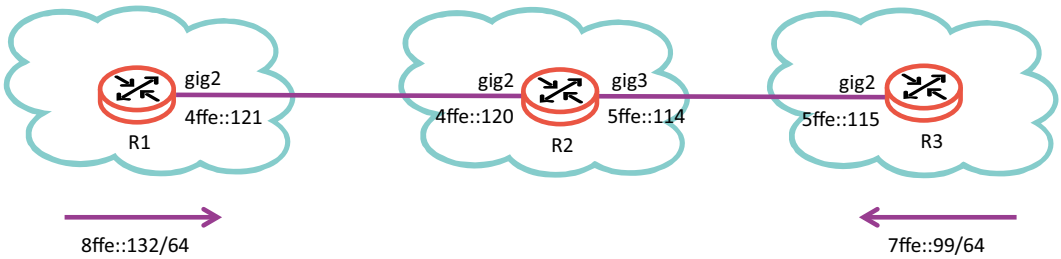
To verify the configuration, use the commands **show ipv6 isis route** and **show ipv6 route**.

Configure IS-IS IPv6 Distance

Administrative distance can be configured for an IS-IS IPv6 address family.

This example shows configuring the IS-IS administrative distance for the IPv6 address family.

Figure 120 IS-ISv6 Distance



To configure IS-ISv6 distance, do the following:

1. Configure router R1:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 address 4ffe::121/24`	Assign the IPv6 address on this interface (gig2).
`iBOS(config-if)# ipv6 router isis 1`	Enable IS-ISv6 routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.

`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0001.00`	Establish a Network Entity Title (NET) for this instance, specifying the address and the system ID.
`iBOS(config-router)# address-family ipv6`	Enter IPv6 Address-Family mode.
`iBOS(config-router-af)# redistribute static`	Redistribute the static routes.

2. Configure router R2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 address 4ffe::120/24`	Assign the IPv6 address on this interface (gig2).
`iBOS(config-if)# ipv6 router isis 1`	Enable IS-ISv6 routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# interface eth2`	Specify the interface (eth2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 address 5ffe::114/24`	Assign the IPv6 address on this interface (eth2).
`iBOS(config-if)# ipv6 router isis 1`	Enable IS-ISv6 routing on interface eth2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0002.00`	Specify the NET address.
`iBOS(config-router)# address-family ipv6`	Enter IPv6 Address-Family mode.
`iBOS(config-router)# distance 60`	Configure the administrative distance in IPv6 Address-Family mode.

3. Configure router R3:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface gig2`	Specify the interface (gig2) to configure, and enter Interface mode.
`iBOS(config-if)# ipv6 address 5ffe::115/24`	Assign the IPv6 address on this interface (gig2).
`iBOS(config-if)# ipv6 router isis 1`	Enable IS-ISv6 routing on interface gig2.
`iBOS(config-if)# exit`	Exit Interface mode, and return to Configure mode.
`iBOS(config)# router isis 1`	Create an IS-IS routing instance (1).
`iBOS(config-router)# net 49.0001.0000.0003.00`	Specify the NET address.

`iBOS(config-router)# address-family ipv6`	Enter IPv6 Address-Family mode.
`iBOS(config-router)# redistribute static`	Redistribute the static routes.

Refer *iBOS Command Reference* for detailed descriptions for the following commands:

- **ipv6 router isis**
- **net**
- **distance**
- **router isis**
- **address-family ipv6 unicast**
- **redistribute static**

To verify the configuration, use the following commands:

- **show clns neighbours**
- **show ipv6 route**
- **show ipv6 route database**
- **show ipv6 isis route**
- **show ipv6 protocols isis**

Access-lists

Packet filtering is achieved using access-lists, a versatile tool in iBOS.

Understanding iBOS Access-lists

Access-lists in iBOS mirror the industry standard. The access-list basically consists of three parts.

The first part uniquely identifies each row in an access list, a sequence number, action to take and the protocol layer and type to filter.

The second part consist of the matching statements for the packet source or origin. This part matches on the source IP address/IP subnet, source protocol port number etc.

The third part is the matching statement for the packet destination or nexthop. This part matches on the destination IP address/IP subnet, destination protocol port number etc.

An access-list can consist of multiple rows. Packets are checked against each row of the access-list in sequence number order. The first row to match the packet also takes effect. If the access-list has action permit the packet is allowed through the filter, if the action is deny the packet is dropped. The name used on each access-list row combines the access-list into an access-group. When a configuration command requires an access-list to be specified it is really the name of the access-group that must be entered.

All access-lists have an implicit deny any any as final row. This means that if a packet is matched against all configured rows in the access-list and no match is found the packet will be dropped. The assumption is that if an access-list is applied on the packet stream, there is something to protect. If communication is not explicitly permitted it is not allowed through.

Each access-list row indicates on which protocol layer to perform the source and destination filtering. Keywords such as **ip**, **icmp**, **udp** and **tcp** defines which header of the data packet to analyse in the access-list.

There are two types of access-lists in iBOS. The normal **access-list** and the **interface access-list.** They work exactly the same way, but the interface access-list can be defined in interface context.

Access-list Configuration Example

This section provides a set of access-list examples:

- Access-list for Telnet Access Control
- Access-list with Shortcut
- Access list with DSCP

In "Access-list for Telnet Access Control" on page 428, the access-list includes two rows. The second line of the example, which is the first row to be processed, matches on the IP header where the source IP address is 10.11.12.13. There is no destination part on this row, but that is not required for the intended use and therefore it can be skipped. The third line also matches on the IP header, but in this case the source IP address can be any address in the 10.0.0.0/24 subnet. The access-list is used with the iBOS telnet daemon that accept incoming telnet connections to the command line interface. By applying the configuration above, iBOS will verify that any telnet connection attempt to the system is only allowed from the IP addresses matching either of the two rows in the access-list. The implicit **deny any any** at the end of the list will prevent any other IP address from connecting to the configured system.

Example 85 Access-list for Telnet Access Control

```
access-list telnet-acl
  seq 10 permit ip host 10.11.12.13
  seq 20 permit ip 10.0.0.0/24

telnetd access-group telnet-acl
```

In "Access-list with Shortcut" on page 428, the access-list **`cust-in`** is created as a separate context. This enhances the readability of the access-list when displaying the configuration, but offers no functionality difference. This access-list uses the **shortcut** keyword to treat some specific packets differently in the system. In a network where the **ip strict-clients** command is used to allow traffic only from clients that have been identified and approved. The client must first be able to request an IP address using DHCP. The DHCP DISCOVER message occurs before the client has been identified and before the **ip strict-clients** command would not allow traffic from the client. In this example, the shortcut feature that bypasses the **ip strict-clients** command is applied on any packet matching the first two rows of the access-list. They match DHCP traffic sent on standard UDP ports and therefore will allow the client to obtain an IP address.

Any traffic that is not DHCP will match on the third line of the access-list. This line permits any ip traffic meaning that only approved clients on fastethernet1 will be able to send traffic, but if they do, any traffic as long as it is IP is accepted and forwarded by the system.

Example 86 Access-list with Shortcut

```
interface fastethernet1
 ip access-group cust-in in
 ip strict-clients

access-list cust-in
 seq 10 permit udp any 68 any 67 shortcut
```

```
 seq 15 permit udp any 67 any 67 shortcut
 seq 20 permit ip any any
```

Also note that the **ip access-group** command defines that the access-lists in cust-in should be applied on packets received by the interface (**in**). The **ip access-group** configuration command allows access-lists to be applied both on packets in and out of the interface. It is possible to define different access-lists for each direction.

In "Access list with DSCP" on page 429, the access list uses the Differentiated Services Code Point (DSCP) value for QoS on an interface. The access-list can either be attached directly to an interface (step 2), or indirectly using a policy-map (see step 3).

Example 87 Access list with DSCP

```
#step 1
#configure ACL with dscp options
access-list DSCP
seq 110 permit icmp any any dscp 1
seq 120 permit icmp any any dscp 8
seq 130 deny icmp any any dscp 16
seq 140 permit icmp any any dscp 56
seq 150 deny icmp any any dscp 63
seq 160 deny icmp any any
seq 210 permit tcp any any dscp 1
seq 220 permit tcp any any dscp 8
seq 230 deny tcp any any dscp 16
seq 240 permit tcp any any dscp 56
seq 250 deny tcp any any dscp 63
seq 260 deny tcp any any
seq 310 permit udp any any dscp 1
seq 320 permit udp any any dscp 8
seq 330 deny udp any any dscp 16
seq 340 permit udp any any dscp 56
seq 350 deny udp any any dscp 63
seq 360 deny udp any any
seq 410 deny ip any any

#step 2
#apply acl DSCP to specific interface
interface tengigabitethernet0/3
remote vid 77
ip access-group DSCP in
ip address 10.1.1.1 255.255.255.0
no shutdown

#step 3
#apply acl DSCP using policy-map to specific interface
interface tengigabitethernet0/3
policy DSCP access-group DSCP in
policy-map 0.0.0.0/0 seq 10 policy DSCP
remote vid 77
ip address 10.1.1.1 255.255.255.0
no shutdown
```

Traffic Control

Traffic control is achieved using policy maps. A policy is a definition of actions to enforce on IP traffic sourced or destinated for a specific IP address or subnet.

The policy consists of a set of configuration statements in the interface context of the interface where the host or subnet affected by the policy is connected. To activate the policy, it is mapped to an IP prefix that can be as small as a single host. Any traffic sourced from or destinated to a policy-mapped prefix is compared to the conditions and actions set in the policy. When the policy-map is defined it can optionally be configured with an access-list name. Entries in the access-list can narrow the source/destinations affected by the policy. The default is to match any traffic sent to or from the IP prefix listed, but if an access-group is specified, the match must be from the prefix to an entry in the access-list or from an entry in the access-list to the prefix in order for the policy to be enforced.

