ZebOS-XP®
Network Platform

Version 1.4
Extended Performance

Troubleshooting Guide

December 2015

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900
http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:
support@ipinfusion.com

Contents

Preface

This guide contains tips for troubleshooting issues when building, configuring, and managing ZebOS-XP.

Audience

This guide is intended for network administrators, developers, and other engineering professionals.

Conventions

Table P-1 shows the conventions used in this guide.

Table P-1: Conventions

Convention	Description
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, functions, parameters, files, and directories

Contents

This guide contains these chapters and appendices:

- Chapter 1, *Debugging and Logging*
- Chapter 2, *Troubleshooting BGP*
- Chapter 3, *Troubleshooting OSPF*
- Chapter 4, *Troubleshooting RIP*
- Chapter 5, *Troubleshooting LDP*
- Chapter 6, *Troubleshooting VRRP*
- Chapter 7, *Troubleshooting PIM*
- Chapter 8, *Miscellaneous Issues*
- Chapter 9, *Frequently-Asked Questions*

Related Documents

Note: All ZebOS-XP technical manuals are available to licensed customers at http://www.ipinfusion.com/support/document_list.

Support

For support-related questions, contact support@ipinfusion.com.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

CHAPTER 1 Debugging and Logging

ZebOS-XP has a comprehensive debugging and logging facility in various protocols and components. This chapter describes how to start and stop debugging and logging. For complete information, see the *Network Services Module Command Reference*. The protocol debug commands are in the corresponding command references.

About Debugging

In ZebOS-XP, every protocol has debug commands that log parameter-specific information. For example, using the `debug ldp nsm` command results in ZebOS-XP writing all messages exchanged between LDP and NSM such as interface, bandwidth, and address updates.

You can direct the output from the debug command to:

- Standard output (stdout)
- A file
- The system log (Linux syslog)

ZebOS-XP generates debug output until the `no` form of the `debug` command is given.

Start Debugging Output

To start debugging output, turn on the debug options by giving the relevant `debug` command. For example:

```
> enable
# configure terminal
(config)# log file <filename>
(config)# debug <protocol> (parameter)
(config)# exit
```

Log to Standard Output

To direct debugging output to `stdout`, give the `terminal monitor` command.

```
# terminal monitor
```

This is a sample output of the `debug rsvp events` command displayed on the terminal:

```
# terminal monitor
```

Dec 2 16:41:49 localhost RSVP[6518]: RSVP: RSVP message sent to 10.10.23.60/32 via interface eth0

Dec 2 16:41:57 localhost RSVP[6518]: RSVP: Received an RSVP message of type RSVP Reservation from 192.168.0.60 via interface eth0

Dec 2 16:41:57 localhost RSVP[6518]: RSVP: Received a RESV message from 10.10.23.60/32

Log to a File

To send debugging output to a file:

1. Use the `log file` command and specify the path and file name where the information is to be logged.

   When logging to a file, you can simultaneously log to stdout by using the `terminal monitor` command.

2. Use the `no` form of the command to turn off logging to a file:

```
(config)# no log file (filename)
```

Log to the System Log

To send debugging output to syslog:

1. Use the `log syslog` command:

```
(config)# log syslog
```

2. Use the `no` form of the command to turn off system logging:

```
(config)# no log syslog
```

Stop Debugging

To turn off debugging, use the `no debug` or `undebug` command. When a protocol is specified with the `no debug` or `undebug` commands, debugging is stopped for the specified protocol. To stop all debugging, use the `all` parameter with these commands.

```
(config)# no debug bgp events
```

or

```
# undebug all
```

CHAPTER 2 Troubleshooting BGP

This chapter contains steps for resolving BGP issues.

Refer to the *Border Gateway Protocol Command Reference* for details about commands used in this chapter.

No BGP Adjacency

1. Use the `show ip interface brief` command to make sure that the interface is not administratively shutdown. Remove this configuration setting with the `no shutdown` command, if `shutdown` is configured.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that the BGP configuration is correct. To establish BGP, configure a TCP session with another router using the `neighbor remote-as` command.

   When using iBGP:

   - Make sure the two routers know how to reach each other's loopback addresses, if you have established iBGP using loopback interface. Typically, you have an IGP (say OSPF) running between the two routers. In this case, enable OSPF on the loopback interface or redistribute the loopback address into OSPF.
   - Ping to each other's loopback address to ensure mutual reachability.

   When using eBGP:

   - Make sure you have configured the multihop number for an eBGP neighbor that is not directly connected.
   - Use the `neighbor ebgp-multihop` command to specify the maximum hop count to reach the neighbor.

3. Make sure you can reach the neighbor using the `ping A.B.C.D` command.

4. Verify if a firewall is present. If there is a firewall, it might be configured to block TCP packets. Verify the existing firewall configurations (in Linux) by using:

   ```
   ipchains -L
   ```

   Flush the existing firewall configurations by using:

   ```
   ipchains -F
   ```

No BGP4+ Adjacency

1. Use the `show ipv6 interface brief` command to make sure that the interface is not administratively shutdown. Remove this configuration setting with the `no shutdown` command, if `shutdown` is configured.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that the configuration is correct. To establish BGP, configure a TCP session with another router using the `neighbor remote-as` command. See the *Border Gateway Protocol Command Reference* for details about this command.

   When using iBGP:

   - Make sure the two routers know how to reach each other's loopback address, if you have established iBGP using loopback interface. Typically, you have an IGP (such as OSPF) running between the two routers. In this case, enable OSPF on the loopback interface or redistribute the loopback address into OSPF.
   - Ping to each other's loopback address to ensure mutual reachability.

   When using eBGP:

   - Make sure you have configured the multihop number for an eBGP neighbor that is not directly connected.
   - Use the `neighbor ebgp-multihop` command to specify the maximum hop count to reach the neighbor.

3. Make sure you can ping to the neighbor using the `ping A.B.C.D` command.

4. Verify if a firewall is present. If there is a firewall, it might be configured to block TCP packets. Verify the existing firewall configurations (in Linux) by using:

   ```
   ipchains -L
   ```

   Flush the existing firewall configurations by using:

   ```
   ipchains -F
   ```

CHAPTER 3 Troubleshooting OSPF

This chapter contains steps for resolving OSPF issues.

Refer to the *Open Shortest Path First Command Reference* for details about the commands used in this chapter.

No OSPFv2 Adjacency

1. Use the `show ip interface brief` command to make sure that the interface is not administratively shut down. Remove this configuration setting with the `no shutdown` command, if `shutdown` is configured.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that OSPF is enabled on the interface. To enable OSPF on a particular interface, use the `network area` command with a specified Area ID. Use the `show ip ospf interface` to confirm that OSPF is enabled for the interface.

   ```
   eth2 is up, line protocol is up
    Internet Address 56.168.1.7/24, Area 0.0.0.0, MTU 1500
    Router ID 7.7.7.7, Network Type BROADCAST, Cost: 10
    Transmit Delay is 1 sec, State DR, Priority 1
      Designated Router (ID) 7.7.7.7, Interface Address 56.168.1.7
      Backup Designated Router (ID) 8.8.8.8, Interface Address 56.168.1.8
      Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
       Hello due in 00:00:05
    Neighbor Count is 1, Adjacent neighbor count is 1
    Crypt Sequence Number is 0
    Hello received 625 sent 645, DD received 3 sent 4
    LS-Req received 1 sent 1, LS-Upd received 5 sent 13
    LS-Ack received 8 sent 5, Discarded 0
   ```

3. Ensure that the interface is not configured as a passive using `show run`:

   ```
   !
   router ospf
   passive interface eth0
   !
   ```

   If the interface is passive, remove this configuration setting by using this command:

   ```
   no passive interface eth0
   ```

4. Check the interface to make sure that OSPF Hello packets are being sent and received correctly. You can use either a packet sniffer (such as Ethereal or TCP dump) or ZebOS-XP log messages to verify the hello packet.

   To turn on ZebOS-XP logging, type:

   ```
   # configure terminal
   (config)# debug ospf event
   (config)# debug ospf packet hello
   ```

   To display the logging message on the terminal, type:

   ```
   # terminal monitor
   ```

5. It is possible that there is a mismatch between Hello parameters. Make sure that you have specified the same hello interval and dead interval values on both machines by using the `show ip ospf interface` command on each machine.

6. Run `show ip ospf neighbor` if you see the neighbor but the state is not full.

   Make sure that both routers have the same MTU size for the interfaces.

No OSPFv3 Adjacency

1. Use the `show ipv6 interface brief` command to make sure that the interface is not administratively shutdown. Remove this configuration setting with the `no shutdown` command, if `shutdown` is configured.

```
# configure terminal
(config)# interface eth0
(config-if)# no shutdown
```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that OSPF is enabled on the interface. To enable OSPF on a particular interface, use the `ipv6 router ospf area` command with a specified Area ID. Use the `show ipv6 ospf interface` to confirm that OSPF is enabled for the interface.

```
# show ipv6 ospf interface
    eth0 is up, line protocol is up
      Interface ID 3, Instance ID 0, Area 0.0.0.0
      IPv6 Link-Local Address fe80::248:54ff:fec0:f32d/10
        Router ID 1.2.3.4, Network Type BROADCAST, Cost: 10
        Transmit Delay is 1 sec, State Backup, Priority 1
        Designated Router (ID) 5.6.7.8
          Interface Address fe80::203:47ff:fe4c:776e
        Backup Designated Router (ID) 1.2.3.4
          Interface Address fe80::248:54ff:fec0:f32d
        Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
          Hello due in 00:00:01
        Neighbor Count is 1, Adjacent neighbor count is 1
    lo is up, line protocol is up
      OSPFv3 not enabled on this interface
    sit0 is down, line protocol is down
      OSPFv3 not enabled on this interface
```

3. Ensure that interface is not configured as a passive interface using `show run`:

```
!
router ipv6 ospf
passive interface eth0
!
```

   If the interface is configured as passive (as shown above), remove this configuration setting by using this command:

```
no passive interface eth0
```

4. Check on the interface to make sure that OSPF Hello packets are being sent and received on the interface. You can use either packet sniffer (such as, Ethereal or TCP dump) or ZebOS-XP log messages to verify the Hello packets.

   To turn on ZebOS-XP logging, type:

```
# configure terminal
(config)# debug ipv6 ospf event
```

```
(config)# debug ipv6 ospf packet hello
```

To display the logging message on the terminal, type:

```
# terminal monitor
```

5. It is possible that there is a mismatch between Hello parameters. Make sure that you have specified the same `hello interval` and `dead interval` values on both machines by using the `show ipv6 ospf interface` command.

6. Run the `show ipv6 ospf neighbor` command; if you see the neighbor but the state is not full, make sure that both routers have the same MTU size for the interfaces.

7. Verify if a firewall is present. If there is a firewall, it blocks the UDP packet. You must remove the firewall if you have one. To display the existing firewall configurations, in Linux, use:

   ```
   ipchains -L
   ```

   Flush the existing firewall configurations by using:

   ```
   ipchains -F
   ```

CHAPTER 4 Troubleshooting RIP

This chapter contains steps for resolving RIP issues.

Refer to the *Routing Information Protocol Command Reference* for details about commands used in this chapter.

No RIP Adjacency

1. Use the `show ip interface brief` command to make sure that the interface is not administratively shutdown. Remove this configuration using the `no shutdown` command, if `shutdown` is configured.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Confirm that RIP is enabled on the interface. To enable RIP on a particular interface, use the `network` command. Use the `show ip rip interface` to make sure that RIP is enabled for the interface.

   ```
   # show ip rip interface
   fxp0 is up, line protocol is up
    Routing Protocol: RIP
       Receive RIP packets
       Send RIP packets
       Passive interface: Disabled
       Split horizon: Enabled with Poisoned Reversed
       IP interface address:
         10.15.0.60/16
   ```

3. Make sure that the interface is not configured as a passive interface using the `show run` command:

   ```
   !
   router rip
   passive interface eth0
   !
   ```

   If the interface is configured as passive (as shown above), remove this configuration setting by using this command:

   ```
   no passive interface eth0
   ```

4. Make sure that RIP advertisements are being sent and received on the interface. You can use either a packet sniffer (such as, Ethereal or TCP dump) or the ZebOS-XP log messages to verify the RIP advertisements.

   To turn on ZebOS-XP logging, type:

   ```
   # configure terminal
   (config)# debug rip event
   (config)# debug rip packet detail
   ```

   To display the logging message on the terminal, type:

   ```
   # terminal monitor
   ```

5. One router configured as RIPv1 and the other router as RIPv2 results in no RIP adjacency.

Configure the router running RIPv2 as follows:

```
!
interface eth1
ip rip send version 1-compatible
ip rip receive version 1 2
!
```

1. Verify whether a firewall is present. If there is a firewall, it blocks the UDP packet. You must remove the firewall if you have one. To display the existing firewall configurations, in Linux, use:

   ```
   ipchains -L
   ```

   Flush the existing firewall configurations by using:

   ```
   ipchains -F
   ```

No RIPng Adjacency

1. Use the `show ipv6 interface brief` command to make sure that the interface is not administratively shutdown. Remove this configuration by `no shutdown` command, if `shutdown` is configured.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that RIPng is enabled on the interface. To enable RIPng on a particular interface, use the `ipv6 router rip` command. Use the `show ipv6 rip interface` to confirm that RIP is enabled for the interface.

   ```
   # show ipv6 rip interface eth1
   eth1 is up, line protocol is up
     Routing Protocol: RIPng
       Passive interface: Disabled
       Split horizon: Enabled with Poisoned Reversed
       IPv6 interface address:
         3ffe:1::10/64
         fe80::204:76ff:fec8:28cc/10
   ```

3. Make sure that interface is not configured as a passive interface using the `show run` command:

   ```
   !
   router ipv6 rip
   passive interface eth0
   !
   ```

   If the interface is configured as passive (as shown above), remove this configuration setting by using this command:

   ```
   no passive interface eth0
   ```

4. Check on the interface to make sure that RIPng advertisements are being sent and received on the interface. You can use either a packet sniffer (such as, Ethereal or TCP dump) or the ZebOS-XP log messages to verify the RIPng advertisements.