Policy classification is also possible on other aspects of the traffic than the IP prefix, for example port or VLAN can also be used. The following is described:

- "Rate-limiting" on page 432
- "VLAN Translation" on page 432
- "Policy Map EtherTypes" on page 434
- "Load Balancing" on page 435
- "Policy Routing" on page 435

Rate-limiting

When a packet has been analysed and found to be affected by a policy, one of the possible actions is to rate-limit the traffic to/from the prefix. iBOS has several different methods for rate-limiting of traffic depending on the router hardware.

The purpose of rate-limiting is to reduce the maximum allowed throughput of traffic on an interface that is physically capable of higher throughput. This is achieved by controlling the rate of which packets are sent out on the interface or forwarded when received on the interface. Different queueing mechanisms and different actions to take if the rate of packets exceed the allowed threshold differ the methods from each other.

The methods available are described with the **policy conditioner** command.

Policy Mapping for VLAN and Ports

It is possible to configure a policy per port and per VLAN, as well as per MAC-address.

For more details, refer configuration commands:

- **policy-map port**
- **policy-map vid**

Automatic Creation of Policy-Map for Snooped DHCP Clients

A configuration command to automatically create policy-maps for a snooped DHCP client is available.The command for the auto-policy-map configuration is **ip dhcp snoop**.

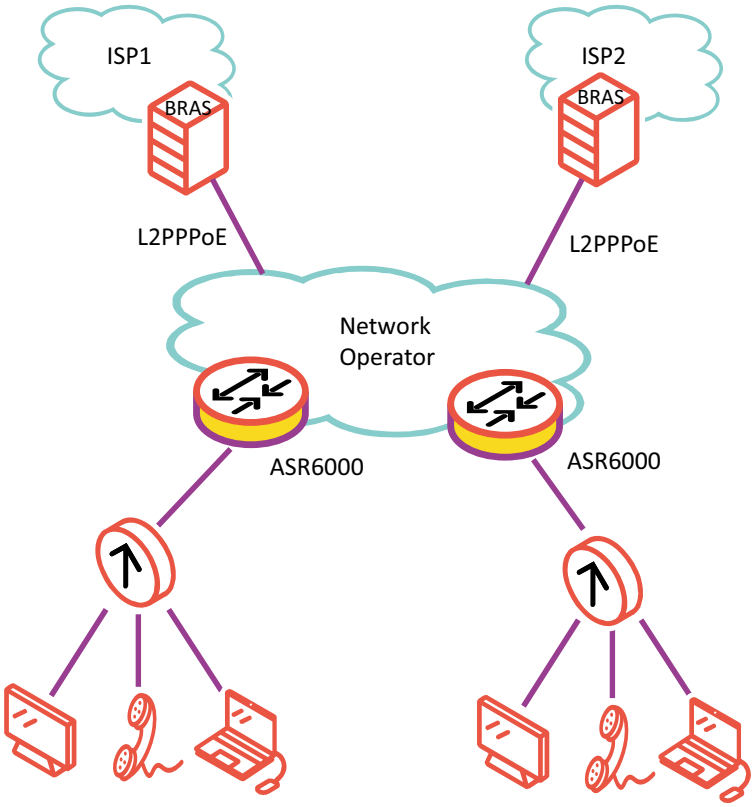
VLAN Translation

VLAN Translation, also known as VLAN mapping or VLAN switching, is used in L2 networking. It is used for selective inter-VLAN switching of non-routable protocols, and also to extend a single VLAN topology across mixed switching environments.

When an interface has VLAN translation enabled, iBOS can translate the VLAN IDs carried in the data packets it receives from private networks into those used in the carrier network.

The **clear mac-address-table** command should be avoided when VLAN translation is enabled. If absolutely necessary, clear the MAC address table for one interface at a time to avoid overloading the switch.

Figure 121 VLAN Translation



To manage VLAN translation, the **policy vlan-translation** configuration command has been added.

The following example shows how the ASR uses VLAN translation to send all IPv6 via tunnel1:

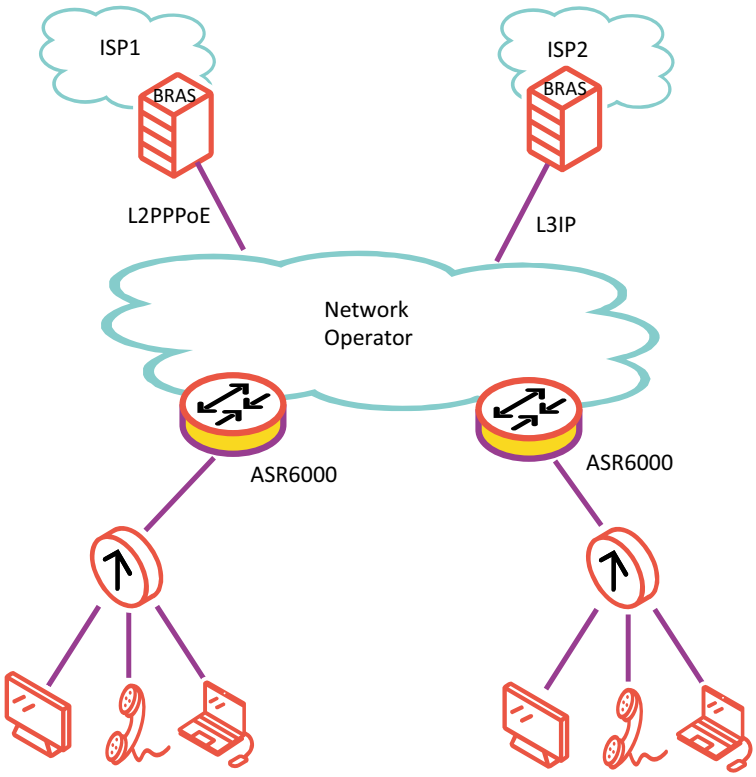
```
interface loopback0
ip address 10.0.0.101 255.255.255.0
no shutdown

interface tunnel1
  tunnel mode l2tpv3
  l2tpv3 10.0.0.1 t1
  vlan untagged 54
  vlan member 54
  no shutdown

interface gigabitethernet8
  remote vid 8
  ip address 10.31.8.1 255.255.255.0
  no shutdown

interface gi8/vid8
  policy v6pol vlan-translation internal 54 uni external untagged
  policy-map ether-type ipv6 seq 10 policy v6pol

l2tpv3 10.0.0.1 source-interface loopback0
```

Policy Map EtherTypes

The EtherType field is used to indicate which protocol is being transported in an Ethernet frame. Using an ethertype with the **policy-map** command makes it possible to manage traffic based on the protocol being used.

For example, this feature can be used in combination with policy vlan-translation to map a specific ethertype (such as PPPoE or IPv6) to a bridged vlan while still using standard IPv4 features on the interface.

Figure 122 Mixed L2/L3 ISP Network



For example, PPPoE has two ethertypes, one for discovery stage (pppoe-discover) and one for session traffic (pppoe-session).

This example will redirect all PPPoE traffic on VLAN 2, all other EtherTypes are routed as usual.

```
interface fastethernet3
  policy pppoe vlan-translation internal 2 uni external untagged
  policy-map ether-type pppoe-session seq 10 policy pppoe
  policy-map ether-type pppoe-discover seq 10 policy pppoe
```

Load Balancing

iBOS supports Equal Cost Multi Path (ECMP) load balancing which can substantially increase bandwidth by load-balancing traffic over multiple paths. The implementation is based on IEEE standard 802.1Qbp. For more details on multi-path routing, see RFC2991.

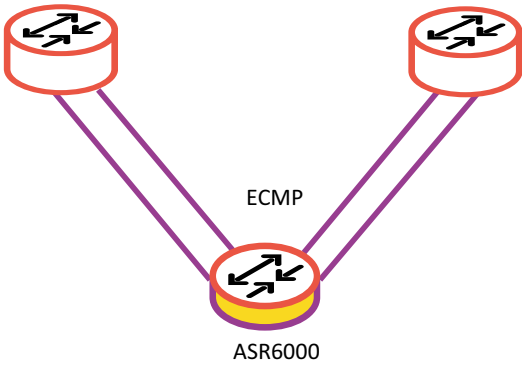
When using ECMP on multiple uplinks to one distribution unit, it is recommended that the following OSPF configuration is used towards the iBOS router:

```
area <area ID> nssa default-information-originate no-summary
```

This disables the LSA summary into NSSA, and therefore iBOS will receive the default network as OSPF inter-area, and not as OSPF NSSA external type 2.

By default, ECMP is enabled in iBOS, and it is configured using the command **ip ecmp**.

Figure 123 ECMP Load Sharing Multiple GE Links



Policy Routing

When a router receives a packet, it normally decides to make routing decisions based on the destination address which it uses to lookup an entry in a routing table. To route packets based on other criteria, such as destination, packet size, protocol payload or any other information available from packet header or payload, a policy routing method is used.

iBOS supports a policy based routing (PBR) Layer 3 feature for routing redirection: specific rules are applied to L3 related packet types, and when the packets match a PBR rule, iBOS handles them in a manner that overrides the normal L3 routing lookup behaviour.

PBR can be configured to redirect packets to another nexthop (IPv4 neighbour address).

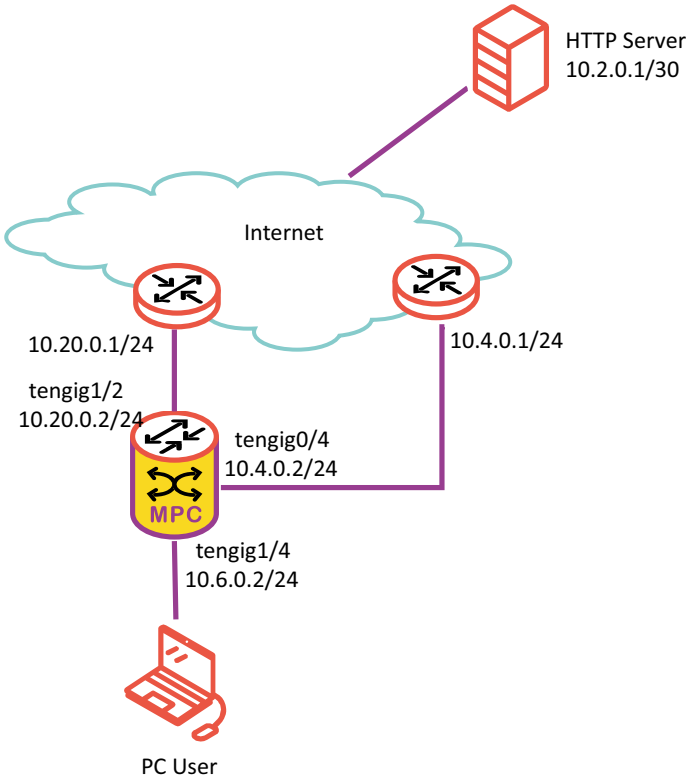
To match rules, PBR uses policy-map statements, with or without access-lists, to classify packets in a similar way to the generic iBOS packet classification. Packets that are not IP routed, such as L2 bridging packets, can not use PBR since they neither support VRF nor IP nexthop configuration. PBR also requires that the packet ingress interface is a Layer 3 interface with an IP address.

Currently, PBR supports 31 different next-hops.

Configure PBR

PBR can be used to redirect traffic to a proxy server by using a PBR-aware L3-switch (router). In such a deployment, specific types of traffic (e.g. HTTP, FTP) can be redirected to a cache engine. This is also known as virtual inline deployment.

Figure 124 Configure PBR



In the example, the PBR feature on the MPC480 redirects the PC User client to the web page on the HTTP server via 10.4.0.1 - the rerouting is performed even if the route entry for destination 10.2.0.1 does not exist. PBR policy-matched traffic does not use the routing table and has a priority over the normal routing process:

To configure PBR on the MPC480, do the following:

- The MPC480 interface tengigabitethernet1/2 is used for normal routing of traffic. If no PBR was configured on tengigabitethernet1/4, a normal routing entry for destination of HTTP sever would select this interface for egress. In this example, the traffic from PC to HTTP sever does use this interface because the PBR configuration overrides the normal routing and redirects traffic to the nexthop 10.4.0.1. Configure interface tengigabitethernet1/2:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface tengigabitethernet1/2`	Specify the interface on MPC480 to configure, and enter Interface mode.

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config-if-te)# ip address 10.20.0.2/24`	Set IP address for interface
`iBOS(config-if-te)# no shutdown`	Bring up interface.

For traffic originating from the HTTP server to PC User, and for un-matched PBR traffic, this interface will be use normal routing matching.

1. Configure PBR on interface tengigabitethernet1/4:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface tengigabitethernet1/4`	Specify the interface on MPC480 to configure, and enter Interface mode.
`iBOS(config-if-te)# ip address 10.6.0.2/24`	Set IP address for interface Bring up interface.
`iBOS(config-if-te)# no shutdown`	
`iBOS(config-if-te)# policy http-pbr ip next-hop 10.4.0.1`	Create PBR policy for next-hop redirect
`iBOS(config-if-te)# policy-map 0.0.0.0/0 seq 20 policy http-pbr full-access-group acl2`	Create a policy-map with the PBR policy and access-list, and attach it to the interface
`iBOS(config-if-te)# interface access-list acl2 seq 30 permit tcp host 10.6.0.1 host 10.2.0.1 80`	Define access list under this interface. The access list acl2 is attached to that policy-map http-pbr which will match ingress packets.
`iBOS(config-if-te)# no shutdown`	Bring up interface

2. The interface tengigabitethernet0/4 is the egress interface for redirected traffic and is connected to the device on interface 10.4.0.1/24 that receives the redirected traffic. Configure interface tengigabitethernet0/4:

`iBOS# configure terminal`	Enter Configure mode.
`iBOS(config)# interface tengigabitethernet0/4`	Specify the interface on MPC480 to configure, and enter Interface mode.
`iBOS(config-if-te)# ip address 10.4.0.2/24`	Set IP address for interface
`iBOS(config-if-te)# no shutdown`	Bring up interface

Refer *iBOS Command Reference* for a detailed description of the **policy ip next-hop** command.

To verify the configuration, use the **show policy** and **show policy-map** commands.

Show that policy-map/policy/ACL is attached under tengigabitethernet1/4:

```
iBOS# show policy-map
Interface
      Key                     Seqno Access-Group      Policy
------------------------------------------------------------------------
------------
tengigabitethernet1/4
```

```
         0.0.0.0/0                              20 acl2
http-pbr
```

Show that traffic is matched by PBR policy under tengigabitethernet1/4:

```
iBOS# sh policy
                            Input                        Output
Name                     Packets   Bytes   Drops   Packets   Bytes
Drops
-------------------------------------------------------------------
-------------
tengigabitethernet1/4
  http-pbr                528.0M  52.53G      0       0    0      0
```

Flow Inspection

The flow inspection feature is a simple deep packet inspection (DPI) that examines the data in packets outside the packet header. This type of DPI allows iBOS to intelligently look for a specific packet in the traffic flows forwarded through the switch.

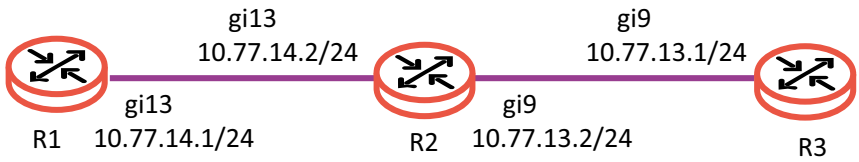
Flow inspection is used for troubleshooting:

- To determine if a service is working correctly by retrieving the current traffic rate for a specific user from a BNG.
- To determine how much bandwidth that is consumed by individual applications by the end-user by retrieving detailed traffic information. For example, to ascertain if a big download is consuming all the bandwidth at the end-user.
- To determine why an outage occurs for groups of users by retrieving the total bandwidth going to all or groups of end users. For example, to verify that traffic is flowing to and from affected customer VLANs.

Configure Flow Inspection

The example is configured to be able to filter and check OSPF consistent flows from the flow-cache table.

Figure 125 Flow Inspection Example



To work, the flow inspection feature must be enabled:

```
R2(config)# flow-cache rate-calculate
```

To configure flow inspection, the procedure has two steps:

- Configure OSPF protocol for both IPv4 and IPv6 on switches R1, R2 and R3.
- Enable flow inspection to check Tx and Rx rates in the switch R2 (BNG device).
- After configuring flow inspection, check Tx and Rx rates (using **show ip-flow-cache proto icmp bit-rate duplex**).

1. Configure switch R1:

```
R1# config terminal
R1(config)# interface gig13
R1(config-if-gi)# ip address 10.77.14.2/24
R1(config-if-gi)# ipv6 address 2001:db8:1:1::24/64
R1(config-if-gi)# no shutdown
R1(config-if-gi)# exit
R1(config)# router ospf
R1(config-router-ospf)# router-id 10.77.77.1
R1(config-router-ospf)# network 10.77.14.0 255.255.255.0 area 0
R1(config-router-ospf)# network 10.77.77.1 255.255.255.255 area 0
R1(config-router-ospf)# exit
R1(config)# router ospfv3
R1(config-router-ospfv3)# router-id 1.1.1.1
R1(config-router-ospfv3)# exit
R1(config)# interface gig13
R1(config-if-gi)# ip ospf hello-interval 1
R1(config-if-gi)# ipv6 ospf hello-interval 1
R1(config-if-gi)# ipv6 router ospf area 0.0.0.0
R1(config-if-gi)# end
R1#
```

2. Configure switch R2:

```
R2# configure terminal
R2(config)# interface gig13
R2(config-if-gi)# ip address 10.77.14.1/24
R2(config-if-gi)# ipv6 address 2001:db8:1:1::25/64
R2(config-if-gi)# no shutdown
R2(config-if-gi)# interface gig9
R2(config-if-gi)# ip address 10.77.13.1 255.255.255.0
R2(config-if-gi)# ipv6 address 3103:10:67:23::100/64
R2(config-if-gi)# no shutdown
R2(config-if-gi)# exit
R2(config)# router ospf
R2(config-router-ospf)# router-id 10.77.77.2
R2(config-router-ospf)# network 10.77.14.0/24 area 0.0.0.0
R2(config-router-ospf)# network 10.77.13.0/24 area 0.0.0.0
R2(config-router-ospf)# network 10.77.77.2/32 area 0.0.0.0
R2(config-router-ospf)# exit
R2(config)# router ospfv3
R2(config-router-ospfv3)# router-id 2.2.2.2
R2(config-router-ospfv3)# exit
R2(config)# interface gig9
R2(config-if-gi)# ip ospf hello-interval 1
R2(config-if-gi)# ipv6 ospf hello-interval 1
R2(config-if-gi)# ipv6 router ospf area 0.0.0.0
R2(config-if-gi)# exit
R2(config)# interface gig13
R2(config-if-gi)# ip ospf hello-interval 1
R2(config-if-gi)# ipv6 ospf hello-interval 1
R2(config-if-gi)# ipv6 router ospf area 0.0.0.0
```

```
R2(config-if-gi)# end
R2#
R2# configure terminal
R2(config)# flow-cache rate-calculate
```

3. Configure switch R3:

```
R3# config terminal
R3(config-if-gi)# interface gig9
R3(config-if-gi)# ip address 10.77.13.2 255.255.255.0
R3(config-if-gi)# ipv6 address 3103:10:67:23::101/64
R3(config-if-gi)# no shutdown
R3(config-if-gi)# exit
R3(config)# router ospf
R3(config-router-ospf)# router-id 10.77.77.3
R3(config-router-ospf)# network 10.77.13.0 255.255.255.0 area 0
R3(config-router-ospf)# network 10.77.77.3 255.255.255.255 area 0
R3(config-router-ospf)# exit
R3(config)# router ospfv3
R3(config-router-ospfv3)# router-id 3.3.3.3
R3(config-router-ospfv3)# exit
R3(config)# interface gig9
R3(config-if-gi)# ip ospf hello-interval 1
R3(config-if-gi)# ipv6 ospf hello-interval 1
R3(config-if-gi)# ipv6 router ospf area 0.0.0.0
R3(config-if-gi)# end
R3#
```