   To turn on ZebOS-XP logging, type:

   ```
   # configure terminal
   (config)# debug ipv6 rip event
   (config)# debug ipv6 rip packet detail
   ```

   To display the logging message on the terminal, type:

   ```
   # terminal monitor
   ```

5. Verify if a firewall is present. If there is a firewall, it blocks the UDP packet. You must remove the firewall if you have one. To display the existing firewall configurations, in Linux, use:

   ```
   ipchains -L
   ```

   Flush the existing firewall configurations by using:

   ```
   ipchains -F
   ```

CHAPTER 5 Troubleshooting LDP

This chapter contains steps for resolving LDP issues.

Refer to the *Label Distribution Protocol Command Reference* for details about commands used in this chapter.

LDP Session is not UP

1. Make sure that the transport address is selected correctly by the system. The transport address is the IP address that the LDP router advertises in its `Hello` message so that other routers can communicate using this IP address while setting up a TCP connection for the LDP session.

   IP Infusion Inc. recommends that the loopback address be used as a transport address so that link loss on any one interface does not cause the LDP session to go down. Use `show ldp` to verify the status of the transport address:

   ```
   Router# show ldp
   Router ID                : 192.168.0.42
   LDP Version              : 1
   Global Merge Capability  : N/A
   Label Advertisement Mode : Downstream Unsolicited
   Label Retention Mode     : Liberal
   ...
   Targeted Hello Receipt   : Disabled
   Transport Address data   :
     Labelspace 0           : 192.168.0.42 (in use)
   Import BGP routes        : No
   ```

   In this show output, the transport address data indicates that 192.168.0.42 is the transport address and is using 0 labelspace.

   When loopback address is configured, it is automatically selected as transport address. If no loopback address is available, the LDP process selects the first available physical interface IP address.

2. When the transport address is verified to be correct on the peering systems, make sure that the transport address of the remote peer is reachable from both the routers. To verify that the remote transport address is reachable, ping the remote transport address. Use `show ip route` and check if there is a route to the neighbor's transport address.

   ```
   ....
   S    23.23.23.0/24 [1/0] is directly connected, eth0
   O    48.48.48.48/32 [110/30] via 172.168.27.82, eth1, 15:44:20
   C    81.81.81.81/32 is directly connected, lo
   O    82.82.82.82/32 [110/20] via 172.168.27.82, eth1, 17:45:31
   C    127.0.0.0/8 is directly connected, lo
   ...
   ```

   In the output above, `82.82.82.82/32` is the neighbor's transport address and can be reached via `172.168.27.82`.

   If there is no route to the neighbor's transport address:

   Add a static route to the neighbor's transport address.

   Or

   Use an IGP (OSPF) to reach the neighbor's transport address.

3. Use the `show ldp interface` command to verify the status of LDP.

```
Interface     LDP Identifier  Label-switching  Merge Capability
lo            10.30.0.16:0    Disabled            N/A
eth0          10.30.0.16:0    Disabled            N/A
eth1          10.30.0.16:0    Enabled          Merge capable
eth2          10.30.0.16:0    Enabled          Merge capable
```

   If the label-switching is disabled on an interface, use the `show run` command to ensure that LDP and label-switching are enabled on that interface. The configuration should show as follows:

```
!
interface eth1
 label-switching
 ip address 172.168.27.81/24
 enable-ldp
!
```

4. Use the `show run` command to ensure that targeted peer hello receipt is enabled.

   If not, then run the `targeted-peer-hello-receipt` command to configure the LSR to respond to requests for targeted hello messages.

CHAPTER 6 Troubleshooting VRRP

This chapter contains steps for resolving VRRP issues.

See the *Virtual Router Redundancy Protocol Command Reference* for details about commands used in this chapter.

Topology

The following topology is used for illustration purposes.

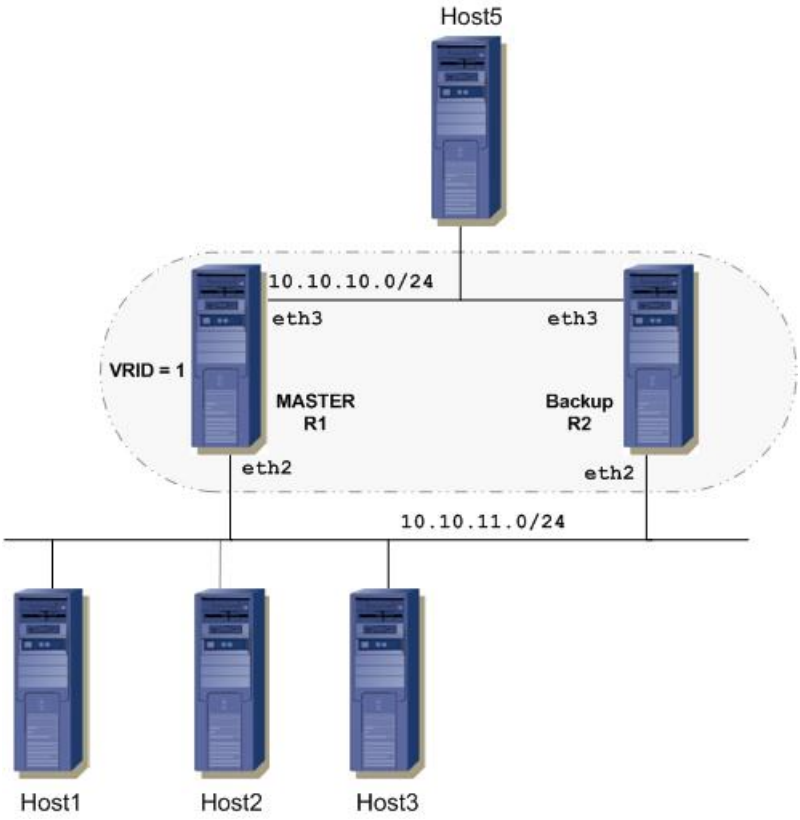


Figure 6-1: VRRP Topology

Incorrect VRRP States

1. Make sure the interfaces are up and running by using the `show interface` command. In the following sample output interface `eth1` is:

```
# show interface eth1
Interface eth1
  Hardware is Ethernet, address is 0002.b3d4.436f
  index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  ...
  input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0..
```

If the interface is down:

Use `no shutdown` command in `interface` mode to bring up the interface.

Or

Use the `ifconfig IFNAME up` command to bring up the interface.

2. Make sure that both VRRP routers can reach each other by pinging.

```
# ping 10.10.11.2
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.
64 bytes from 10.10.11.2: icmp_seq=1 ttl=255 time=0.202 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=255 time=0.201 ms
```

If both routers cannot reach each other, check the network connections for the default Master and default Backup routers.

3. Check the advertisement interval on Master and Backup routers. The advertisement interval must be the same on both.

The default advertisement interval = 1.

Use the `advertisement-interval` command in router mode to configure the advertisement interval.

CHAPTER 7 Troubleshooting PIM

This chapter contains steps for resolving PIM issues.

See the *Protocol Independent Multicasting Command Reference* for details about commands used in this chapter.

No PIM adjacency

1. Use the `show run` command to make sure that the interface is not administratively shutdown. If `shutdown` is configured, remove this configuration with the `no shutdown` command.

   ```
   # configure terminal
   (config)# interface eth0
   (config-if)# no shutdown
   ```

   Use the `show interface` command to make sure that the interface is up.

2. Make sure that PIM is enabled on the interface by using the `show ip pim sparse-mode interface` command.

3. If you are trying to establish adjacency between ZebOS-XP and Cisco and are not successful, use the `ip pim exclude-genid` command on the interface. Some old Cisco IOS do not recognize the `GenID` option in the PIM Hello packet and discard the packet.

No BSR and RP information

Check your Unicast routing configuration to make sure that you can reach BSR and RP. Use the `show ip route` command to display the unicast routing table.

RP not advertised in the BSR

This happens when Cisco is a BSR and ZebOS-XP is the candidate RP. In this case, you must give the following command on the ZebOS-XP candidate RP router:

```
ip pim crp-cisco-prefix
```

Double Multicast Traffic

In this example, PIM is running on all routers, ZebOS-XP and the Check Point FireWall are running on R3. NAT is configured to translate Source address of the packets coming from `eth1` to `eth2` interface address. The multicast Source address is `192.168.2.2`. R2 is the RP and has a Receiver attached to it.

The Source router sends multicast data packets to R3. R3 acts as the Designated Router (DR) of the Source and sends unicast Register packets to the RP. At the Receiver end, double multicast traffic is coming out. For each multicast packet sent by Source, the Receiver is receiving two copies of the packet instead of one.

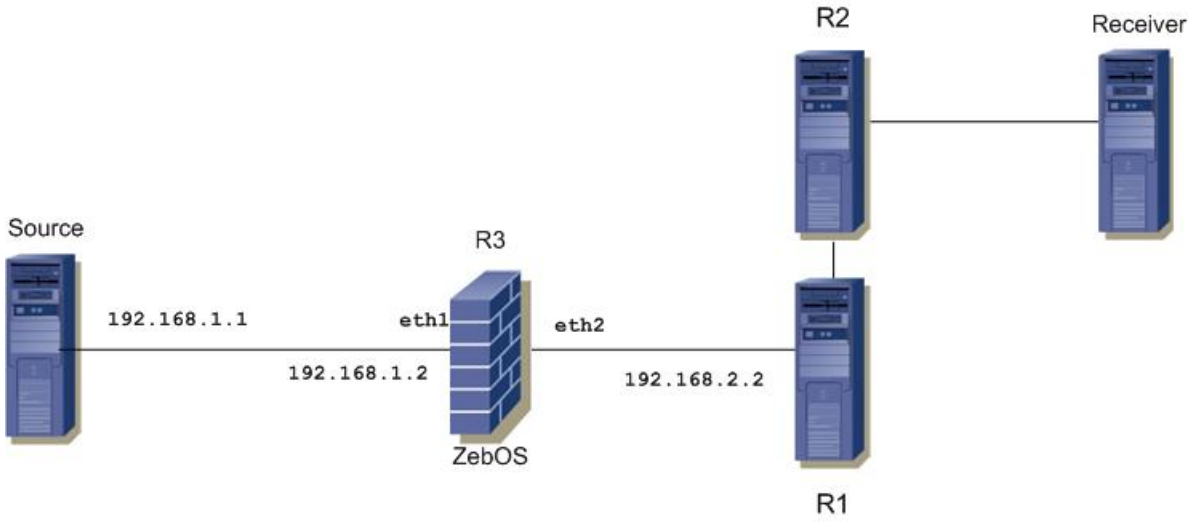


Figure 7-1: PIM-SIM Topology

1. Using your packet sniffer, observe the packets sent out by R3. Notice that both native multicast traffic and Register packets come out from `eth2` and the Register packet encapsulates the multicast packet.

2. Check the ZebOS-XP multicast routing table by running the `show ip pim sparse-mode mroute` command on ZebOS-XP:

```
# show ip pim sparse-mode mroute
IP Multicast Routing Table

(*,*,RP) Entries: 0
(*,G) Entries: 0
(S,G) Entries: 1
(S,G,rpt) Entries: 1

(192.168.1.1, 224.0.1.3)
RPF nbr: 0.0.0.0
RPF idx: None
SPT bit: 1
Upstream State: JOINED
 Local     .l.............................
 Joined    ..j............................
 Asserted  ...............................
 Outgoing  .oo............................

(192.168.1.1, 224.0.1.3, rpt)
RP: 0.0.0.0
RPF nbr: 192.168.2.35
RPF idx: eth2
Upstream State: RPT NOT JOINED
 Pruned    ...............................
 Outgoing  ...............................
```

3. In this output, the SPT bit for the (S,G) state is set to 1, indicating that the source tree is formed on R3 and is already delivering multicast traffic over the source tree. However, Register VIF is still on outgoing list. This indicates that ZebOS-XP is not receiving Register-Stop from RP. This is confirmed by observing packets coming in on R3 `eth2`. It also explains what we observed in `step 1`, both native multicast traffic and encapsulated Register packets come out from `eth2`.

4. Using your packet sniffer, observe the packet on R3 `eth2` further. All multicast packets coming out from `eth2` have the same source IP address `192.168.2.2`, except the Register packet, which has an IP address `192.168.1.2`.

5. Run the `show ip route` command on R2 (RP). There is no route to reach `192.168.1.2`. Because RP does not have a route to source, it failed to send the Register-stop message.

6. According to the protocol specification, the Source address of the Register packet is the DR address of the source, which is the IP address of the interface toward the source (in this case, 192.168.1.2).

   Typically, the firewall does not NAT locally generated packets. When the Register packet is sent out from `eth2` on `R3`, the Source address (in this case, `eth1`) is not NATed by the fireWall.

   A solution for this is to change the IP source address of the Register packet. Use the `ip pim register-source` command to configure the source address of Register packet:

   ```
   # configure terminal
   (config)# ip pim register-source 192.168.2.2
   ```

   After changing the source address, ZebOS-XP sends Register packets with source address as 192.168.2.2 and receives Register-Stop packet from the RP. ZebOS-XP stops encapsulating multicast data packet in the Register packet. The receiver now receives only one copy for each multicast data packet.

Note: When running ZebOS-XP PIM with checkpoint NAT, the Register source address must be a reachable address (visible to the external network) to be used by the RP to send corresponding Register-Stop messages in response. You might use the `ip pim register-source` command to change the Register packet source address.

CHAPTER 8 Miscellaneous Issues

This chapter contains steps for resolving miscellaneous issues.

Kernel does not notify NSM about updating MTU

Live MTU updates are sent by NSM to the protocols on Linux and all flavors of BSD. For Solaris, the kernel does not notify NSM about updating the MTU/metric. To trigger this information from NSM to protocols, you need to administratively bring the interface down, modify the MTU/metric and then bring the interface up. This sends the new MTU/metric update to the protocols.

OSPF adjacency lost (System Clock)

When changing the system clock (moving it backward or forward) the OSPF daemon on Solaris loses adjacency. OSPF adjacency is lost and stuck in “Init” state. This is a known Solaris issue. When changing the system time using any mechanism, you need to shut down the system and bring it up. So when changing system time on Solaris, shut down the system and restart ZebOS-XP protocols.