To check OSPF neighbours and status:

```
R1# show ip ospf neighbour

OSPF process 0:
neighbour ID     Pri   State           Dead Time   Address
Interface
-------------------------------------------------------------------
--------------
10.77.77.2        1   Full/DR          00:00:03    10.77.14.1
gigabitethernet13

R2# show ip ospf neighbour

OSPF process 0:
neighbour ID     Pri   State           Dead Time   Address
Interface
-------------------------------------------------------------------
--------------
10.77.77.3        1   Full/DR          00:00:04    10.77.13.2
gigabitethernet9
10.77.77.1        1   Full/Backup      00:00:04    10.77.14.2
gigabitethernet13

R3# show ip ospf neighbour

OSPF process 0:
neighbour ID     Pri   State           Dead Time   Address
Interface
-------------------------------------------------------------------
--------------
10.77.77.2        1   Full/Backup      00:00:04    10.77.13.1
gigabitethernet9
```

To check OSPF IPv6 neighbours and status:

```
R1# show ipv6 ospf neighbour

OSPFv3 Process (*null*)
      neighbour ID     Pri   State           Dead Time   Interface
Instance ID
     2.2.2.2         1   Full/DR       00:00:04   gigabitethernet13
0

R2# show ipv6 ospf neighbour

OSPFv3 Process (*null*)
      neighbour ID     Pri   State           Dead Time   Interface
Instance ID
     3.3.3.3         1   Full/DR       00:00:03   gigabitethernet9 0
     1.1.1.1        1   Full/Backup    00:00:03   gigabitethernet13
0

R3# show ipv6 ospf neighbour

OSPFv3 Process (*null*)
      neighbour ID     Pri   State           Dead Time   Interface
Instance ID
     2.2.2.2         1   Full/Backup    00:00:04   gigabitethernet9
0
```

Flow Inspection for Troubleshooting

When OSPF is setup and flow inspection is configured, the switch R2 can be used to retrieve statistics for troubleshooting. The Tx and Rx rates can be retrieved using the following filters:

- IP (with prefix option)
- IPV6
- SNPA
- DSCP
- Interface
- Protocol
- Port

To check all packet flows on switch R2:

```
R2# show ip flow-cache

Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                       Destination
Type
```

```
------------------------------------------------------------------
--------------
   10.77.13.2                    224.0.0.5                      C
   10.77.14.2                    224.0.0.5                      C
ICMP   10.77.13.2:0:0              10.77.13.1                     C
```

To check packet flows on switch R2 filtered on IP:

```
R2# show ip flow-cache ip 10.77.13.2
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                      Destination
Type
------------------------------------------------------------------
--------------
   10.77.13.2                    224.0.0.5                      C
   10.77.13.2:0:0                10.77.13.1                     C
```

To check packet flows on switch R2 filtered on protocol ICMP:

```
R2# show ip flow-cache ip 10.77.13.2 proto ICMP
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                      Destination
Type
------------------------------------------------------------------
--------------
ICMP   10.77.13.2:0:0              10.77.13.1                     C
```

To check packet flows on switch R2 filtered on protocol and interface:

```
R2# show ip flow-cache ip 10.77.13.2 proto ospf interface
gigabitethernet9
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                      Destination
Type
------------------------------------------------------------------
--------------
OSPF   10.77.13.2                  224.0.0.5                      C
```

To show detailed packet flows on switch R2 filtered on IP, OSPF, interface, and SNPA:

```
R2# show ip flow-cache ip 10.77.13.2 proto ospf interface
gigabitethernet9 snpa 0008.ae87.0e60 detailed
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto   Source                        Destination
Type
          Activity index
-------------------------------------------------------------------
--------------
OSPF    10.77.13.2:{48}               224.0.0.5                   C
         0008.ae87.0e60                    0100.5e00.0005
         gigabitethernet9
           0.10
```

To show all detailed IPv6 packet flows on switch R2:

```
R2# show ipv6 flow-cache detailed
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto   Source                        Destination
Type
          Activity index
-------------------------------------------------------------------
--------------
IP-89   fe80::208:aeff:fe87:e60:{0}     ff02::5
Cm
         0008.ae87.0e60                    3333.0000.0005
        gigabitethernet9
           0.26
IP-89   fe80::208:aeff:fe87:15c0:{0}    ff02::5
Cm
         0008.ae87.15c0                    3333.0000.0005
         gigabitethernet13
           0.49
```

To show detailed IPv6 packet flows on switch R2 filtered on IP:

```
R2# show ipv6 flow-cache ipv6 ff02::5
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto   Source                        Destination
Type
-------------------------------------------------------------------
```

```
--------------
IP-89  fe80::208:aeff:fe87:e60        ff02::5
Cm
IP-89  fe80::208:aeff:fe87:15c0       ff02::5
Cm
```

To show detailed IPv6 packet flows on switch R2 filtered on IP and protocol :

```
R2# show ipv6 flow-cache ipv6 fe80::208:aeff:fe87:15c0 proto IP-89
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                       Destination
Type
-------------------------------------------------------------------
--------------
IP-89  fe80::208:aeff:fe87:15c0       ff02::5
Cm
```

To show detailed packet flows on switch R2 filtered on IPv6, protocol and interface:

```
R2# show ipv6 flow-cache ipv6 fe80::208:aeff:fe87:15c0 proto IP-89
interface gigabitethernet13
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                       Destination
Type
-------------------------------------------------------------------
--------------
IP-89  fe80::208:aeff:fe87:15c0       ff02::5
Cm
```

To show detailed packet flows on switch R2 filtered on IPv6, protocol. interface and SNPA:

```
R2# show ipv6 flow-cache ipv6 fe80::208:aeff:fe87:15c0 proto IP-89
interface gigabitethernet13 snpa 3333.0000.0005
Type: R - Routed, B - Bridged, D - Denied, C - Cpu, M - Multicast
      G - GRE decapsulation, L - L2TPv3 decapsulation, X - Cross
connect
      Nd - Destination NAT, Ns - Source NAT
      mI - MPLS Ingress, mF - MPLS forward, mE - MPLS Egress
      i - Ingress filter, e - Egress filter, r - Rejected route
      m - unknown multicast (tbm policer)

Proto  Source                       Destination
Type
-------------------------------------------------------------------
--------------
IP-89  fe80::208:aeff:fe87:15c0       ff02::5
Cm
649
```

Extended iBOS

iBOS has extensions that complement the generic functionality available in iBOS. The extensions may be hardware specific, and not necessarily included in all iBOS devices. The iBOS extensions are described in:

- "Realtime Protocol Monitoring" on page 446
- "Scripts" on page 447
- "Mirrored Traffic" on page 451
- "Lawful Interception" on page 452
- "Selective Q-in-Q" on page 460

Realtime Protocol Monitoring

The Realtime Protocol Monitoring (RPM) feature allows network owners to monitor the TV quality in broadband networks in real time. RPM ensures both higher service quality and faster, more efficient troubleshooting of IPTV streams, which ultimately lowers the operational expenses of running a triple-play network. It also removes the need for investment in external probes and tools, and the time consuming manual handling that such tools require.

TV viewers expect a high quality service. The smallest disturbance or interruption to the picture is an immediate source for customer complaints and viewers will be quick to let you know if a multicast TV service does not meet their expectations. A good TV diagnostics tool is essential. RPM allows network owners to monitor the multicast channels that run through each iBOS device in realtime, making it possible to discover any disturbances even before end users experience it as “bad TV quality”, or freezing pictures that require a restart of set top boxes. RPM tracks data, such as packet loss and reordered packets within the network, and can be setup to determine if the loss of packets are outside the network, e.g. in the satellite feed.

Using RPM network owners can locate the origin of the problem without expensive diagnostic tools, and even more important, without spending the time and resources on manually tracking suspected element(s) where the fault might originate. Because all data is stored, it is easily transferred to a customer care system for graphical presentation or for further analysis.

To measure MPEG quality in multicasts, the following metrics are used:

- Metrics at RTP level
- Metrics at Transport Stream level
- Metrics at Packetized Elementary Stream level

To configure RPM, the following commands have been added:

- **rpm multicast mpeg rtp access-group**
- **rpm multicast mpeg udp access-group**
- **clear rpm**
- **show rpm**

Realtime Protocol Monitoring Logging

Network owners can check the multicast group quality state manually using the iBOS CLI or SNMP MIBs.

This feature extends the functionality of the RPM module with logging and automatic SNMP traps when there is any quality degradation in multicasts. This support for logging and trap events provides more flexible monitoring of MPEG quality in multicast traffic. The error events supported are:

- sequence-error per RTP multicast group
- jitter per RTP multicast group
- missing-sync-byte per TS multicast group
- misaligned per TS multicast group

The commands to configure realtime multicast logging are:

- **rpm multicast rtp**
- **rpm multicast mpeg**
- **rpm log**
- **rpm snmp trap**

Scripts

To extend the functionality of a standard iBOS device, the Scribos script engine allows scripts to interact with iBOS and add additional processes that perform specific tasks during runtime. The iBOS scripts are described in:

- "802.1X Authentication Script" on page 447
- "DHCP Relay Script" on page 448
- "DHCPv6 Relay Script" on page 449
- "DHCPv6 Snooper Script" on page 450
- "PPPoE Intermediate Agent Script" on page 450
- "Static RADIUS Script" on page 451

802.1X Authentication Script

In Waystream networks, services provided for clients are often deployed using a network management system. In iBOS, a service is a group of configuration statements, for example, policies, policy-maps, access-lists, vlan members etc.