Remote Devices are Unreachable

If you cannot reach remote devices when restarting NSM, make sure that there is at least one static IP address configured from the primary interface of the operating system. Failure to do so can result in the device becoming unreachable from the outside. The connectivity established through ZebOS-XP is lost when ZebOS-XP is killed and the device requires manual intervention. To configure a static address from the primary interface, use the `ifconfig` command or edit a file in the network-scripts:

1. Edit file `ifcfg-eth<x> in /etc/sysconfig/network-scripts` with this minimum configuration:

   ```
   DEVICE = eth<x>
   ONBOOT = yes
   PADDR = 10.10.10.222
   NETMASK = 255.255.255.0
   BROADCAST = 10.10.10.255
   ```

2. Make sure `/etc/rc.d/init.d/network` does exist to configure a network interface with a static IP address at boot time.

CHAPTER 9 Frequently-Asked Questions

The chapter contain answers to frequently-asked questions (FAQ) about ZebOS-XP. This chapter is organized in these sections:

General

Q: How do I configure a loopback address?

Use the following command to configure the loopback interface lo with address 127.0.0.1:

```
ifconfig (loopback-address) 127.0.0.1
```

where (loopback-address) is “lo” for Linux and FreeBSD and “lo0” for Solaris.

Q: Is there a simple method to count the number of routes in the routing table?

To count the number of routes in the routing table, use the `show ip route` command, and direct the output to a text file using the “>” token.

```
#show ip route > /home/ipi/route.txt
```

Open the text file in a text editor and turn on the line counter to see the total number of routes.

Q: How can I increase the size of the routing table in the Linux kernel?

By default, Linux allows installing a maximum number of 4096 routes (IPv4 or IPv6) in the kernel. To change this limit, edit the `max-size` file:

IPv4: `/proc/sys/net/ipv4/route/max-size`

IPv6: `/proc/sys/net/ipv6/route/max-size`

The maximum number of routes allowed in the kernel is 2147482645.

Q: Why does the DHCP server fail to start when an interface related to the server comes up?

The basic requirement to start the `dhcpd` daemon is:

- A pool must be defined in `dhcpd.conf`
- An interface must be associated with that network

So, when the interface is in “shutdown” although ZebOS-XP tries to invoke the `dhcpd` daemon, it fails because there is no interface associated with it. So when you give the `dhcpd stop` command and then the `dhcpd start` command, both the above conditions are satisfied and the daemon runs.

Now, when the `dhcpd` daemon runs, it won't check whether the interface associated with it is up or down. That is just the initial requirement. This is inherent to DHCP.

Q: Is there any particular order in which the ZebOS-XP daemons must be started?

It is strongly recommended that you start NSM first, followed by the protocol daemons, and then followed by IMI. However. there is no hard and fast rule regarding the sequence.

Q: What is the correct order in which I should start NSM and the protocols? Is there a reconnect procedure between NSM and the protocols?

There is no order to start NSM and the protocols. A protocol can be run independent of NSM. When NSM is killed, the protocols can still run. There is no reconnect procedure between NSM and the protocols.

Q: Assuming a ZebOS-XP process fails, and we detect the failure, can we just restart that process? Or, will other processes (for example, HSL, NSM, IMI and other protocol modules) be affected by traffic impact if the restart is not performed in a specific order?

Protocol modules can start and stop dynamically. The advantage of using each module as a separate process is minimum impact on the device.

Q: How is memory initiation and clean up in protocol modules handled in ZebOS-XP?

The `memmgr_init_memtable` and `memmgr_free_memtable` functions are called by each of the protocol modules for initiation and clean up.

In a flat memory model, such as, VxWorks - all protocol modules share the same memory space. Allocations and freeing happens local to that particular protocol module, and would not have an impact on other modules.

Q: What blocks of our code are GPL restricted?

ZebOS-XP is pure commercial software: none of the software modules we sell are GPL-bound.

To evaluate our Layer-2 software modules, IP Infusion Inc. has developed the Linux kernel forwarder to allow some of our Layer-2 protocols to run on a Linux x86 PC environment. The ZebOS-XP Layer-2 forwarder for Linux has been adapted and enhanced for the Linux Ethernet bridge, which is GPL software. We provide it free of charge, and primarily for evaluation purposes.

To evaluate our MPLS software modules, IP Infusion Inc. has developed the Linux kernel forwarder to verify our MPLS modules for Layer-3 VPNs. The ZebOS-XP MPLS forwarder requires a patch to the Linux kernel, which is GPL software. We provide this free of charge.

Q: From the OS point of view, is each daemon is single threaded?

Yes, OS treats each daemon as a single thread.

Q: Do you have a detailed memory-consumption description that includes code size, variable size and dynamic memory allocation, as a function of routes, peers, and so on?

All calls to memory management within in ZebOS-XP are associated with a type. ZebOS-XP has a memory management implementation which maintains the memory statistics per type, for example: count and size of each type. ZebOS-XP provides measurements for customers to engineer flash and DRAM requirements.

Q: Are there any hard limitations on the number of interfaces, peers, route entries, and so on?

No, ZebOS-XP has no hard limitations on the number of interfaces, peers, routes, and so on

Q: Do you have In-Service Software Upgrade (ISSU) support?

Because ZebOS-XP’s architecture is modular, it is feasible to support ISSU. Although ZebOS-XP does not have the system upgrade process, ZebOS-XP can be integrated with an external ISSU module, which can upgrade ZebOS-XP on a module-by-module basis.

Q: Which parts of the software packages can be distributed across different CPUs?

All protocol modules can be distributed across different CPUs.

Q: What are the debugging and logging tools used for remote debug and assistance?

ZebOS-XP supports syslog which can be used for remote debug and assistance. At the protocol level, ZebOS-XP has the ability to perform in depth-debugging and logging of protocol timers, packets, finite state machines, calculations, and so on.

Q: How do I enable the remote syslogging and kernel debugging options?

Perform the following:

1. Update `/etc/syslog.conf` in the machine generating logs:
   ```
   *.* @hostname --> Syslog server name/IP address
   ```
2. In the machine where `syslog` runs, add the “-r” option in the `/etc/sysconfig/syslog` file:
   ```
   SYSLOGD_OPTIONS="-m 0 -r"
   ```
3. Restart `syslogd`, and make sure that it is listening on port 514:
   ```
   #service syslogd restart
   #netstat -an | grep 514
   udp 0 0 0.0.0.0:514 0.0.0.0:*
   ```
4. To log kernel messages, update the `/etc/syslog.conf` file as:
   ```
   kern.* /var/log/messages
   ```

Q: How do I trigger/simulate alarms/traps, and so on in the code?

Use signals:

1. Register the signal in the code with the signal handler.
2. Do the processing in the signal handler. For example, in `nsm_main.c`, register the `SIGUSR1` signal:
   ```
   pal_signal_set(SIGUSR1, siguser_handler);
   ...
   int siguser_handler(int signo)
   {/* Do the processing */}
   ```
3. From the command line, invoke:
   ```
   kill -SIGUSR1 <pid of NSM>
   ```

Q: How can I handle crashes and generate a core file?

Perform the following:

1. Give the `ulimit -a` command on the box, and check whether or not the size is set for core (0 or unlimited).
2. If set for 0, set to unlimited using the `ulimit -c unlimited` command.
3. Copy the above command to the `/root/.bashrc` file to keep the settings for all sessions. The core file will be written to the root directory.
4. Use file core.5356 to get the module information. For example:
   ```
   [root@SJMCAST-1 /]# file core.5356
   core.5356: ELF 32-bit LSB core file Intel 80386, version 1 (SYSV), SVR4-style,
   SVR4-style, from 'mstpd'
   core file is for mstp
   ```
5. Run GDB for the daemon with the core file to get the back trace. For example:
   ```
   [root@SJMCAST-1 /]# gdb /usr/local/sbin/mstpd ./core.5356
   GNU gdb Red Hat Linux (6.1post-1.20040607.62rh) Copyright 2004 Free Software
   Foundation, Inc.
   GDB is free software, covered by the GNU General Public License, and you are
   welcome to change it and/or distribute copies of it under certain conditions.
   Type "show copying" to see the conditions.
   There is absolutely no warranty for GDB. Type "show warranty" for details.
   This GDB was configured as "i386-redhat-linux-gnu"...Using host libthread_db
   library "/lib/tls/libthread_db.so.1".
   ```

```
warning: exec file is newer than core file.
Reading symbols from shared object read from target memory...done.
Loaded system supplied DSO at 0xffffe000 Core was generated by `./mstpd -d'.
Program terminated with signal 11, Segmentation fault.
#0 0x0804da63 in mstp_msti_rr_timer_handler (t=0xbf8f2270) at
mstp_timer.c:444
444 mstp_timer.c: No such file or directory.
in mstp_timer.c

(gdb) bt
#0 0x0804da63 in mstp_msti_rr_timer_handler (t=0xbf8f2270) at
mstp_timer.c:444
#1 0x0804a145 in mstp_start (deamon_mode=1, config_file=0x0, vty_port=2618,
progname=0xbf8f3c30 "mstpd") at mstp_main.c:80
#2 0x08049fa4 in main (argc=5356, argv=0xbf8f2374)
at ../../platform/linux/mstp.c:105
(gdb) q
```

Q: Do I need to stop a process to start in gdb?

No. Use `attach <pid>` in `gdb`:

```
[root@Albatross_IDC:/root]# ps -ef | grep " -d"
root 6708 1 0 Jan01 ? 00:00:02 ./HSL -d
root 6710 1 0 Jan01 ? 00:00:00 ./nsm -d <<<<<<<<
[root@Albatross_IDC:/root]# ../Sai/gdb
GNU gdb 6.4
Copyright 2005 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "powerpc-wrs-linux-gnu".
(gdb) attach 6710 <<<<<<<<
Attaching to process 6710 <<<<<<<<
```

Q: Does ZebOS-XP mainly use socket-based interprocess communication (IPC) to communicate between NSM and the protocol modules and the data plane?

Yes, ZebOS-XP uses a socket-based (UNIX domain for Linux) IPC mechanism to communicate between NSM and the protocol modules and the data plane.

Q: What is the Export Control Classification Number (ECCN) for ZebOS-XP software?

ZebOS-XP is classified under ECCN 5D991.

Q: Please provide more information about the reasons for having separate daemons for IPv4 and IPv6 for RIP and OSPF, whereas for BGP and IS-IS there is one daemon each.

There are separate daemons for IPv4 OSPF and RIP because they are defined in separate specifications. BGPv4 and v6 are defined in one specification, so ZebOS-XP has a single daemon for BGP and ISIS.

Q: How is synchronization between the CLI, SNMP, and the configuration API done?

ZebOS-XP is a single-threaded modular architecture. This architecture simultaneously resolves the synchronization issues when using the CLI, SNMP, and configuration API features, because only one of these features is serviced at a time.

Q: I am using MD5 authentication and notice a failure to reach full adjacency across different platforms.

Sometimes a different version of the libcrypto library might cause this problem. Check the libcrypto library version. ZebOS-XP uses "libcrypto.so.0.9.7" in its testing environment.

Q: Is ZebOS-XP compliant with the uClibc environment (http://www.uclibc.org/FAQ.html)?

ZebOS-XP Layer-3 is compliant with uClibc, but the Layer-2 software forwarder is not compliant with uClibc.

Q: What additional value do we get from EmbeddedMind (http://www.oracle.com/us/industries/communications/communications-embedded-mind-1375715.html) integration with your code, compared to your package?

EmbeddedMind integration provides a unified management layer. Most importantly, it supports an XML and Web interface, apart from the CLI and SNMP. ZebOS-XP supports only CLI and SNMP.

Q: Where are QoS and ACL management located within your architecture?

QoS is maintained as part of the Network Services Module (NSM). ZebOS-XP does not support data-plane ACLs.

Q: Why is the prefix of an IPv6 link-local address /64 instead of /10.

The prefix of fe80::/10 identifies the link-local scope of an IPv6 address where the higher-order 10 bits indicate the scope. The actual prefix length is usually configured to be /64.

RFC 3515, section 2.5.6, states that the 54 bits following 1111111010 should be set to 0 in the link-local address. This implies an allowable prefix of /64 based on a 64-bit interface identifier that occupies the lower-order bits. Therefore, it is preferable to retain the prefix length as 64.

Q: Is a link-local address required for protocols such as RIPng and OSPFv3?

Yes, a link-local address is required for IPv6 protocols such as, RIPng or OSPv3 to get activated. ZebOS-XP provides options to configure its own link-local address. Also, a user can visualize this using the `show running-config` command.

Q: How is IPv6 support enabled on RH9?

Perform the following:

1. Open the `/etc/sysconfig/network`.
2. Add the following two lines:
   ```
   NETWORKING_IPV6=yes
   IPV6INIT=yes
   ```
3. Restart the network service:
   ```
   #service network restart
   ```

It will show an additional interface (sit0) in the interface list (`ifconfig -a`) for ipv6-in-ipv4.

Q: How is an interface with an IPv6 address configured on RH9?

```
ifconfig <IFNAME>
inet6 add <IPV6 ADDR>
```

For example:

```
ifconfig eth1
inet6 add fec0::1111:2222:3333:4444/64
```


Q: Can “prefix-list” be used with the “distribute-list” command?

No. ZebOS-XP supports only standard or extended access-lists in the `distribute-list` command and does not support “prefix-list”.

Q: Which modules support passive-interface?

Passive interface is supported in RIP, OSPF, and PIM.

Q: How many designated routers (DRs) can be present in an area?

A DR and/or back-up designated router (BDR) are elected per LAN segment. There can be zero or more DRs or BDRs in the segment depending on the type of connectivity.

IMI/IMISH

Q: I set the hostname using the ZebOS-XP CLI, and it changed correctly, but when I set the hostname in the kernel, it did not change.

Changing the hostname using the ZebOS-XP CLI takes precedence over changing the hostname in the kernel. If you change the hostname in the CLI and then change the hostname in the kernel, the hostname changed in the CLI will remain. For details on the `hostname` command, refer to the *NSM Command Reference*.

Q: Where is the configuration file for protocols?

Configuration files are stored in `/usr/local/etc`. See the *Integrated Management Interface Developer Guide* for more.

Q: How are the configuration files organized? What is the ZebOS.conf file used for and what is stored at the per protocol (for example, ospf.conf)?

For configuration storage, ZebOS-XP provides two mechanisms: one using Integrated Management Interface (IMI), and the other using individual protocol modules. Using IMI, the configuration management is unified, and configuration for all protocol modules is maintained in the `ZebOS.conf` configuration file. In non-IMI mode, the configuration is maintained by individual protocol modules.

Q: How is configuration backward compatibility handled?

ZebOS-XP maintains the configuration compatibility by maintaining the old CLI configuration. Because most configuration remains unchanged for a specific protocol module, we can guarantee back-compatibility.

Q: What does the “copy running-config startup-config” command do?

The `copy running-config startup-config` command copies the current running configuration to the startup configuration file.

Use this command if you want to write the current configuration to the file that is read at startup. This command is the same as the `write memory` command and is available in the Privileged Exec mode.

Example:

```
>enable
#copy running-config startup-config
```

Q: How do I enable logging in ZebOS-XP?

See Chapter 1, *Debugging and Logging*.

Q: What functions are called if the “write” command is entered?

Every module registers a callback function with the CLI using `cli_install_config`. This call back function is called to save the running configuration.

Q: What is the maximum length for the host name?

The host name can be set to any length but when displayed it is restricted to the `MAXHOSTNAMELEN` ZebOS-XP constant.

Q: How do we display data using the “show” command? And in what mode should the “show” commands be registered?

For all `show` commands, you need to use the `cli_out` function to display the required data. All `show` commands, except `show running-config` need to be installed in EXEC_MODE. This is because `show` commands are handled in a different way from configuration commands. When a user gives a show command, this creates a show connection between IMISH and the protocol module to fetch and display the data. This, in turn, calls the respective `cli_out` function to display the required data. This is mentioned in *Integrated Management Interface Developer Guide*.

Q: What function should be used to dump “debug” command related data?

To dump debug related data, the various `zlog_*` functions available in ZebOS-XP can be used. They will automatically take care of sending the message to the terminal. For example, look into the code to understand how `zlog_info` can be turned on or off for different debug settings.

Q: What is the “privilege” command for?

The `privilege` command is used to set a new command privilege level. Refer to the ZebOS-XP IMI Command Reference for the syntax and command modes to set and show the privilege level.

Understanding Privilege Levels

By default, the ZebOS-XP CLI has two levels of access to commands:

- User EXEC mode (level 1)
- Privileged EXEC mode (level 15).

However, you can configure additional levels of access to commands called privilege levels to protect your system from unauthorized access. Up to 16 privilege levels can be configured, from level 0, which is the most restricted level, to level 15, which is the least restricted level.

For example, if you want a certain set of users to be able to configure only certain interfaces, but not allow them access to other configuration options, you could create a separate privilege level for only specific interface configuration commands, and distribute the password for that level to those users.

Q: What is the use of the cli_install_gen and cli_install_imi functions?

If the command is to be executed only by the IMISH module (as `show history`), it must be installed using the `cli_install_gen` function.

If the command is to be executed only by a protocol module (as `aggregate-address A.B.C.D/M`), it must be installed only inside the protocol module by using the `cli_install_gen` function.

If the command is to be executed by the IMISH module and by IMI (as `write`), it must be installed:

- In IMISH by using the `cli_install_imi` function
- In IMI by using the `cli_install_gen` function

All IMI related commands common to many protocols should be installed using `cli_install_imi` with the suitable module mask. To be specific, all the commands related to DHCP, NAT, NTP, access list, prefix lists and route map are installed using the `cli_install_imi` function.

If the command is to be executed by IMISH, IMI, and by a protocol module PM_Y (as `hostname WORD`), it must be installed:

- In the protocol module by using the `cli_install_gen` function
- In IMISH by using `cli_install_imi` in this way:
  ```
  cli_install_imi (ctree, X_MODE, PM_Y, PRIVILEGE_NORMAL, 0, &imish_func);
  ```
- In IMI by using `cli_install_imi` in this way:
  ```
  cli_install_imi (ctree, X_MODE, PM_Y, PRIVILEGE_NORMAL, 0, &imi_func);
  ```

Also, all commands, except the mode change commands in ZebOS-XP are installed in IMI and IMISH using the `cli.pl` script to build the CLI tree during compilation. For all such commands, IMISH and IMI simply pass on the command to the respective the protocol module. You can confirm this from the generated files `imi_cmd.c` and `imish_cmd.c`.

Q: How does the “show running-config” (or the “write”) commands work? Who decides the order of commands on the configuration file?

The interface mode is a subset of the config mode, that is, to access interface mode, you need to be in the config mode.

The command tree is organized and indexed based on the modes in which the commands are installed. Hence, a command installed in config mode is always placed before a command installed in interface mode.

The `show running-config` and `write` commands use the same order to display and save the configuration. To understand the implementation, refer to the `imi_config_write_full` function in `imi_config.c`. This function is called from both the `show running-config` and `write` commands.

The `imi_config_write_full` function, in turn, calls the `imi_config_write_config` function, which decides the order in which the configuration is displayed or saved

Q: How do we get the CLI in imish and imi to do different tasks in a particular order?

The flag CLI_FLAG_LOCAL_FIRST is used when you want the command in `imish` and `imi` to do different tasks in a particular order. If this flag is not set, the command will always be executed in IMI.

Take the example of the `reload` command defined in imish_cli.c. Here, the CLI in `imish` waits for the user input, “yes/no”, before rebooting the system. Note that the actual system reboot is done only in `imi`.

Q: The “do show run” command, though executed at the Interface mode (config-if), returns to config mode, upon execution

The `do show run` command does not exist at the interface level. It is a config level command. Even though we provide the functionality to execute this command from the interface level, on exiting the CLI, it will return to the config level to which it belongs.

Q: What type of access lists can be used for NAT?

The ZebOS-XP type access-list supports NAT.

Q: How do I enable an access-list when I create an access-list and an IP NAT pool with the necessary commands?

The `ip access-group access-list-name in/out` command is necessary to enable the access-list on that interface:

- “in” works on packets coming into the interface.
- “out” works on packets being sent from the interface.

Q: Does IMISH prompt the user name and password?

IMISH only invokes the shell, and does not prompt for a user name. With IMI builds, there is no way to specify a user name. Password protection is only available for the `enable` command.

Q: With the cli_out function, how many characters is it possible to display?

The `cli_out` function internally calls the `fprint` function, which writes the formatted string to a specified output stream (such as `stdout` or file). Thus, the limit for `fprintf` is same as the limit of `cli_out`.

Q: Can the IMISH inactivity timeout be changed?

Yes, by using the `exec-timeout <0-35791>` command, where the value is in minutes.

Q: “no” commands do not work in additional “configure terminal” sessions.

It is related to the `--enable-multi-conf-ses` build option, which allows multiple configuration sessions to be simultaneously active (2 or more users configuring the system via different IMISH sessions). The “no ... commands are disabled” message is displayed when at least 2 sessions are configuring the same daemon, one user has opened a nested configure mode, and another user is trying to execute the “no ...“ command. This prevents deletion of the context for the other command.

Q: Does ZebOS-XP provide a password encryption service?

Yes, CLI encrypts passwords for:

- Enable passwords
- Passwords with a user name. For example, if the command `username test password test123` is given, the “test123” string is encrypted.

However, the passwords are encrypted only after the configuration is saved using the `write` command.

Q: Which encryption standard does ZebOS-XP use when the password encryption service is activated?

ZebOS-XP uses the Data Encryption Standard (DES) using the Linux crypt function.

Q: How can configured passwords be removed, and when is the password removal effective?

Remove the password line in the corresponding `.conf` file and restart the daemons to access the router to make the password removal effective.

Q: When using the “terminal monitor” command to display the debug logs, the debug logs are not displayed in the output. Does this command require any specific daemons to be started?

To make the `terminal monitor` command effective, the Linux `syslogd` utility must be running with the configuration file `/etc/syslog.conf`.

Q: If there are more than 1000 routes, and the show ip route command is entered, why does the CLI display only the first 20 entries, stop, then wait for the user to press enter, space, or q to quit?

To avoid this, enter the command `term length 0` in Exec mode; then enter the `show ip route` command.

NSM

Q: What is the default status of the if-arbiter? When do I need to enable/disable the if-arbiter?

In ZebOS-XP, the if-arbiter is disabled by default. When interface-related operations are performed outside of ZebOS-XP (for example, when using OS ifconfig), enable if-arbiter for a transient time to complete synchronization. When synchronization is complete, disable it using the if-arbiter CLI. Refer to the *NSM Command Reference* for details on this command.

Q: How does NSM notify the protocols about MTU/metric updates? In Solaris and VxWorks, how do I get the kernel to notify NSM about MTU/metric updates?

Live MTU/metric updates are sent by NSM to the protocols on Linux and all flavors of BSDs (FreeBSD, NetBSD, OpenBSD, LynxOS, VxWorks-IPNET and OSE-IPNET).

For Solaris and VxWorks, the kernel does not notify NSM about the MTU/metric updates. To trigger this information from NSM to protocols, you need to administratively bring the interface down, modify the MTU/metric, and then bring the interface up. This will send the new MTU/metric update to the protocols.

Q: How can I limit bandwidth? I used the bandwidth command, but it did not limit the bandwidth.

ZebOS-XP uses bandwidth limitation only when TE protocols such as RSVP request bandwidth from NSM for setting up LSPs.

The `bandwidth` command does not affect the physical bandwidth of a system. It defines the logical bandwidth available to the protocols. It simply allows users to define the total bandwidth of an interface for operating systems in which the kernel does not send bandwidth information to NSM when interfaces come up.

Thus, QoS is configured through the kernel, not through NSM.

Note: Refer to the *NSM Command Reference* for details on the `bandwidth` command.

Q: I have ripd and ospfd running: what will happen when I restart NSM?

Restarting NSM does not affect `ripd` or `ospfd`. Even when NSM is killed, it does not affect the functionality of other protocol daemons.

Q: Why does setting the MTU size under ZebOS conf not affect the kernel MTU size?

To change the MTU size in ZebOS-XP, use the `ifconfig` command. Use the `ip ospf mtu` command to control the MTU size of OSPF packets.

Q: I have configured an IP address on an interface using “ip address A.B.C.D/M”. Yet, I am unable to ping the interface.

The command `show interface IFNAME` will confirm if the interface is up and if it is a VLAN, whether the VLAN is active. Also Layer 3 support is necessary. This is as per design; we cannot ping an interface in pure Layer 2 support.

Q: The IPC between NSM and other daemons takes place through UNIX domain sockets. Is it possible to change this communication mechanism to use TCP (AF_INET) sockets? Is it possible for NSM and IMI to reside on one machine, and the protocols on another machine?

Yes, TCP/IP sockets can be used for IPC between NSM and the protocol modules. Add `--enable-tcp-message` in the `config.sh` file, and follow the steps in the install manual to compile the code. Technically, it is possible for NSM and IMI to reside on a different machine from the protocols, but the code may have to be modified to ensure that NSM is aware of all interfaces participating in routing across all machines.

Q: Is there a command that can register the MAC table and the ARP table in Static?

1. Create a static ARP entry:
   ```
   (config)#arp 1.1.1.10 aaaa.bbbb.cccc
   ```
2. Register the MAC Table:
   ```
   (config)#bridge 1 address aaaa.bbbb.cccc forward eth1
   ```

Q: How do you enable Neighbor Discovery?

The Route Advertisement in neighbor discovery is suppressed by default. To enable it, use the `no ipv6 nd suppress-ra` command.

Q: Does NSM recover the routing information from the underlying platform during graceful restart?

Yes, NSM recovers the routing information from the underlying platform/kernel while restarting. During NSM restart process, NSM keeps the kernel FIB intact, and all protocol modules repopulate their routing information after NSM starts.

Note: To retain NSM routers in the FIB, use the `fib retain` command. Otherwise, NSM cleans up the FIB when it exits.

See the *Architecture Guide* for details.

Q: I am not able to set the MTU to 2000 on a fast Ethernet interface. What is the problem?

The accepted MTU values are from 65 to 1500 for a fast Ethernet interface.

Q: What is the difference between the ip-access-list and access-list commands?

The `access-list` command configures an access list for filtering packets. This filtering list will be used by all protocol modules.

The `ip-access-list` command creates an IP access-control list (ACL) (based on the source address) or creates an IP extended ACL (based on the source and destination address.) This command is related to QoS and until QoS is enabled using `mls qos enable`, an IP access list cannot be created.

Q: How do you merge routes from different sources into the main RIB? Is it possible to configure the weights of routing information from the various sources?

In ZebOS-XP, the Routing Information Base (RIB) is part of the Network Services Module (NSM). Each protocol module populates the routes to NSM, which runs the route selection process, based on administrative weights attached to each protocol module. The administrative weights assigned to each protocol type are based on industry-standard implementations, but can be changed through the source code.

Q: Is it possible to create ACLs that also inspect Layer-4 parameters?

No.

Q: From NSM's point of view, is there any constraint with regard to the time between stopping the daemon and restarting it?

This depends on the graceful restart period configured before the restart. The helper router retains the routes and adjacency, until the grace time expires. If the other router restarts before the graceful restart time expires, then the helper router maintains the state; otherwise, it flushes the neighbor adjacency and routes learned from the neighbor.

Q: What are summary addresses and what is the use of a NULL0 interface?

Summary addresses are aggregation of routes.

The installation of a NULL interface occurs as soon as a summary address is created. For example, if the `show ip route` command is issued, the following is displayed:

```
i       4.4.4.0/22 [115/0] is a summary, Null, 00:04:54
```

Any packets destined for summarized routes have a longer match in the routing table. Packets that do not match one of the summarized routes, but match the summary route, are dropped. For example, if the `show ip route` command is issued, the following is displayed:

```
i       4.4.4.0/22 [115/0] is a summary, Null, 00:09:45
C       4.4.4.0/24 is directly connected, eth1
```

The example above shows that 4.4.4.0/24 is directly connected and has a route in the routing table.

Q: If a summary address matches multiple routes, which metric is chosen?

The metric chosen for the advertised summary is the smallest metric of the matching routes.

Q: If a summary address matches both internal and external routes, which route is preferred?

The internal route is preferred over external routes, even though the external metric is less than the internal metric.

Q: Is there a command to configure the ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) prefix for the ISATAP subnet?

When the IPv6 address is configured on the ISATAP router, the IPv4 address is embedded into the IPv6 address: this provides the ISATAP address.

Q: Does ZebOS-XP support LDP-ISIS synchronization?