Using Scribos, it is possible to allow autonomous or semi-autonomous service configuration in the iBOS device.

To configure service selection using 802.1X, an Administrator Server (AS) is setup to determine which service is forwarded to clients. The iBOS script ***dot1x-server*** adds support for a RADIUS server to authenticate clients and then the script selects a service for clients based on 802.1X.

To configure authentication and service selection using 802.1X, the following commands are used:

- **port authentication 802.1X**
- **service-definition**
- **service-selection**

The command **script run dot1x-server show ports status** shows the authentication status of 802.1X enabled ports. The command **script run dot1x-server show clients** shows the authentication status for the clients on the 802.1X enabled ports.

For more information, refer *iBOS Command Reference*.

DHCP Relay Script

In an iBOS IPv4 network, client host configuration including the assignment of IP addresses normally use DHCP (RFC 2131). If server and a client are not on the same physical subnet, the DHCP message need to be relayed.

iBOS can act as a DHCP relay agent. In such a scenario, clients get the IP address from an external DHCP server via the iBOS device, and then iBOS automatically enforces the service for the client on the interface (using a group of policy/policy-map).

To deploy the feature of dhcp relay support for an external DHCP server on an iBOS device, the following commands are used:

- **ip dhcp relay script**
- **parameter dhcp client-roaming**
- **parameter dhcp service service-definition**
- **parameter dhcp relay-to dhcp-server**
- **parameter dhcp match**
- **service-selection**
- **service-definition**

The command **script run dhcp show clients** displays information about the clients online.

For more information, refer *iBOS Command Reference*.

DHCPv6 Relay Script

In an iBOS IPv6 network, client host configuration including the assignment of IPv6 addresses normally use stateful auto-configuration or DHCPv6 (RFC 3315). If server and a client are not on the same physical subnet, the DHCPv6 message need to be relayed.

iBOS can act as a DHCPv6 relay agent, in this scenario clients can get the IPv6 address from a server via iBOS, then iBOS auto enforce the service (group of policy/policy-map) for the client on the interface.

In some IPv6 deployment scenario, client (probably CPE) may also request IPv6 Prefix from DHCP server, which can be used by CPE to assign IPv6 address for its internal network clients (RFC 3633). iBOS can also relay such DHCPv6 negotiation packets. Normally, the upstream router does not has any knowledge of the route to the delegated prefix. iBOS dhcpv6 relay script will snoop the IPv6 address and prefix assigned for client, and install a route entry for the prefix.

To deploy dhcpv6 relay agent feature on an iBOS device, the following commands are used:

- **ipv6 dhcp relay script**
- **ipv6 nd managed-config-flag**
- **ipv6 address**
- **parameter dhcpv6 service service-definition**
- **parameter dhcpv6 relay-to**
- **parameter dhcpv6 tag**
- **parameter dhcpv6 match**
- **service-selection**
- **service-definition**

The command **script run dhcpv6 show clients relay** displays information about the relayed clients online.

For more information, refer *iBOS Command Reference*.

DHCPv6 Snooper Script

In an iBOS IPv6 Layer 2 network, client host configuration including the assignment of IPv6 addresses normally uses stateful auto-configuration or DHCPv6 (RFC 3315). The server and the client are on the same Layer 2 VLAN, the DHCPv6 message is multicasted or bridged between the client and server in the same subnet.

The DHCPv6 snooping functionality adds support for inserting interface and remote client information into DHCPv6 packets which then be used by the DHCPv6 server to find out the location of the client requests an address. The information is stored in two specific options: the option 18 **interface-id** contains information of the interface that received the DHCPv6 packets sent by the client, and the option 37 remote-id contains the system ID of the remote client.

iBOS can act as a DHCPv6 snooper using the script **dhcpv6**. To do this, the script picks all DHCPv6 packets from specified VLANs on a port. The script subsequently inserts option interface-id and remote-id into the DHCPv6 packets sent by the client and forwards them to the DHCPv6 server. When a reply is received from the DHCPv6 server, the same options is removed from the DHCPv6 packets before being sent to the client.

To deploy the dhcpv6 snooper feature on a device, the following commands are used:

- **ipv6 dhcp snoop script vlan**
- **parameter dhcpv6 interface-id**
- **parameter dhcpv6 remote-id**

The command **script run dhcpv6 show clients snoop** displays information about the relayed clients online.

For more information, refer *iBOS Command Reference*.

PPPoE Intermediate Agent Script

The iBOS script *pppoe-ia* allows iBOS to act as a PPPoE intermediate agent for a set of VLANs. As such, iBOS acts as a PPPoE intermediate agent (RFC2516) by intercepting PPPoE control traffic from clients and inserting a vendor specific TAG as defined in the TR-101 specification.

To configure the PPPoE-IA, the following commands are used:

- **port pppoe intermediate-agent**
- **parameter pppoe-ia agent-circuit-id**
- **parameter pppoe-ia agent-remote-id**

The command **script run pppoe-ia show clients** get information about the clients.

For more information, refer *iBOS Command Reference*.

Static RADIUS Script

iBOS supports script configuration of service deployment by RADIUS for static clients.

In an iBOS IPv4 network, clients host may be static, i.e. the client get permanently assigned IP address, IP mask, and gateway from the service provider. This is different to normal clients that get dynamic IP addresses using DHCP. To make full use of available bandwidth, a service provider will usually apply the appropriate service (i.e. policy/policy-map, access-list) when the client comes online, and then removes the service when the client goes offline.

For DHCP clients, the client online/offline state is an obvious trigger for DHCP-REQUEST/DHCP-RELEASE messages. For static clients, however, another method has to be used to determine if the client is online/offline and this is to listen for the first ARP-Request from the client.

In a static RADIUS network, there are a number of devices involved:

Device	Description
client Host	client PC (or any other terminal device) that requires IP network access.
BNG	Border Network Gateway, a server connecting all client hosts into a service provider network. It accepts the client-host network access request (ARP-Request) and functions as a RADIUS client that authenticates the client host towards RADIUS server, and subsequently applies the service(s) authorized by RADIUS server for that client. The BNG also provides and updates related service accounting statistics of client host to RADIUS server. iBOS acts as a BNG in the static radius network scenario.
Captive-Portal Server	A Login server which provides a webpage to let the user of client host to input username/password to authenticate access right. It actually use user provided username/password to authenticate towards RADIUS server.
RADIUS Server	A server authenticates client hosts via RADIUS protocol. It can provides service authorization, accounting, dynamically update service for client host via CoA request, or dynamically disconnect client host via CoA disconnect request as well.

To configure the static RADIUS service deployment, the following commands are used:

- **ip static script**
- **service-definition**
- **service-selection**
- **parameter static client-roaming**
- **parameter static radius server**

The command **script run static show clients** displays information about the static clients.

For more information, refer *iBOS Command Reference*.

Mirrored Traffic

In some countries, there are requirements for lawful interception of network traffic. iBOS has built-in support for Lawful Interception (LI), however if no mediation server has been implemented in the network there is a basic mirroring function that may be used instead. The mirroring function can also be useful for troubleshooting network problems.

The mirror functionality must not be visible to the end user. Mirrored traffic is copied to another interface which then sends the intercepted traffic to a configured port.

Mirrored traffic is configured for each downlink interface. Traffic may be selected based on whether it is unicast, multicast or broadcast.

For multicast traffic, only software forwarded traffic is mirrored, e.g. IGMP messages, but not the actual IPTV data.

Mirrored traffic is implemented for unicast, broadcast and multicast.

To configure mirrored traffic, the **mirror** and **show mirror** commands are used.

Mirroring on some platforms can also be sent to a GRE tunnel interface. This allows the mirrored traffic to be received and analysed at a central location instead of on a different physical port of the device itself. Remote mirroring is useful for network troubleshooting as well as to meet legal intercept requirements.

Lawful Interception

Overview

Lawful Interception also known as wiretapping is a requirement which is basically defined for providing official access to private communication in a legalized way. The basic principle of LI is to duplicate in real time, the traffic of any specific target or customer and forward it to the authorities concerned. It calls for the use of a mediation platform i.e., mediation server to handle the provisioning of the interceptions as well as the collection and delivery of intercepted traffic.

Following are the two deliveries of LI:

- *Intercepted Related Information (IRI):* It includes the signalling information
- *Content of Communication (CC):* It includes voice, video or message content

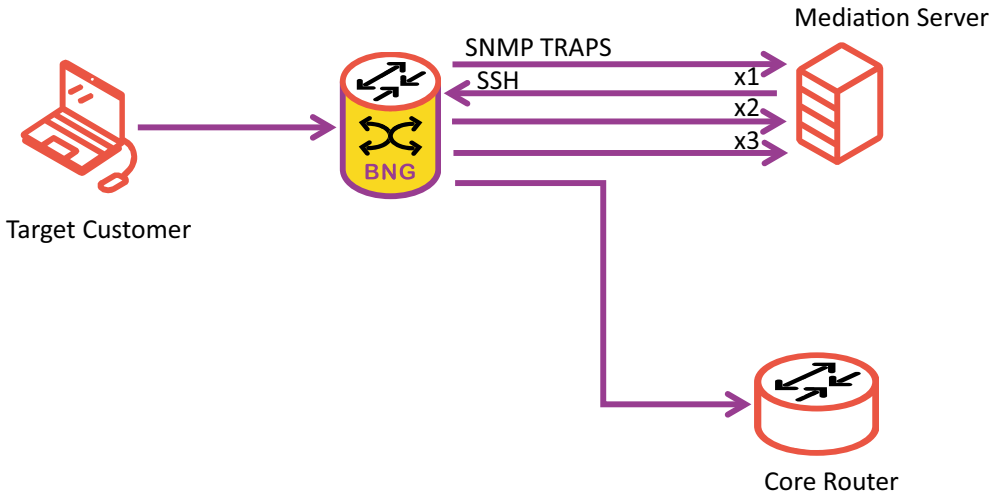
The prerequisite prior to defining any LI requirement is that due to this functionality, there should not be any impact on the normal service provided to the individual.