Yes, ZebOS-XP supports LDP-ISIS synchronization. LDP-ISIS synchronization occurs by default because in ZebOS-XP all daemons send learned routes to NSM, and NSM handles routing between systems. In LDP-ISIS synchronization, ISIS sends routes to NSM, and LDP assigns labels to the routes found in the NSM database. The functions that correspond to these operations are:

- `nsm_server.c`: `nsm_server_send_route_ipv4`
- `ldp_nsm.c`: `ldp_nsm_recv_route_ipv4`

Q: What happens if LDP is not ready after ISIS has converged? How does ZebOS-XP route the packets? Is it possible to control LDP so it is synchronized before the IGP converges?

- If LDP comes up *after* ISIS has converged, it will get all current routes from NSM, exchange FEC, and assign labels for the route (standard LDP behavior). If there is no FEC, the packet will be forwarded as a normal IP packet.
- If LDP comes up *before* ISIS has converged, it registers with NSM, and once NSM gets the routes from ISIS, LDP starts assigning labels to those routes found in the NSM routing table.

Q: What is the meaning of recursive next-hop flag in the RIB?

Consider the following case:

10.1.1.0/24

(OSPF) Box 1 eth1 (.1) ---------- (.2) eth1 Box 2 (OSPF)

199.1.1.0/24 is learned through 10.1.1.1 on Box 1. If you configure a static route on Box 1 such as 188.1.1.0/24 through 199.1.1.0, this will show as a recursive route, because 188.1.1.0 is resolved through 199.1.1.0,which is resolved through 10.1.1.0.

Q: What is non-stop-forwarding?

Non-stop forwarding (also called graceful restart) is available for OSPF, ISIS, RIP, NSM, LDP, RSVP and BGP. On a high level, for example: in ISIS, if R1 is configured as the restarting router, until the restart-hold time is expired, NSM maintains the routes learned through ISIS. If ISIS comes back up before the hold time, the routing continues normally. If ISIS does not come back up before the configured hold time, NSM removes all of the routes learned through ISIS.

Q: What is the difference between router-id commands in configure mode and in router mode?

If there is no router-id defined for a protocol, the highest IP address in the router is used as the router ID. This can be changed by either specifying an NSM-level router ID or a protocol-level router ID. Note that when an NSM-level router ID is specified, it supersedes any other configured protocol router ID.

Q: Is the following procedure possible? 1) Start ZebOS-XP with configuration A. 2) Call the stop_zebos function. 3) Replace ZebOS.cong via some means (resulting in configuration B). 4) Call the start_zebOS function. 5) ZebOS-XP is now running with configuration B.

Yes. After the ZebOS.conf is replaced, and a call is issued to `stop_zebos` then to `start_zebos`, the new configuration is seen.

Q: The interface index (ifindex) value is not the same when interface changes are made between switchport and router port from IMISH.

The `ifindex` for a Layer 3 interface is generated by the TCP/IP stack in the kernel, while an `ifindex` for a Layer 2 interface is generated by ZebOS-XP. The reason for this is that the TCP/IP stack only maintains information about Layer 3 interfaces, but does not maintain information about Layer 2 interfaces. ZebOS-XP assigns the `ifindex` for all Layer 2 interfaces beginning with 5001.

The kernel assigns a new `ifindex` every time an interface is changed from Layer 2 to Layer 3. For example, if GE1 is configured as a Layer 3 interface with an `ifindex` of 3, each time the `switchport` or `no switchport` command is given, GE1 is assigned a new `ifindex` by the kernel.

Since there is no control over the `ifindex` assignment for Layer 3 interfaces, it is not possible to maintain the same `ifindex` for both Layer 2 and Layer 3.

Q: How do I enable Equal-Cost Multipath (ECMP)?

The Equal-Cost Multipath (ECMP) enables a router to have several next-hops available for any given destination.

For example, if you configure the following static routes in ZebOS-XP:

```
ip route 1.1.1.1/32 10.0.0.2
ip route 1.1.1.1/32 10.0.0.3
```

Run the `ip route` command in the kernel. The static routes will be displayed in the kernel routing table as:

```
1.1.1.1 proto zebra
nexthop via 10.0.0.2 dev eth0 weight 1
nexthop via 10.0.0.3 dev eth0 weight 1
```

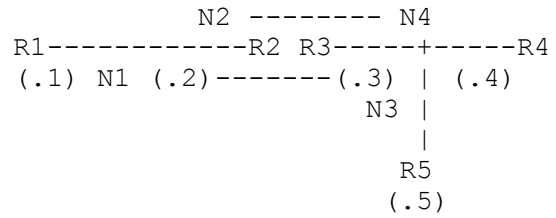
How do I verify load balancing on a Linux kernel?

The ZebOS-XP implementation leverages the Forwarding Plane Load Balancing when the underlying kernel supports ECMP (Equal Cost Multipath).

ZebOS-XP installs the maximum number of ECMP routes supported by the kernel. With this, load balancing can be performed with more than one nexthop to reach a destination. In case the router receives and installs multiple paths with the same administrative distance and cost to a destination, load balancing is possible.

Ideally, multiple nexthops have different interfaces to the destination, but this is not mandatory. The algorithm for distributing traffic across ECMP routes is dependant on the kernel, and typically based on the protocol, source address, destination address and the port.

To verify if load balancing is working on the Linux kernel, use the `ping` utility to verify that more than more than one nexthop is being used to send packets to destinations.



In this topology, OSPF is enabled on all the routers. R2 is ECMP enabled, and Wireshark is enabled on both links of R2.

Once R1 learns about R4 and R5 through R2, ping to R4, and see ping packets on one link (for example N3) of R2. Then, ping to R5 from R1, and see ping packets on the other link (N2) of R2.

In this example, R2's kernel is load balancing, as it is using both the links to send packets.

In case only one link is used to send packets to R4 and R5, R2's kernel is not load balancing and might not be supporting ECMP.

Q: How does load balancing on NSM work for a Linux kernel?

NSM relies on the underlying kernel for load balancing. For example, suppose there are two static routes to the same destination - this can be seen from the following output:

Note: The `route -n` or `netstat -rn` commands do not show the Equal Cost Multipath Routes (ECMP) routes; the `ip route` command does shows the ECMP routes.

```
[root@localhost ~]# ip route
192.168.2.0/30 dev vlan1.3 proto kernel scope link src 192.168.2.1
192.168.1.0/30 dev vlan1.2 proto kernel scope link src 192.168.1.1
216.20.15.128/25 dev eth0 proto kernel scope link src 216.20.15.225
169.254.0.0/16 dev eth0 scope link
10.0.0.0/8 proto zebra
nexthop via 192.168.1.2 dev vlan1.2 weight 1 ------> ECMP
nexthop via 192.168.2.2 dev vlan1.3 weight 1 -------> ECMP
default via 216.20.15.254 dev eth0

[root@localhost ~]# route -n <--------Does not show the ECMP routes
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.2.0 0.0.0.0 255.255.255.252 U 0 0 0 vlan1.3
192.168.1.0 0.0.0.0 255.255.255.252 U 0 0 0 vlan1.2
216.20.15.128 0.0.0.0 255.255.255.128 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
10.0.0.0 192.168.1.2 255.0.0.0 UG 0 0 0 vlan1.2
0.0.0.0 216.20.15.254 0.0.0.0 UG 0 0 0 eth0
```

Also, the Linux stack treats the ECMP routes as flow based. That is, even if the kernel has two routes to the same destination, as long as the source IP is the same, the path taken to deliver the packets will not change. But, if you ping the same destination with a different source IP address, the other path will be used for load balancing.

Q: Does NSM acknowledge whether DCHP is enabled on a particular interface?

No, NSM does not acknowledge whether DHCP is enabled on an interface. For example, when the `ip address dhcp` command is executed through IMISH, it is directly executed in the kernel from IMISH.

Q: Is there a way to configure a VLAN such that the configured VLAN takes precedence over the default VLAN?

In interface mode, give the `switchport access vlan 2` command. This command makes VLAN 2 take precedence over the default VLAN.

Layer 2

Q: When GVRP configures a VLAN on an interface, does it automatically add the MSTP instance that to which the VLAN belongs to that same interface?

A VLAN GVRP configured on an interface does not automatically add the MSTP instance that the VLAN belongs to that same interface.

Q: What are the different switchport types with regard to the Spanning Tree Protocol?

The layer 2 CLI provides three different types of switchports:

- Switchport mode access: Access mode is used for end stations or ports connected to hosts. In access mode, untagged frames only are classified. Received frames are classified, based on the VLAN characteristics, then accepted or discarded, based on the specified filtering criteria.
- Switchport mode trunk: Trunk mode is used to connect two switches, so that it can be used to pass multiple VLANs. In trunk mode, tagged frames only are classified.
- Switchport mode hybrid: Hybrid mode can be used in either of the scenarios above. Hybrid mode is used for receiving both tagged and untagged frames. Cisco does not have an equivalent command in the Layer-2 module.

Q: How do you switch from RSTPD to STPD in IMISH?

To switch from `rstpd` to `stpd`:

1. Kill the `rstpd` daemon.
2. Start the `stpd` daemon.
3. Create a new IEEE bridge. Thus, the hello-time can be set.

Also, if you would like to send STP packets while still running the `rstpd` daemon, there is an alternate method that is interface-specific:

1. Enable switchport on an interface and associate it to the bridge group.
2. Give the `spanning-tree force version` command to enable you to forcefully set the protocol running on that interface.
3. Verify the configuration using the `show spanning-tree interface` command.

Q: What is the default bridge priority?

By default, a bridge has a priority of 32768 or 0x8000.

Q: What is the maximum number of STP instances that can be created for an MSTP?

The permitted range of instances is 0-15. Instance 0 refers to the internal spanning tree.

Q: Why is SVLAN translation supported on the customer network port?

ZebOS-XP supports both port-based service and S-tagged service. As per S-tagged service, multiple SVLANs can be configured on a single customer network port.

Q: A pseudowire (PW) type can be configured as Ethernet and VLAN: do they correspond to the types “raw-mode” and “tagged-mode”?

It depends whether the PW type is configured as Ethernet or VLAN:

- When a PW type on an interface is configured as Ethernet, only one PW can be bound to that interface: There is no VLAN tag for the packet.
- When a PW type on an interface is configured as VLAN, the VLAN ID must be configured for the PW: A VLAN tag is required for the packet.

Q: What is the relationship between pseudowire (PW) type and switchport mode (access and trunk)?

PW type “Ethernet” can be configured to switchport mode access, whereas PW type “VLAN” must be configured to switchport mode trunk.

Q: What is the maximum number of MAC addresses that can be learned on a switch?

There is no limit on the number of MAC addresses that can be learned on a switch; it is only limited by the available memory on the system.

Q: How are the MAC address flush messages sent in MSTP? Is it on per-port basis or per-instance basis?

In MSTP, the MAC address flush messages are sent on a <per-port per-instance> basis.

Q: What is the relationship between admin_xx and oper_xx in MSTP?

The main difference between admin_xx and oper_xx is: admin_xx parameters are usually set by the administrator, and oper_xx (operational status) parameters are set whenever any state changes or events occur in MSTP.

Q: What is required for interfaces to be aggregated using LACP?

Interfaces on which LACP is configured should always be in Full-Duplex mode with the same bandwidth on all of the interfaces: This ensures that the ports remain part of the aggregated interface.

Q: Does ZebOS-XP support Marker PDU generation in LACP?

ZebOS-XP does not support Marker PDU generation in LACP, however, it sends a response if a Marker PDU is received.

Q: What is the process of creating a VLAN interface through ZebOS-XP?

When a VLAN is created, ZebOS-XP sends the VLAN ID (for example, 2000) and corresponding bridge ID (for example, 1) to the kernel. The kernel creates the VLAN interface, not ZebOS-XP. After creating the VLAN interface, the kernel informs NSM about the new interface. After receiving this message, NSM updates its global interface list and informs all registered protocol modules.

Q: Does MSTP differentiate a normal interface from an aggregated interface?

No, MSTP does not differentiate a normal interface from an aggregated interface.

Q: How does MSTP send and receive Bridge Protocol Data Units (BPDUs) in an aggregated interface?

While sending out BPDUs on an aggregated interface, the Hardware System Layer (HSL) module finds the first child of the aggregator and sends the BPDU out. This ensures the BPDUs are not sent on all interfaces of the aggregator.

The only difference in the receiving functionality is that the HSL module checks whether the port on which the BPDU is received is part of an aggregated interface (that is, part of an aggregator). If so, the logical interface index of the aggregator is used, not the interface index of the actual port that received the BPDU.

Q. How many bridges can be created in the software forwarder? Can it be more than 32?

There should be no problem creating more than 32 bridges. This does not have any effect on the protocols. However, IP Infusion Inc. recommends limiting the number of bridges to less than 100 per system.

Q: Reading the documents related to OAM and with respect to ITU Y.1731, some of the requirements are not referenced. Please advise whether ZebOS-XP supports the following: 7.5 Ethernet Remote Defect Indication (ETH-RDI) 23; 7.5.1 CCM with ETH-RDI Transmission 24; 7.5.2 CCM with ETH-RDI Reception 24

Yes, RDI is supported; it is mandatory. See page 33 of ITU-T Y.1731 05/2006. RDI is carried in the flag field of a CCM and is supported in the ZebOS-XP Ethernet OAM implementation.

Q: What is the difference between the “spanning-tree portfast” and “spanning-tree edgeport” commands?

The `spanning-tree edgeport` command is used to set a port as an edge port and to enable rapid transitions. This command is an alias for `spanning-tree portfast`, so the two commands can be used interchangeably.

Q: When should the “cisco-interoperability” option be set in ZebOS-XP?

IP Infusion Inc. has tested Cisco software version 12.2 (44) SE3 (fc2) and did not find any issues. With newer Cisco software versions, the `cisco-interoperability` command does not need to be used.

Q: What is the maximum number of VLANs that can be created on a switch for non-MPLS ZebOS-XP software?

4032 of the available 4096 - the remaining are reserved VLAN IDs.

Q: How do you remove a tagged port from VLAN 1?

VLAN1 is the default VLAN. When a bridge gets created, the default VLAN gets created, and now when a port is made 'switchport trunk', it gets associated to the default VLAN. It is as per the design, and cannot be deleted. The port needs to be made 'no switchport trunk' to delete it from the default VLAN.