Architecture

The figure depicts a generalized view of Lawful Intercept process, wherein a network element with support for Lawful Interception (LI) implements the functionality to monitor any specific target or customer. This functionality is achieved by intercepting the customers data packets and reporting them to the mediation server.

A tunnel encapsulation is imposed to the intercepted packets before the packets are sent to the remote endpoint (Mediation Server).

Figure 126 Functional Diagram



The different interfaces are illustrated in the tabular column:

Table 42 Interfaces

Interface	Description
x1	The configuration of lawful intercept tasks is performed using x1 interface. It is only available for LI user who authenticate themselves using SSH.
x2	The reporting of events with respect to any specific target under surveillance is performed using this interface. x2 interface is not supported in the current release.
x3	Also known as data reporting interface is responsible for the tunnelling of intercepted packets over UDP/IP. It makes use of proprietary shim header (wMirror) between the tunnel encapsulation and the intercepted packet.
SNMP Trap	At the time of iBOS restart, the transfer of cold or warm trap to the mediation server is performed using this interface.

LI User Configuration and Global LI Settings

The LI configuration tasks include add, delete, and update. All these tasks are performed by an LI user whose account has been created by the system administrator. When the account has been created, only then the LI user is able to login the system. The process of creating LI user account is much similar to the way of creating normal user account. But, the difference is that the former creation process even assigns the LI role (optional CLI command attribute) to a user.

Apart from creating an account, the administrator is even responsible for defining and configuring global LI parameters, which includes the following:

- *Maximum count of LI tasks:* The command **lawful-intercept max-task <num>** is used for defining the count of LI tasks on iBOS. By default, all the LI tasks which are supported by a particular system has to be enabled.

```
ibos(config)# [no] lawful-intercept max-task <num>
```

- *Set intercept/filter mode to either IP address or Interface:* The command **lawful-intercept filtering-mode** is used to configure interface or IP address filtering mode. The default

filtering mode is set to interface. As the intercept system allows only one filtering mode to be configured at a time, the configuration of LI tasks related to other mode will not be accepted.

```
ibos(config)# [no] lawful-intercept filtering-mode ipaddress
```

If the target to be monitored is considered on a specific interface, then it is mandatory for this target interface to be linked only with the target customer. The main reason for this association is to avoid interception of traffic related to other customers.

- *Set Global LI tunnel encapsulation parameter and MTU:* The parameters associated with LI tunnel encapsulation and MTU are:
  - IPv4 TTL
  - IPv4 ToS/DSCP
  - LI Tunnel MTU

```
ibos(config)# [no] lawful-intercept tos <0-255>
ibos(config)# [no] lawful-intercept ttl <1-255>
ibos(config)# [no] lawful-intercept mtu <1464-1536>
```

LI user accounts and global LI parameters defined by the system admin are stored in persistent memory using write command, which means that this configuration will be available after a system restart.

LI Task Configuration [x1 Interface]

The configuration of LI interface tasks (add, delete, and update) is performed using x1 interface. These configuration tasks can only be carried out by an LI user who has logged into the system using SSH. Any login attempt using TELNET or CONSOLE would be rejected.

It is encouraged that the LI user resets his password from the one created by the system admin.

The logged in LI user can perform operations like delete or display the status of running tasks that are owned by self i.e.,the operation of a specific LI user is independent from other LI users. For example, if LI user1 is performing any operation, then this user is not able to view the tasks of any other LI user.

In order to intercept traffic from one specific target (i.e., interface or ip address), LI task is configured.

Create LI task for an Target IP address (in default VRF):

```
ibos(config)# lawful-intercept task interceptid 1 mediation-server
10.12.2.2 port 49153 target-ip 10.192.1.2
```

Create LI task for an Interface:

```
ibos(config)# lawful-intercept task interceptid 2 mediation-server
10.13.3.3 port 49154 target-interface gigabitethernet4
```

Create LI task for an Target IP address in VRF:

```
ibos(config)# lawful-intercept task interceptid 3 mediation-server
10.14.4.4 port 49155 vrf north target-ip 10.10.1.2
```

LI Task configuration is stored in volatile memory, which means that this configuration will not be available after system restart and the LI tasks has to be reconfigured again by the LI user. To notify the mediation server about the system

restart, it is encouraged that the mediation server is added as an SNMP trap receiver to the system configuration.

Example 88 Volatile LI Configuration:

```
!volatile lawful-intercept task interceptid 1 mediation-server
10.12.2.2 port 49153 target-ip 10.192.1.2

!volatile lawful-intercept task interceptid 3 mediation-server
10.14.4.4 port 49155 vrf north target-ip 10.10.1.2

!volatile lawful-intercept interceptid 2 mediation-server
10.13.3.3 port 49154 target-interface gigabitethernet4
```

Packet Interception and Reporting [x3 interface]

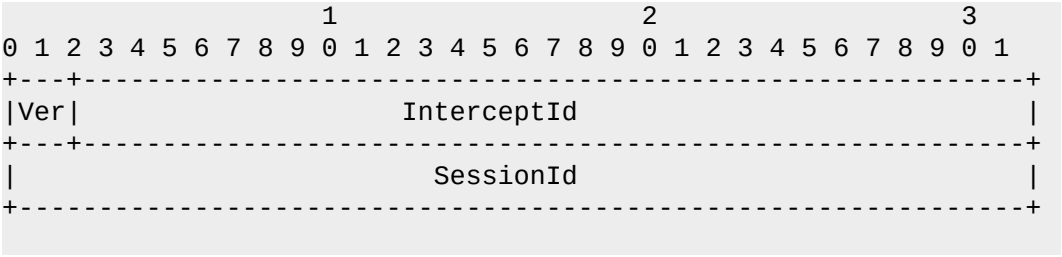
The intercepted packets are encapsulated over UDP/IPv4 and forwarded to the mediation server without changing the original packet which is forwarded to the desired destination. The encapsulated packets needs to undergo fragmentation before being forwarded to the mediation server if their size exceeds the MTU size.

The tunnel MTU size is determined by,

- Global tunnel MTU parameter
- MTU, which is set for outgoing interface

The proprietary Waystream header (8 octets size) as shown in figure is inserted between the UDP/IPv4 encapsulation and intercepted packet.

Figure 127 Waystream Shim Header (wMirror)



The Waystream Shim header consists of the following fields:

- *Version id:* It is a 2-bit field whose default value is set to zero for this version.
- *Intercept id:* It is a 30-bit field whose value is configured by the LI user with respect to a specific task.
- *Session id:* It is a 32-bit field which is basically a random unique number that is associated with every LI task.

Apart from these fields, the following fields are also set before forwarding the intercepted packet to mediation server.

- *UDP destination port:* Configured for each LI task and must correspond to mediation server configuration.
- *IPv4 destination address:* Configured for each LI task and must correspond to mediation server configuration

- *UDP source port:* The value of this field is set identical to the value of UDP destination port
- *IPv4 source address:* This value is set by resolving the mediation server IP address, in accordance with the routing information base and associated outgoing interface configuration.

State of LI Task:

An LI task may be in any of the following two states: UP and DOWN.

UP: An LI task is said to be in UP state, if the mediation sever is reachable.

DOWN: An LI task is said to be in DOWN state, if the mediation server is not reachable. In this state, the LI module executes the functionality to retry for resolving LI tasks, which happens if any corresponding routing change occur.

SNMP Cold/Warm Start Trap Notification

Apart from x1,x2, and x3 interfaces, a standard SNMP trap interface is also required which basically informs the mediation server regarding the planned or unplanned restart of the system. This information is send to the server by using Cold/Warm Start SNMP Trap/Inform messages. After receiving the messages, the mediation server can start the reconfiguration of LI tasks.

In the following example, the <ipaddress> is the IP address of the Mediation Server:

```
ibos(config)# snmp-group version 2c
ibos(config-snmp-group)# notify <ipaddress> community public trap
```

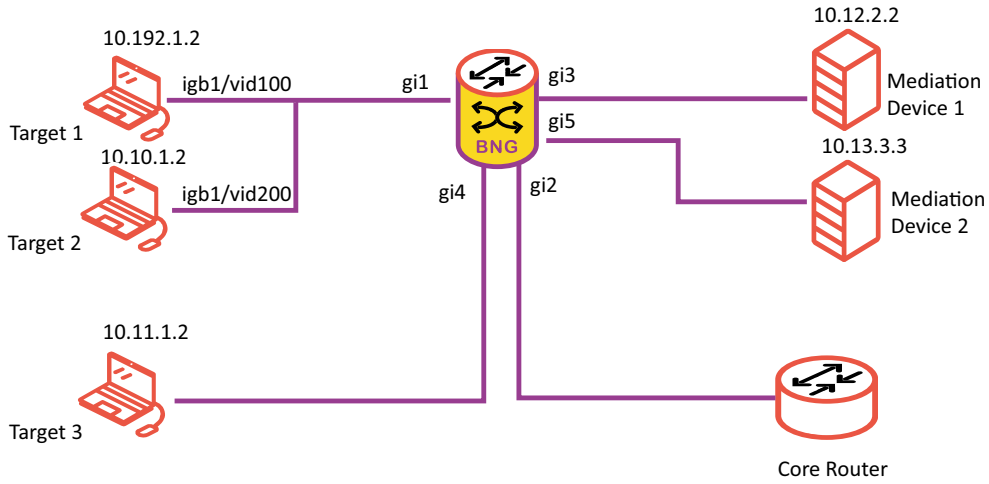
Configuring SNMP trap interface is performed by the system admin.

For more information about SNMP trap configuration, refer to section “Simple Network Management Protocol” and *iBOS Command Reference*.

Configuration Examples

To explain the concept of LI in a more detailed manner, let us consider the following use cases from the example diagram shown.