RIP

Q: While redistributing routes using route maps and access lists in a telnet build, the redistribution does not occur as per the route map.

Whenever a telnet build is used, make sure that all RIP related commands are entered in the RIP protocol module daemon. For example:

```
telnet <ip address> 2602
```

The access lists and route maps to use for route redistribution must be entered through the RIP daemon, not the NSM daemon, even though they are not configured through the RIP router mode. If the route maps and access lists are

configured through NSM (port 2601), and redistributing routes in RIP is done using them, all routes will be blocked by default, because the route map will not be found. Thus, access lists and route maps to be used for redistribution of routes must be configured through the RIP daemon itself.

Q: What happens to the RIP learned routes whose network address is same as one of the directly connected IPs, when the prefix length is greater than, less than, or equal to, that of the directly connected interface?

All routes with greater prefix length and less prefix length will be displayed in the RIP routing table. The equal prefix-length entry will be discarded.

For example. if the 1.1.0.0/24, 1.1.0.0/25 and 1.1.0.0/23 routes are redistributed into RIP, and a neighbor has an interface with IP 1.1.0.2/24, the RIP route entry of the 1.1.0.0/24 network will not be available. But, the 1.1.0.0/23 and 1.1.0.0/25 entries will be available in the RIP routing table. The 1.1.0.0/24 entry is available as a directly connected entry in the routing table. If this interface is down, the RIP route will become active, until the interface comes up.

Q: When redistributing other routes to RIP, why does the redistributed route always override the routes learned by RIP?

RIP learned routes have lower priority than redistributed routes (for example, connected/static/ISIS). Therefore, the redistributed routes always override the routes learned by RIP.

BGP

Q: Why is the BGP session reset after any BGP capability is configured?

In BGP, capabilities are advertised in the OPEN message during session initialization. If a capability is enabled or disabled after the session is established, the BGP session needs to be reset, and a new capability is included in the OPEN message.

The draft-ietf-idr-dynamic-cap-02.txt document proposes a mechanism to dynamically update BGP capabilities without resetting the session. ZebOS-XP supports this draft for the following capabilities:

- BGP address family
- Graceful restart
- ORF (not Route-reflector capability)

If two peers are configured to dynamically update BGP capabilities for the capabilities above, the BGP session will not need to be reset.

Q: How do I set up “neighbor send-community” in BGP?

The “neighbor send-community” is set up by default. By default, on receiving the communities attribute, the router re-announces them to the neighbor. This command does not appear in the list of available commands in the Router mode. It is visible only when the user has used the `no neighbor send-community` command. Please refer to the *BGP Command Reference* for details on how to use this command.

Use the `show ip bgp neighbor` command to confirm that the neighbor send-community is set up.

Q: What is Route Reflection used for?

Route Reflection is used in IBGP to resolve the IBGP full mesh problem. Configuring one or more routers as Route Reflectors reduces the number of connections between BGP peers within an Autonomous System (AS).

The Route Reflecting BGP peer has to be configured with the peer addresses of all its route reflection Clients. The Route Reflector is responsible for:

- Sending updates from a client peer to other client peers, as well as non-client peers.
- Sending updates from non-client peers to client peers.

Q: When I set up router BGP views, the sessions come up, but when I run the “show ip bgp summary” command, I only see the first view under the same ASN.

Since you are using BGP in a BGP route-server-client setup, you must use the `show ip bgp view summary` command, instead of the `show ip bgp summary` command to check the status of the BGP sessions. Refer to the *BGP Command Reference* to see details of this command.

Q: Can you explain more about the BGP “neighbor next-hop-self” command?

The neighbor next-hop-self command is only effective in the case of IBGP. For example:

```
                EBGP              IBGP
R3 ---------------------- R1 ---------------------- R2
```

On executing the `neighbor <a.b.c.d> next-hop-self` command on R1, the R1 router advertises the routes (if any) with the next-hop attribute that equals the R1 IP address.

This command is useful for advertising the routes learned by R1 from other routers, and are not reachable by R2.

Q: Does the “no bgp graceful restart” command turn off graceful restart? Or does it just set the parameters back to the default?

The `no bgp graceful-restart` command turns off the graceful restart functionality.

The `no bgp graceful-restart restart-time` and `no bgp graceful-restart stalepath-time` commands reset the timer values to the default values.

Q: Which BGP draft version is used to test conformance?

IP Infusion Inc. uses the IXIA ANVL test suite for testing conformance. The latest version of ANVL is based on draft-ietf-idr-bgp4-26.txt.

Q: What does “internal attribute hash information” mean in the “show ip bgp attribute-info” command?

ZebOS-XP maintains a hash table for all BGP attributes. The `show ip bgp attribute-info` command, iterates the BGP attribute hash table to display all BGP attributes (such as nexthop, community list, and as-path). Currently, ZebOS-XP displays only nexthop attribute information.

Q: What does the duration field mean in the “show ip bgp dampening” command?

The duration field displays the time elapsed since the router flap dampening (RFD) record was created, that is, since the first penalty points were evaluated.

Q: According to the description in the BGP Command Reference of the “neighbor remove-private-as” command: if the update includes both private and public AS numbers, the system treats it as an error. Does this mean ZebOS-XP drops the packet or is an error message generated?

No, ZebOS-XP does not drop the packet, it just checks whether or not the given value is within the range (1-65535).

While sending the UPDATE packet, it checks whether or not the remove-private-as flag is set on that EBGP peer. If it is set, it sends the UPDATE packet, which moves the private-AS number (for the routes received from another EBGP peer with a private-AS number in the message) to the configured EBGP peer.

Q: How does the route reflector use the cluster ID? Does it send it out in packets?

The cluster ID is used by the route reflectors to reflect the routes between different clusters. By default, a single route reflector for a cluster is identified by its router ID. When more than one route reflector is configured for a cluster, all route reflectors in the cluster must be configured with the cluster ID.

While sending updated packets, the cluster ID gets appended to the cluster list. A cluster list is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from the route reflector clients to non-clients outside of the cluster, the route reflector appends the local cluster ID to the cluster list.

Q: Does ZebOS-XP support BGP multipath?

ZebOS-XP does not support BGP multipath. However, NSM supports load balancing.

Q: In BGP, when the prefixes are limited by using the neighbor maximum-prefix command, is the route information input exceeding the maximum value deleted?

The behavior depends upon the warning-only option associated with the neighbor maximum-prefix command:

- If the warning-only option is used with this command, the software displays a warning message on reaching the limit, and will not remove the old routing information. It also accepts more prefixes from the neighbor
- If the warning-only option is not used with this command, if any extra prefixes are received, the router ends the peering. That is, the router sends a notification message, removes all prefixes received from that particular neighbor, and resets the peer session.

Q: Explain BGP auto-summary functionality.

When auto-summary is enabled, it summarizes the locally originated BGP networks to their classful boundaries. When a subnet exists in the routing table, and the following three conditions are satisfied, any subnet of that classful network in the local routing table prompts BGP to install the classful network into the BGP table:

- Classful network statement for a network in the routing table
- Classful mask on that network statement
- Auto-summary enabled

For example, if the subnet in the routing table is 75.75.75.0 mask 255.255.255.0, auto-summary is enabled, and you configure network 75.0.0.0 under the router bgp command, BGP introduces the classful network 75.0.0.0 mask 255.0.0.0 in the BGP table.

Q: What is the preferential order of attributes when some attributes, or all attributes, are applied to one neighbor in BGP?

For inbound updates, the preferential order of attributes is:

1. route-map
2. filter-list
3. prefix-list, distribute-list

For outbound updates, the preferential order of attributes is:

1. prefix-list, distribute-list
2. filter-list
3. route-map

Q: What is the difference between the AS_PATH and NEW_AS_PATH attributes?

AS_PATH is a well-known transitive attribute, whereas, NEW_AS_PATH is an optional transitive attribute. NEW_AS_PATH is used to propagate four-byte based AS path information across BGP speakers that do not support four-byte AS numbers.

Q: What is the significance of AS number 23456 in the AS_PATH attribute in BGP?

AS number 23456 is reserved. If the BGP speaker is new BGP, and its neighbor is old BGP, while sending the OPEN/UPDATE message in the packet, it sets the AS number in the AS_PATH attribute to 23456 and sets the original AS number in the NEW_AS_PATH attribute.

Q: Please clarify the behavior of the BGP neighbor command collide-established.

This command is used to detect connection collision upon receipt of an OPEN message. The local system must examine all of its connections in the OpenConfirm and OpenSent states, and then perform the conflict resolution, accordingly.

This command is enabled by default for graceful restart.

Q: How does ZebOS-XP implementation of the Extended ASN Capability feature function?

ZebOS-XP implements the Extended ASN Capability feature by allowing either 2-byte capable or 4-byte capable BGP speakers to communicate with other neighbors, regardless of whether the neighbor is a 2-byte or 4-byte BGP speaker.

Q: The BGP “neighbor password” command does not exist and MD5 authentication cannot be configured without this command.

There is no configuration option to enable BGP MD5. When the Linux kernel is patched with the MD5 patch while configuring ZebOS-XP, it finds the MD5 patch and builds BGP with the command shown below.

```
#ifdef HAVE_LINUX_TCP_MD5_H
CLI (neighbor_password,
     neighbor_password_cmd,
     NEIGHBOR_CMD2 "password WORD",  CLI_NEIGHBOR_STR
     NEIGHBOR_ADDR_STR2.
     "Set password to the neighbor",
     "The password")
```

If BGP does not find the corresponding header file (the kernel was not patched with the MD5 patch), this command is not available.

Q: When configuring BGP, the redistribute command can be used under Router mode (not address-family ipv4) and also under address-family ipv6 mode. How is the behavior affected by the modes?

When the redistribute command is configured under Router mode, only IPv4 route information redistribution is controlled; when configured under address-family IPv6 mode, only IPv6 route information redistribution is controlled.

OSPF

Q: Can I change the Area ID of an existing OSPF network configuration?

No, you cannot change the Area ID without deleting the existing configuration. You need to remove the network A.B.C.D/X area Y before changing the area ID of this network.

For example, entering “network 10.73.0.0/16 area 0.0.0.5” when “network 10.73.0.0/16 area 0.0.0.1” already exists, will display a warning message.

Q: What is the function of the “refresh timer” command?

The `refresh timer` command sets the Refresh Timer value in seconds. OSPF requires each LSA to be refreshed by the originating router every 30 minutes. The `refresh timer` command sets the time interval for refresh function call of an LSA. This causes all the LSAs to be refreshed which have almost reached OSPF_LS_REFRESH_TIMER. The default value of the refresh timer is set to 10 seconds. The range of this timer is 10 - 1800 seconds.

Please refer to the “OSPF Command Reference” for details on this command.

Q: How do I display information about max-age? I used the “show ip ospf database max-age” command: max-age was not displayed.

The `show ip ospf database max-age` command maintains a list of the all the LSAs in the database that have reached the max-age (3600 seconds).

If the LSAs have not reached the max-age, it is not displayed.

To test the functionality of max-age, follow these steps:

1. Connect two routers, R1 and R2 both of them running the OSPF daemon.
2. After a few minutes, kill the OSPF daemon on one of the routers (say on R2).
3. Wait for one hour to get a display of the list of LSAs that have reached the max-age on R1.

The following is a sample output of the `show ip ospf database max-age` command on R1:

```
# show ip ospf database max-age
OSPF Router process 100 with ID (3.3.3.4)
MaxAge Link States:
Link type: 7
Link State ID: 37.37.37.0
Advertising Router: 3.3.3.1
LSA lock count: 6
Link type: 7
Link State ID: 10.0.0.0
Advertising Router: 3.3.3.1
LSA lock count: 6
Link type: 7
Link State ID: 20.255.37.37
Advertising Router: 3.3.3.1
LSA lock count: 6
```

Q: How do I create a secondary loopback address? This address has to be advertised by LSAs to make it reachable from other routers and hosts.

Configure a secondary loopback address as follows:

```
(config)# interface lo
(config-if)# ip address A.B.C.D/32
```

For this loopback address to be advertised by LSAs, enable OSPF on this interface by configuring the routing process, and specifying the Process ID. The Process ID should be a unique positive integer identifying the routing process. Then define the interface and associate the area ID with the interface.

```
(config)# router ospf [process id]
(config-router)# network A.B.C.D/32 area 0
```

Q: What is the effect of cost on different routes: default, redistributed: and static?

To explain the effect of cost on different routes, we have used the following topology. Router R1 and R2 are connected. Interface eth1 on R2 has been assigned a cost of 25:

```
(config)# interface eth1
(config-if)# ip ospf cost 25
```

Now. three different routes are advertised from R2 to R1:

- Default route 0.0.0.0/0
- Static route 192.168.0.0/16
- Regular

Only the network of eth1 is assigned metric 25. The metric of the static route is 20. The metric of the default route is 10

All redistributed routes will not be advertised with this metric. In ZebOS-XP, redistributed routes (static, kernel, connected, as well as, routes learned from other protocols) are advertised with a metric of 20.

Note: Only BGP routes redistributed into OSPF are advertised as 1 (for compatibility with Cisco).

The metric of the default routes depends on the configuration. If “default-information-originate” is configured, the existing default routes are advertised with a metric of 10. Whereas, if “default-information originate always” is configured, the metric is advertised as 1.

The effect of cost does not affect the cost on default, redistributed, and static routes.

Please refer to the “OSPF Command Reference” for details on the commands above.

Q: Is there a limit to the size of the database in OSPF? How does the “overflow-database-external” command affect the maximum size of the OSPF database?

There is no limit to the size of the OSPF database. However, the OSPF external database alone can be limited in size by using the `overflow-database-external` command. The `overflow-database-external` command does not affect the maximum size of the OSPF database.

Refer to the *OSPF Command Reference* for details about this command.

Q: We have limited the number of ZebOS-XP routes to 5000. How does this affect the maximum OSPF database size?

Limiting the maximum routes to 5000 does not affect OSPF. OSPF learns routes from its neighbor and sends them to NSM. It is up to NSM to enter them into the kernel. After NSM has reached its maximum routes limit it drops received routes.

Q: The CPU utilization by our OSPF daemon went up to 100%.

The OSPF daemon could be stuck in an infinite loop, due to the system clock going backward. This issue has been fixed in the ZebOS-XP ARS 5.3 Release. Please request Technical Support for a patch for this issue.

Q: When I change the system clock (move it back or forward), the OSPF daemon on Solaris loses the adjacency. The OSPF adjacency is lost, and stuck in “init” state.

This is a known Solaris issue. When changing the system time using any mechanism, users need to shut down the system, then bring it up. So when changing system time on Solaris, you must shut down the system, and restart ZebOS-XP protocols.

Q: I configured 445 loopback interfaces with unique addresses. When I enabled OSPF on these loopbacks, the OSPF daemon could not send out the ls-update because of the size of the packet.

The Linux kernel does not support fragmentation of outbound RAW IP packets. Even if the packet is fragmented by the application, the kernel will only forward the first fragment, and ignore the rest.

The workaround for this is to run the `redistribute connected` command in OSPF. This will allow the connected routes to be redistributed into OSPF, thus providing reachability for these networks.

Q: Is there a way to log neighbor changes in OSPF?

You can log to a file or turn on `terminal monitor` and turn on `debug ospf nfsm`. This will log neighbor finite state machine information.

Q: How can I control the routes that are installed into NSM from the OSPF database? That is, while I cannot control which routes get into the OSPF database, I would like to be able to control which routes get installed into my routing table from the database.

The `distribute-list in` functionality achieves this requirement:

```
(config-router)#distribute-list 1 ?
  in   Filter incoming routing updates   <----------
  out  Filter outgoing routing updates
```

Q: According to RFC 2328, a router running OSPF should become the Area Border Router (ABR) if it has more than one actively attached area (though no backbone area). But, ZebOS-XP router behavior is not the same.

ZebOS-XP can accept the following ABR types:

- Cisco Alternative ABR, Cisco implementation (RFC3509)
- IBM Alternative ABR, IBM implementation (RFC3509)
- Shortcut ABR standard (RFC 2328)

Per RFC 2328, a router running OSPF should become the ABR, if it has more than 1 actively attached area, even if there is no backbone area. This is supported by the ZebOS-XP ABR type “standard” which you must explicitly configure using the command:

```
config-router> ospf abr-type standard
```

The default ABR type of ZebOS-XP router is Cisco (if not explicitly configured using the above command), as per RFC 3509, which says a “router connected to multiple areas without a backbone connection is not an ABR and should function as a router internal to every attached area”. This is because most routers are the Cisco type, or support behavior like Cisco.

Therefore, if the topology requirement is that ZebOS-XP should become the ABR if it has more than one active area, even if that does not include a backbone area, configure the OSPF ABR type as standard.

Q: In OSPF, how much time does it take for the transition of a border router as NSSA translator, if configured as an explicit translator?

It takes 40 seconds. The NSSA elected translator router should continue as the translator for 40 seconds, once the other border router is configured as the explicit translator. This delay is introduced for more stable translator transition.

Q: Among NSSA ABRs, which is elected as the NSSA translator (type 7 to type 5 translation)?

Election is done only if an NSSA ABR is not configured exclusively as a translator: in this case, the NSSA ABR router with highest router ID is elected as NSSA translator.

Q: How does OSPFv3 achieve authentication even though the header does not contain an authentication field?

When running over IPv6, OSPFv3 relies on the IP authentication header (AH) and the IP encapsulating security payload (ESP) to ensure integrity and authentication of routing exchanges.

Q: What is NSSA in OSPF?

Not So Stubby Area. NSSA is a stubby area that can have an ASBR. It uses a Type 7 LSA to advertise external routes. The ABR takes the Type 7 LSA from the ASBR, and changes it into a Type 5, to send to the rest of the OSPF domain.

Q: There are a few OSPF commands (such as “export-list” and “import-list”) with no help access. However, these commands can be successfully executed in the CLI.

These commands are registered with a hidden flag, which is why they do not have help strings. These commands are IP Infusion Inc. internal use.

Q: Is Equal-Cost Multipath (ECMP) supported for OSPFv2 and OSPFv3?

ZebOS-XP supports OSPFv2, OSPFv3, and IPv4/IPv6 static routes for ECMP.

Q: What happens when the designated router (DR) with the higher router ID goes down in an OSPF network?

When the DR goes down, the backup designated router (BDR) becomes the DR. If the original router comes up, the original DR becomes the BDR.

For example: Router1 with router ID 50.50.50.50 is the DR, and Router2 with router ID 40.40.40.40 is initially the BDR. If 50.50.50.50 goes down, 40.40.40.40 becomes the DR. When 50.50.50.50 comes up, 40.40.40.40 remains the DR, and 50.50.50.50 becomes the BDR.

Q: When the command overflow database hard is executed, when the router enters overflow, the OSPF process is terminated. Is there a command or time-out to re-launch the process when the router enters overflow? Is it required to manually restart the OSPF process?

If the overflow database is configured with the parameter, hard, the OSPF process will be terminated when its LSA database reaches the maximum limit after which the OSPF process must be manually restarted.

Q: What is the behavior if the MAXLSAs are configured to a lesser value with the OSPF command, overflow database external?

If:

- A limit is put on the number of external LSAs on a router, for example 5
- The router learned 4 external routes from another router
- The router itself is generating more than 1 external LSA

Then all of these self-generated LSAs are flushed.

Once the router exits the overflow state, a trigger is sent to NSM to re-populate the self-generated LSAs. However, the router still has the 4 external LSAs learned from the other router, and once it gets the first self-generated LSA, it again goes into the overflow state and flushes the self-generated LSAs.

Only when the number of external routes from other routers is reduced, or the limit is increased, does the router exit this loop, and self-generated LSAs are installed.

Q: What is the meaning of “network” in the “network A.B.C.D/M area ID” command? Is it:

- Only network A.B.C.D/M can run OSPF?

or

- The network matched interface can run OSPF?

OSPF only runs on interfaces with IP addresses that match the specified network address.

ISIS

Q: How do we use the ISIS protocol on Ethernet when we have more than one interface in the same subnet?

ISIS needs to be configured in individual interfaces, for it to be enabled on that interface (or subnet); the scenario in which one of the two interfaces in the same subnet is configured for ISIS, and the other not. This is possible, as ISIS is a dual-stack protocol, and explicit interface-wise configuration is necessary.

Q: Does the ZebOS-XP ISIS module create adjacencies on point-to-point links?

Yes. ISIS has been tested using T1 interface cards. Interoperability testing with Cisco routers has been successfully completed.

Q: When do I get the ISIS “SRM already cleared” warning log?

This SRM flag warning should not generally occur.

If between the time the timer expires and is scheduled, the same LSP is received from a neighbor, the LSP does not then need to be flooded and the SRM flag is reset. In this case, when the timer expires, it checks the flag, acknowledges the LSP flag is not set, and does nothing except print the message.

Q: Why is the Sequence 1 LSP not included in the ISIS TLV?

The Sequence 1 LSP is created when the ISIS process is initiated. However, when the adjacent neighbor becomes active, the Sequence 1 LSP is refreshed, becomes Sequence 2, and is sent to the neighbor. This can be viewed in the output of the `show isis database` command.

Q: Which metric is used for narrow SPF calculation in the wide-transition metric style?

In wide transition, only the configured wide metric is considered for both narrow and wide SPF calculation. However, in narrow transition, only the configured narrow metric is considered for both narrow and wide SPF calculation.

Q: Do you support ISIS incremental/partial SPF calculation?

No, we do not support this functionality.

Q: Do you support IS-IS Fast Convergence?

Yes. We schedule SPF when the topology changes. If, between the current time and the time the last SPF is more than the SPF Interval (10 seconds), the hold-time period (default value, 2 seconds) is used as the schedule interval. If the value is less than 10 seconds, the number of seconds left for the 10 second timer to expire is calculated. If that value is greater than the hold time (2 seconds), the SPF at the SPF Interval delay (10 seconds) is calculated, otherwise, the hold-time delay (2 seconds) is used. However, the Exponential Backoff algorithm is not used.

Q: In ISIS, when an L1/L2 router is redistributing prefixes from L1 to L2, or L2 to L1, in which TLV should the prefix be sent?

When an L1/L2 router advertises an L1 route into L2, where that L1 route was learned via a prefix advertised in an “IP External Reachability Information” TLV, that L1/L2 router should advertise that prefix in its L2 LSP within an “IP External Reachability Information” TLV. L1 routes learned via an “IP Internal Reachability Information” TLV should still be advertised within an “IP Internal Reachability Information” TLV. These rules should also be applied when advertising IP routes derived from L2 routing into L1. And in this case, the up/down bit must also be set.

Q: Does ZebOS-XP support multiple instances in ISIS?

Yes, but the Layer-1 routes are not redistributed to the Layer-2 database between various instances.

Q: Does ZebOS-XP allow removing the last net in ISIS?

Yes.

Q: What is the ISIS behavior if max-lsp-lifetime is smaller than lsp-refresh-interval?

IP Infusion Inc. recommends making the lsp-refresh-interval smaller than the max-lsp-lifetime value because LSPs must be periodically refreshed before their lifetimes expire. Otherwise, LSPs time out before they are refreshed and are dropped from the database, if the lifetime is exceeded before a refresh LSP arrives.

Q: Does ZebOS-XP ISIS support the log-adjacency-changes command?

No.

Q: When dynamic host-name is disabled, is there a way to manually configure a static name to the system ID mapping table? Is there any command to show the ISIS host name in ZebOS-XP?

In ZebOS-XP, there is no option to manually configure a static name to the system ID mapping table. To see the host name, use the command show isis database details.

Q: What is the functionality of the “restart isis graceful” command?

If the `restart isis graceful` command is executed (with the optional grace-period parameter), ISIS graceful restart is initialized, and the ISIS daemon immediately shuts down. Also, the ISIS routing information and grace period is preserved by NSM. The ISIS daemon should be manually started after shut-down before the grace period expires to successfully complete the graceful restart.

Q: ISIS parameters configured on an interface are not deleted when the routing instance associated with it is removed.

To provide the flexibility of moving an interface to another routing instance, without having to re-configure the parameters (in case similar parameters are required).

MPLS

Q: For different tunnel IDs, two identical FTN entries can be created. Under the “show mpls forwarding table” output, both entries are displayed, and both are selected. In ASIC, which should be selected?

Multiple FTN entries for different tunnel IDs can be created: in this configuration, an error will not be displayed, because each FTN entry corresponds to a different tunnel ID. If two static FTNs are created for the same tunnel ID, an error will be displayed. In ASIC, only one FTN will be selected, and that depends completely on the ASIC implementation.

For example:

```
QA-125(config)#mpls ftn-entry tunnel-id 10 15.15.15.15/32 100 16.16.16.16 eth2
QA-125(config)#mpls ftn-entry tunnel-id 10 15.15.15.15/32 100 16.16.16.16 eth2
<--same ID
% Tunnel-ID 10 has been used for primary. To update ftn-entry, please provide
ftn index. <--error
QA-125(config)#mpls ftn-entry tunnel-id 11 15.15.15.15/32 100 16.16.16.16 eth2
<--different ID
```

```
QA-125(config)#mpls ftn-entry tunnel-id 12 15.15.15.15/32 100 16.16.16.16 eth2
<--different ID
QA-125(config)#end

QA-125#sh mpls forwarding-table
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
U - unknown FTN

Code FEC Tunnel-id FTN-ID Pri Nexthop Out-Label Out-Intf LSP-Type
K> 15.15.15.15/32 10 1 Yes 16.16.16.16 100 eth2 LSP_DEFAULT
K> 15.15.15.15/32 11 2 Yes 16.16.16.16 100 eth2 LSP_DEFAULT
K> 15.15.15.15/32 12 3 Yes 16.16.16.16 100 eth2 LSP_DEFAULT
```

Q: What is the purpose of the “mpls local-packet-handling” command? Why are only local TCP packets labeled? Is it due to the connection-oriented nature of TCP?

Yes, this command will label only TCP packets. By default, local packets are not labeled: this command is mainly for internal debugging purposes.

Q: In ZebOS-XP’s kernel MPLS forwarding code, there is no use of the EXP bit. Specifically, when attaching a label to an IP packet, 0 is always used as its EXP bit. Why are L-LSP and E-LSP only available in RSVP created LSPs, and not regular LSPs?

This functionality is not supported for the software forwarder, only for the control plane. This feature requires hardware integration.

Q: What is the purpose of the “mpls propagate-ttl” command?

This command is applied to the ingress LSR, and is enabled by default. The TTL value from the IP TTL is copied to the MPLS header TTL, and propagated into the MPLS cloud. If `no mpls propagate-ttl` is configured at the ingress LSR, and no mpls ingress-ttl value is specified, the default value (255) is copied into the MPLS TTL value. The pipe/uniform model should be applied to both the ingress and egress LSRs.

Q: Is it possible to bind multiple interfaces to the same label space?

In certain cases, yes; however, an MPLS-enabled interface is always bound to only one label space at a time.

Q: Do multiple Incoming Label Map (ILM) tables exist if per-interface label space is used?

No, there is one global ILM table, however, ILM entries can be distinguished based on the label space.

Q: Does the MPLS OAM code use the label stack in an incoming frame to validate the FEC in PDU? Please pinpoint the location in code where this validation takes place.

Refer to the `nsm_mpls_oam_decode_fec_tlv` function in `nsm/mpls/oam/nsm_mpls_oam_packet.c` to review the validation considerations. Note that there is no data plane support for MPLS OAM in Linux 2.6; only control plane support is available.

Q: What is a GMPLS-TE LSA? Is it possible to generate more than one GMPLS-TE LSA corresponding to two different neighbors using ZebOS-XP?

GMPLS information is coded in the Sub TLVs of the OSPF TE link TLV, and one opaque LSA contains one OSPF TE link TLV. This opaque LSA can also be called the GMPLS-TE LSA.

GMPLS-LSAs are dependent on links, rather than neighbors. There is no one-to-one correspondence between neighbors and GMPLS-LSAs. However, one GMPLS-LSA is generated for each link that is GMPLS capable.

LDP

Q: How many LSPs does ZebOS-XP support for LDP? Are there any restrictions on the number of LDP peers that can be established?

This depends on the memory available in the system. ZebOS-XP has no limitation of the number of LSPs for the MPLS forwarder. ZebOS-XP has no restrictions on the number of LDP peers.

Q: What is LDP-IGP synchronization?