Figure 128 Configuration Example



USE CASE 1

Consider the case wherein an LI user wants to intercept the in and out traffic of Target1 using IP address as filtering key and report the intercepted packets to Mediation Server 10.12.2.2.

The procedural steps to be followed for interception includes:

- Verify whether the iBOS version have support for Lawful Intercept feature.

```
ibos# show features | grep LAWFUL
LAWFUL_INTERCEPT        Lawful Intercept support
```

- LI user account to be created by the system administrator in order to login the system.

```
ibos(config)# username lawful role li password ******
```

- System administrator to configure system intercept/filtering mode to IP address:

```
ibos(config)# lawful-intercept filtering-mode ipaddress
```

- Optionally, system admin may configure global LI parameters:

```
ibos(config)# [no] lawful-intercept tos 200
ibos(config)# [no] lawful-intercept ttl 200
ibos(config)# [no] lawful-intercept mtu 1536
```

- System admin should add the Mediation Server(s) as SNMP trap receivers

```
ibos(config)# snmp-group version 2c
ibos(config-snmp-group)# notify 10.12.2.2 community public trap
```

- If any changes, the system admin must save the configuration:

```
ibos# write
```

- LI user to login the system using SSH (not TELNET or CONSOLE).
- It is encouraged that the LI user resets the password set by the system admin.

```
ibos(config)# reset password ****** [encrypted]
```

- LI user creates an LI task to monitor target1, using IP address 10.192.1.2, and to report the intercepted packets to Mediation Server 10.12.2.2:49153

```
ibos(config)# lawful-intercept task interceptid 1 mediation-server
10.12.2.2 port 49153 target-ip 10.192.1.2
```

- Monitoring a target belonging to a specific VRF:

```
ibos(config)# lawful-intercept task interceptid 3 mediation-server
10.12.2.2 port 49153 vrf north target-ip 10.10.1.2
```

- Show running config

```
#
# output omitted for brevity
#
!volatile lawful-intercept task interceptid 1 mediation-server
10.12.2.2 port 49153 target-ip 10.192.1.2
!volatile lawful-intercept task interceptid 3 mediation-server
10.12.2.2 port 49153 vrf north target-ip 10.10.1.2
#
# output omitted for brevity
#
```

Finally, to show the status of the active LI tasks, use the following command to display brief or detailed information (only LI tasks owned by the LI user will be shown):

```
ibos# show lawful-intercept tasks brief
Total LI tasks for current user: 2

InterceptId       MediationServer        Target
-------------------------------------------------
1                 10.12.2.2:49153        10.192.1.2
3                 10.12.2.2:49153        10.10.1.2


ibos# show lawful-intercept tasks detailed
Total LI tasks for current user: 2

InterceptId....: 1
MediationServer: 10.12.2.2:49153
Target.........: 10.192.1.2
TunnelStatus...: UP
SessionId......: 200729830
Created Time...: Sep 29 2017 12:11:53
InboundPkts....: 5237
OutboundPkts...: 3911
```

USE CASE 2:

Consider the case wherein an LI user wants to intercept the in and out traffic of Target 3 using interface as filtering key and report the intercepted packets to Mediation Server 10.13.3.3.

The procedural steps to be followed for interception includes:

- Verify whether the iBOS version have support for Lawful Intercept feature.

```
ibos# show features | grep LAWFUL
LAWFUL_INTERCEPT        Lawful Intercept support
```

- LI user account to be created by the system administrator in order to login the system.

```
ibos(config)# username lawful role li password ******
```

- System administrator to configure system intercept/filtering mode to interface:

```
ibos(config)# no lawful-intercept filtering-mode ipaddress
```

- Optionally, system admin may configure global LI parameters:

```
ibos(config)# [no] lawful-intercept tos 200
ibos(config)# [no] lawful-intercept ttl 200
```

```
ibos(config)# [no] lawful-intercept mtu 1536
```

- System admin should add the Mediation Server(s) as SNMP trap receivers

```
ibos(config)# snmp-group version 2c
ibos(config-snmp-group)# notify 10.13.3.3 community public trap
```

- If any changes, the system admin user must save the configuration:

```
ibos# write
```

- LI user to login the system using SSH (not TELNET or CONSOLE).
- It is encouraged that the LI user resets the password set by the system admin.

```
ibos(config)# reset password ****** [encrypted]
```

- LI user creates an LI task to monitor target3, which is connected to interface gigabitethernet4, and to report the intercepted packets to Mediation Server 10.13.3.3:49154

```
ibos(config)# lawful-intercept task interceptid 2 mediation-server
10.13.3.3 port 49154 target-interface gigabitethernet4
```

- Show running config

```
#
# output omitted for brevity
#
!volatile lawful-intercept task interceptid 2 mediation-server
10.13.3.3 port 49154 target-interface gigabitethernet4
```

Finally, to show the status of the active LI tasks, use the following command to display brief or detailed information (only LI tasks owned by the LI user will be shown):

```
ibos# show lawful-intercept tasks brief
Total LI tasks for current user: 1

InterceptId       MediationServer       Target
-----------------------------------------------------------
2                 10.13.3.3:49154       gigabitethernet4

ibos# show lawful-intercept tasks detailed
Total LI tasks for current user: 1

InterceptId....: 2
MediationServer: 10.13.3.3:49154
Target.........: gigabitethernet4
SessionId......: 8379211
TunnelStatus...: UP
Created Time...: Sep 29 2017 12:17:43
InboundPkts....: 4328
OutboundPkts...: 5686
```

Selective Q-in-Q

Selective Q-in-Q adds a second Q-in-Q inner tag that makes it possible to encapsulate information that can be used by service providers to provision multiple services to tagged VLAN clients. The outer tag will then be used for the service and ISP (S-TAG) while the inner tag is used for the customer interface (C-TAG).

To configure the second Q-in-Q tag, the **vlan q-in-q next-tag** command is used.

For more details, refer *iBOS Command Reference*.

iBOS Licenses

Pay as you Grow

Waystream products have optional features that by default are enabled for a grace period and will require additional licenses to continue functioning. The license enables a maximum number of instances of the licensed feature. Once this limit is exceeded, a new license must be purchased for the superfluous instances to continue functioning. In this way, iBOS provides a "pay-as-you-grow" model that extends the default functionality as, and when, the network needs it.

Table 43 iBOS licenses

License type	iBOS function	Description
ibos-lic-mpls-pw	mpls pw	License for MPLS pseusdowire
ibos-lic-vrf	vrf	License for VRF
ibos-lic-remote-vid	remote-vid	License for remote VLAN interface
ibos-lic-rate-limit	rate limiter	License for rate limiter

To activate a license on a device, a unique key file needs to be generated and installed locally. The key file is tied to the device MAC address and can have a defined expiry date after which the license will no longer be valid, as well as a limit to the number of instances for the feature.

Grace Period

Grace period is five (5) days and starts to countdown when one of the following conditions occurs:

- if number of resources configured exceeds the maximum number allowed
- if a license type expires
- if resources are configured without a valid license installed

Note: *When the grace period expires, services using features without valid licenses will be severely throttled or even stop functioning. If the grace period needs to be longer than five days, contact your Waystream sales representative.*

Verify Licenses

The simplest way to check if a valid license is installed is to use the **show license** command on the port requiring the optional feature:

```
mpc# show license
  grace period: Available

  license type: ibos-lic-mpls-pw

  license type: ibos-lic-vrf (NOT INSTALLED)

  license type: ibos-lic-remote-vid (NOT INSTALLED)

  license type: ibos-lic-rate-limit (NOT INSTALLED)
```

To see more detailed information about installed licenses, use the **show license detailed**:

```
mpc# show license
  grace period: Available

  license type: ibos-lic-mpls-pw

  license type: ibos-lic-vrf (NOT INSTALLED)

  license type: ibos-lic-remote-vid (NOT INSTALLED)

  license type: ibos-lic-rate-limit (NOT INSTALLED)

mpc# show license detailed
  grace period: Available

  license type: ibos-lic-mpls-pw
    support function: mpls-pw
    generate time: 2017-02-07
    expire time: 2036-12-31
    max count: 10
    granted: 2

  license type: ibos-lic-vrf (NOT INSTALLED)
    support function: vrf
    generate time:
    expire time:
    max count:
    granted: 0

  license type: ibos-lic-remote-vid (NOT INSTALLED)
    support function: remote-vid
    generate time:
    expire time:
    max count:
    granted: 0

  license type: ibos-lic-rate-limit (NOT INSTALLED)
    support function: rate-limit
    generate time:
    expire time:
    max count:
    granted: 0

mpc#
```

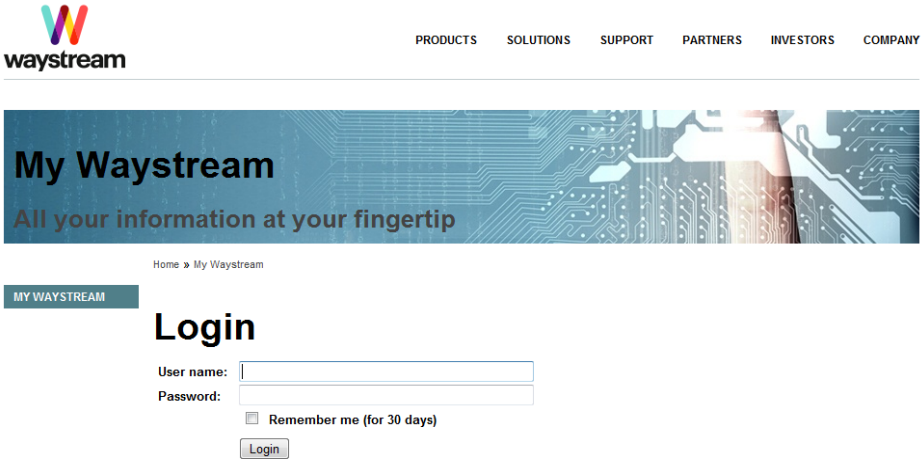
The following iBOS commands are used to configure licenses:

- **license**
- **show license**
- **license reset-grace-period**
- **license disable-granted**

Fetch your License Files

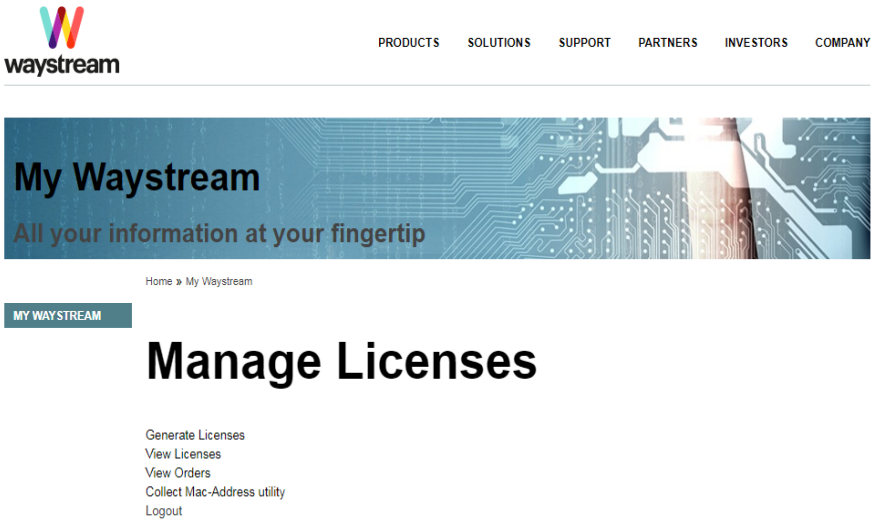
Once a license has been purchased, the license file can be generated and downloaded from at **http://license.waystream.com**. You can either register a single device, submit a file with multiple MAC addresses, or collect iBOS hardware MAC-addresses using a python script that can be found on the web page.

Figure 129 Generate iBOS License File



Login to the website using your support user name and password that you have previously obtained. You will be directed to the Manage licenses web page.

Figure 130 Manage iBOS License File

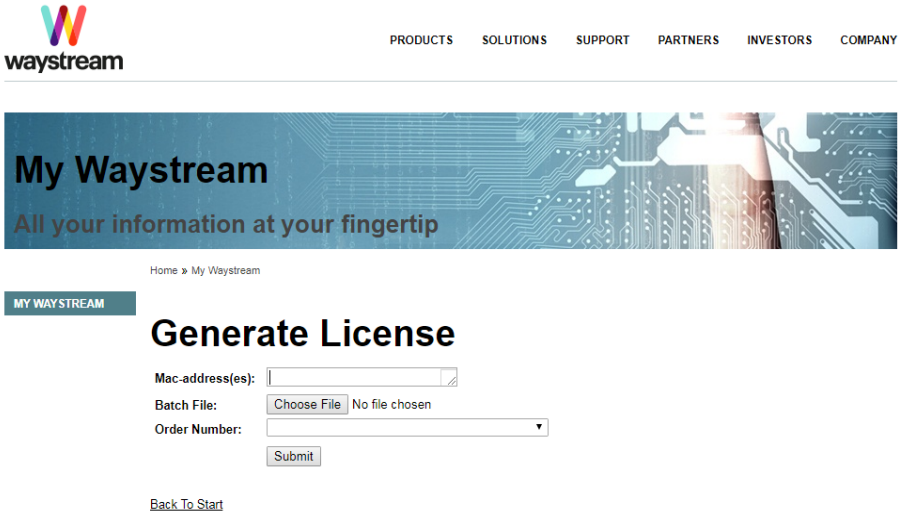


To generate license files, select the Generate licenses option.

Generate Licenses

To generate license files, you need to submit the MAC address for any device that requires an iBOS license. Each device requires a unique license file, for example a switch with MAC address 00:08:AE:00:23:45 will have a license file called *ibos-0000A8E02345.lic*.

Figure 131 Generate Licenses Screen



For a single license, enter the MAC address. For multiple licenses, you can either enter each MAC address manually type and/or paste text strings into the Mac-address field, or you can upload a file with the MAC addresses of multiple iBOS devices using the **Choose file** button. The file is a simple text file with one MAC address per line, for example:

```
00:08:AE:00:23:45
00:08:AE:00:23:46
00:08:AE:00:23:47
```

Note: *For MAC addresses, the delimiters must be either colon (:) or period (.) and valid characters are 0-9 and A-Z.*

The next step is to identify the purchased license by selecting the order number from the Order number drop-down list. The drop-down list is automatically populated with order numbers for your company.

To generate the file, click the Submit button and the file will be created. Multiple license files will be compressed into a single zip file (default name is *licenses.zip*).

Note: *All license files must be extracted from the zip file archive before they can be installed by iBOS.*

Installing Licenses

To install a license file on the switch, the iBOS command **license url** is used. iBOS fetches the license file from the defined location and writes the encrypted license key to flash. When a license has been activated, iBOS will check for new licenses periodically and automatically update the license file if a more recent file is found at the specified location.

Example 89 Install License from TFTP Server

```
mpc480(config)# license url tftp://10.128.1.1/ibos-
0008AE002345.lic
```

If there is no TFTP server available, it is possible to manually enter the license key using **license set**. Save the license key by entering a period (".")

Example 90 Enter License Key

```
mpc# license set
% Copy and Paste license file text and enter "." on a new line to
save the license to flash memory
mac:0008AE8F1380;lic:ibos-lic-vrf;cre:2017-05-18;exp:2036-12-
30;cnt:1;sig:f8VtmK
P9+lL7TkM5AoFl5UvetT9kR5XV4N/
E2vwpEdpETK7abrYUU8xUnuq2mH48yUnFYKP+zzk1USVRj/5XjQ
2DBtLy/k24qhg+ms5udrG92JCnc8omfM/JKHeQb8a681ry5dGJOG0tvtnSfDC/
4dud5U5Brkkb1G10/g
6JatJiwJWX2EgzBrxi2HX670X4IhWiJSUF4tjDJFyGyf1HVV1RiCzUnNkGHrjVF/
LXwTPsA6dbqgXDNp
DsJaXMCr6eWq8RdN+cwccjpzWdUWGILcZPWeXOHAYXkVi8G/
kbOtuyO0V3qYUJOIr6BwiFTRqc/4/ad9
dPMAA1VIAZVchEDA==;
mac:0008AE8F1380;lic:ibos-lic-mpls-pw;cre:2017-05-17;exp:2036-12-
```

```
30;cnt:1;sig:SE
issC8XDCLnJs9RtH7yc+xIZ+aPuH9XW9QEOdH2vF7jA47Yl4Rxm7IyZNjEps70Motv
PduddCCfXgtYLJ
hc1mvK9IDRHGRsMlNCDRBNfRZeAh8OtpdJK7zIi9K+kF+OAsxzfjvxiyqlZJYR2k5L
mptpDiOHyxBE/J
C7ayTOWJp7dPlzUI5lxtaivkeWCPUxMKT5dmgUAiTeSil8lDr1e8DTdjS22nlnM8Vc
uf1GN2GKxF+UVW
vPlYBVHwfKEZXglyfsAEzVoOojowgjPYJpeI+5y82gaVo0azRkIG/
TAd8NemAPBxrA+ymEZtK2LFIybv
WH3LRq2fBv2VMj/pOD6Q==;
.
00:01:19: %LICENSEMGR-5-LIC_INSTALL: License installed to flash
memory.
mpc#
```

To see which licenses are installed, the command **show license** is used.

Example 91 Show Installed Licenses

```
mpc480> show license
  license type: ibos-lic-advanced

mpc480> show license  detailed
  license type: ibos-lic-advanced
  support function: remote-mirror | wfq | rpm | selective-q-in-q |
scribos
  generate time: 2013-06-13
  expire time: 2013-07-17
```

For details about the commands **license url** and **show license**, refer *iBOS Command Reference*.

Appendix A – Copyright Statements

Waystream software products may contain contributed software with other copyright owners, in part or in whole.

NetBSD

```
Copyright 1982-1993 The Regents of the University of California.
All rights reserved.

Copyright 1994-2001 The NetBSD Foundation, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
   distribution.
3. All advertising materials mentioning features or use of this
   software must display the following acknowledgement:
     This product includes software developed by the NetBSD
     Foundation, Inc. and its contributors.
4. Neither the name of The NetBSD Foundation nor the names of its
   contributors may be used to endorse or promote products derived
   from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND
CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Net-SNMP

Copyright 1989, 1991, 1992 by Carnegie Mellon University

Derivative Work - 1996, 1998-2000
Copyright 1996, 1998-2000 The Regents of the University of California

All Rights Reserved

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2001-2002, Networks Associates Technology, Inc
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Networks Associates Technology, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

```
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this code are copyright (c) 2001-2002, Cambridge
Broadband Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

*  Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

*  Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.

*  The name of Cambridge Broadband Ltd. may not be used to endorse or
promote products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ``AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT HOLDER BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Internet Standards Document Copyright

Throughout this documentation references and citations are made to Request For Comments (RFC) or Internet-Draft documents. RFCs are Internet standard documents published by the Internet Society. Internet-Drafts are working documents published by the Internet Society. The following copyright information pertains to all such documentation, references and citations made in this document.

```
Copyright © The Internet Society. All rights Reserved.
This document and translations of it may be copied and furnished to
others, and derivative works that comment on or otherwise explain
it or assist in its implementation may be prepared, copied,
published and distributed, in whole or in part, without restriction
of any kind, provided that the above copyright notice and this
paragraph are included on all such copies and derivative works.
However, this document itself may not be modified in any way, such
as by removing the copyright notice or references to the Internet
Society or other Internet organizations, except as needed for the
purpose of developing Internet standards in which case the
```

procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

waystream

www.waystream.com