LDP will not create an LSP until it sees a route in the NSM RIB. LDP registers with NSM, so once NSM installs a route into the RIB (for example, from IGP), it sends a message to LDP about this route. Even if LDP converges before IGP, because there are no routes in the RIB from IGP, LSPs are not created. If IGP converges before LDP, LDP will start creating LSPs because NSM has routes from IGP.

Q: Why is the ldp_nsm_init function called while creating the LDP instance, but not during system initialization, like RIP?

This reduces the burden on NSM. LDP will establish communication with NSM only when a user creates an instance (using the `router ldp` command), and not while starting the daemon, and disconnects communication with NSM when a user deletes the LDP instance (using the `no router ldp` command).

Q: Can I enable label switching on a loopback interface? If I use a loopback address as the transport address, the basic peer forms, but the TCP session does not come up.

A loopback interface cannot be enabled for label-switching. In a loopback interface, no packets will ever be received. The 127.0.0.1 address cannot be used as a transport address.

In an LDP configuration, the loopback address is used as a transport address. However, if transport address configuration is removed, the process picks up any available address as the transport address.

If the loopback address is being used for the LDP transport address, that loopback address has to be IP reachable (you should be able to ping that address). This is required because Session Init messages are sent directly to the advertised loopback address. If the address is unreachable, then the TCP session will never come up. You can advertise the loopback address either through OSPF or by using static addresses.

Q: Which types of label spaces are present in LDP?

There are two types of LDP label spaces:

- Per-interface: interface-specific incoming labels. This applies only when the LSP peers are directly connected over an interface, and the label is only used for traffic between those interfaces.
- Per-platform: platform-wide incoming labels.

Q: What should the source address be for sending an LDP targeted hello?

The source address for sending a targeted hello should be the configured transport address. If the transport address has not been configured, the targeted hello should be sent with the loopback address.

Q: After executing the “show ldp mpls-l2-circuit” command, why is the Layer-2 LDP VC state displayed as up, even if the VC forwarding entry is displayed as Inactive, after executing the “show mpls vc-table” command?

The entries displayed after executing the `show ldp mpls-l2-circuit` command pertain to the control plane, whereas, the entries displayed after executing the `show mpls vc-table` command pertain to the forwarding plane.

The state of the Layer-2 VC in the control plane is displayed as up, if the LDP signaling is working correctly, regardless of whether or not VC data is installed in the VC forwarding table. The VC forwarding entry is only displayed as Active when VC data is installed.

Q: According to RFC 3815, a user should be able to configure a keepalive hold-timer value from 1 to 65535 using the keepalive-timeout command, but ZebOS-XP checks with 3 times the keepalive interval.

Yes, according to RFC 3815, a user should be able to configure a value from 1 to 65535. However, ZebOS-XP checks while setting the keepalive hold timer with 3 times the keepalive interval as stated in the following link:

- http://www.ops.ietf.org/lists/shim6/msg01726.html

Keepalive messages keep identifying the remote TCP connection status with the neighbor. Thus, the keepalive-timeout value should be several times higher than the keepalive interval, because multiple keepalive messages are generated and should have time to reach the correspondent before they time out.

Q: When disabling LDP on an interface, the FECs are not getting cleared.

When the `router ldp` command is configured, LDP learns OSPF routes from NSM, creates FECs for the routes, and the FECs are present in the FEC table even though LDP is not enabled on the interface.

FECs are created per router LDP instance and not per LDP interface. The FEC table is cleaned in the following circumstances:

- Upon deleting the router LDP instance
- Upon restarting the LDP daemon
- Upon stopping the LDP daemon due to a SIGHUP signal

VR

Q: Is it possible to ping from a virtual router to another one, even if the involved interfaces belong to different networks, or if they are physically disconnected?

Ping between interfaces in the same machine is an expected behavior in native Linux kernels. This is because they do not support multiple FIBs for the different configured VRs. Since the FIB in the kernel holds a routing entry for each interface assigned to both the VRs, we are able to ping the interfaces, even if they are configured on different networks.

Q: Do the values set by the “max-static-routes” and “max-fib-routes” commands affect all of the VRs?

No, they only affect the default VR.

VRF

Q: What is the difference between VR and VRF?

VR and VRF are separate entities:

- VR is virtual routing
- VRF is virtual routing/forwarding used for VPN

VR is the software emulation of a physical router. The concept is to have logically separated routers on a physical router. Every Virtual Router (VR) has a unique VR ID, to identify the particular VR within the physical router.

VRF is within the VR, and has a unique ID per instance of a protocol (like BGP, OSPF, ISIS, and so on). VR and VRF together, determine the context of the protocol. VRF has its own database of interfaces (a subset of VR). NSM builds the VR and VRF context and distributes this information to protocol modules.

Q: Why does enabling VRF automatically enable MPLS?

ZebOS-XP supports only MPLS based Layer-3 VPNs. This is why MPLS is automatically activated when VRF is enabled.

Q: What is the difference between enabling VRF with and without MPLS?

The main difference is the way the routing table is populated:

- For VRF with MPLS, a customer site can exchange routes with another customer site, through the Layer-3 VPN tunnel created by BGP.
- For VRF without MPLS, a customer site can exchange routes only with the provider edge, but not the other customer site because there is no tunnel between the customers.

Refer to the BGP Command Reference.

Q: I am looking for the "--enable-vrf-overlap" configuration option. What is the VRF overlap feature, and does ZebOS-XP still support it?

The VRF overlap configuration option adds support for overlapping address spaces across:

- Multiple virtual routers
- Multiple OSPF instances for "draft-ishiguro-ppvpn-pe-ce-ospf"
- BGP-MPLS-VPN

From the ZebOS-XP 6.1 release onward, the overlapping address space feature is supported in ZebOS-XP by default, and does not require users to enable any configuration option. However, this support is available only when the operating system in question allows assigning of overlapping IP addresses.

On the other hand, the FTN does support multiple VRFs for MPLS routing.

VRRP

Q: What is the purpose of the circuit-failover command in VRRP?

The main purpose of the circuit-failover command is to increase the credibility of the default Backup router to become master when the default master goes down. This is achieved by configuring the priority-delta value as less than the priority configured.

Q: Can ZebOS-XP VRRP and OSPF run simultaneously? In my setup, OSPF is unable to establish adjacency on the VRRP routers.

Yes, both ZebOS-XP VRRP and OSPF can run simultaneously. However, if VRRP routers are configured incorrectly, OSPF might not be able to establish adjacency. This happens in cases where two routers are configured to be default Masters. When both the routers are in the master state, both will have the same MAC address, resulting in OSPF dropping packets (assuming that packets are self generated). In fact, the Master and Backup should not establish adjacency because they are the same “virtual” router.

This behavior is noticed only when OSPF is enabled on VRRP interfaces/networks. If ZebOS-XP VRRP and OSPF are enabled on different networks, then VRRP configuration does not affect OSPF.

Multicast

Q: What are the requirements to use the “mtrace” and “mstat” command?

The `mtrace | mstat` command needs a route to forward its multicast packets out of the box. PIM does not install any routes in the routing table: it relies on the routes provided by the other protocols, such as, OSPF, RIP, BGP or the kernel routes.

Q: What is the meaning of “ip mroute count route limit” and threshold?

The `ip multicast route-limit` command to limit the number of multicast routes that can be added to a router, and generate an error message when the limit is exceeded.

The default values of the limit and threshold are 2147483647.

IGMP

Q: In IGMP, will a group-specific query (GSQ) be flooded to all VLANs or only to the port on which the IGMP Leave is received?

A GSQ is sent to the port that received an IGMP LEAVE. If it does not get a response, and the host from which it received the LEAVE is the last member (or only member) of the group, the Non-Querier deletes the group, and sends a LEAVE to the Querier. On receiving a GSQ from the Querier, it floods to all the ports of that VLAN, because it no longer has the information of the originator that sent the IGMP LEAVE for that group.

Q: Why, when GMRP is configured on a bridge, is there a need to disable IGMP snooping?

IGMP Snooping needs to be disabled in order to enable GMRP. IGMP snooping and GMRP cannot be enabled at the same time on the same bridge. Refer to the *Layer 2 Configuration Guide* where this requirement is documented.

Q: Sometimes ZebOS-XP does not use the IP address of a VLAN interface even though the IP address is assigned on the interface and uses source IP as 0.0.0.0 for queries.

Refer to the extract below from section 1b of part 2.1.1 of RFC 4541, “IGMP and MLD Snooping Switches Considerations”, which discusses this special case:

“The 0.0.0.0 address represents a special case where the switch is proxying IGMP Queries for faster network convergence, but is not itself the Querier. The switch does not use its own IP address (even if it has one), because this would cause the Queries to be seen as coming from a newly-elected Querier. The 0.0.0.0 address is used to indicate that the Query packets are NOT from a multicast router.”

Q: In IGMP, is a Group-specific query flooded to all VLANs or only to the port on which the IGMP Leave was received?

A Group-Specific query is sent to the port from which an IGMP LEAVE was received. If there is no response and the host from which the LEAVE was received is the last, or only, member of the group, the Non-Querier deletes the group and sends a LEAVE to the Querier. Upon receiving a Group-Specific query from the Querier, it is flooded to all ports of that VLAN, because there is no longer any information about which port sent the IGMP LEAVE for that group.

Diff-Serv

Q: In Diff-Serv, only 7, instead of 8, classes are available as parameters with the “primary elsp-signaled” command.

Specifying a class with the `primary elsp-signaled` command means that the specified subgroup-class will be used for this E-LSP. If you want to select all 8 classes for an E-LSP, you do not need to specify a sub-group; simply do not specify a parameter with the `primary elsp-signaled` command.

Q: In Diff-Serv, I can create a preconfigured E-LSP even when no PHBs are supported at ingress (not even “be”). Is this a desired behavior?

Yes. According to RFC 3270, in case the preconfigured EXP-PHB mapping is not configured, the LSR should use a preconfigured EXP-PHB mapping, which maps all EXP values to the Default PHB.

The ZebOS-XP implementation allows the setting up of pre-configured E-LSPs, in all cases. In case there is no support for classes, and a preconfigured LSP is set up, then all incoming IP DSCPs are mapped to “be”.

When only PHB “be” is supported, and a preconfigured LSP is set up, this LSP carries only “be”. Packets with incoming IP DSCP of “be” are forwarded on this LSP, and other incoming IP DSCPs are dropped.

High Availability

Q: Do you have a description of the high-availability mechanism you implemented?

Yes. Please see the following link for a product description:

- http://www.ipinfusion.com/products/has/products_has.html

Q: Does ZebOS-XP support High Availability?

Yes: separate protocols can be run on individual line cards.

PIM

Q: Are there any APIs for SPT switchover in ZebOS-XP PIM-SM? If I implement SPT switchover in the forwarding engine, which type of message should be reported to ZebOS-XP PIM-SM?

ZebOS-XP PIM-SM does not support data-rate-based SPT switchover in the Linux forwarding engine. This is because most standard kernel multicast forwarders do not currently directly support PIM-SM switchover.

If you implement SPT switchover in your own forwarding engine, then the Keep Alive Timer Message (KATMSG) is required to be sent to the PIM-SM daemon. This starts the switchover.

Q: Can ZebOS-XP PIM-SM support only one register interface? If this is true, does ZebOS-XP PIM-SM use one RP for all group ranges?

First of all, we must disassociate the concept of register interface and the number of RPs. The register interface is used to get the whole multicast packet at the DR at source S, so that PIM can send Register packets to the RP. At the RP, the register interface is used for accepting incoming packets from source, and forwarding them based on (*,G) sources.

ZebOS-XP PIM-SM allows static RP and Candidate-RP configuration for only one group range 224/4. However, ZebOS-XP PIM-SM does support multiple group ranges that are reported from the BSR. This means you can configure

a router to be a static RP or a Candidate-RP only for one group range, but you can use multiple group ranges for RP selection of different groups.

Q: How can I support multiple RPs with one register interface?

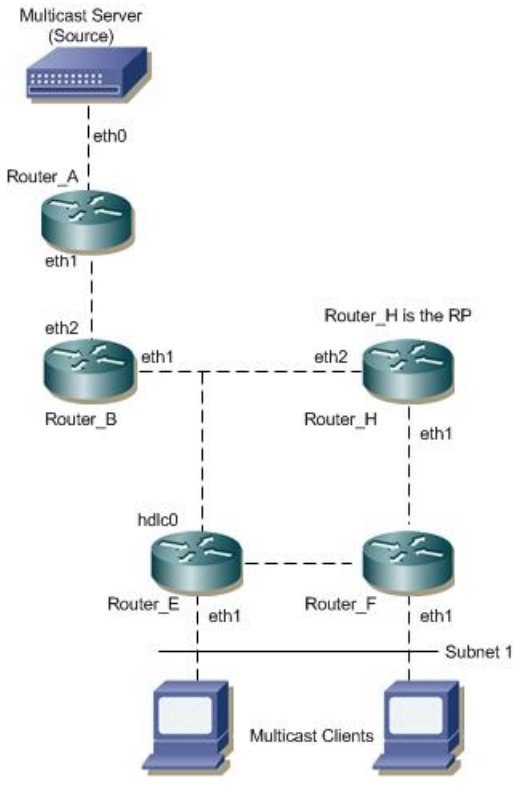
Here is a step-by-step description of how the register interface works, and how multiple RPs are supported by DR at the source.

1. When a packet arrives from a directly connected source, and the forwarder does not have a forwarding entry, the PIM daemon gets a NOCACHE message.
2. The PIM daemon initializes a Register state, and also a (S,G) state. The register interface is added to the outgoing list of the (S,G) forwarding entry in the kernel.
3. The kernel then forwards the packet on the (S,G) interface. When the register interface is in the outgoing list of the (S,G) entry, the kernel sends a WHOLEPKT message to the PIM daemon. The WHOLEPKT message has the whole IP multicast data packet.
4. The PIM daemon receives the WHOLEPKT message, looks up the RP for the group, and then sends a unicast Register message to the RP.

Since the RP for the group is known to the PIM daemon, multiple RPs can be supported.

Q: I want to know about the test beds used by IP Infusion Inc. for testing the PIM-SM daemon.

Following is one of our test beds. PIM-SM and OSPF-V2 are running on all the routers used in this topology. Refer to the illustration of the topology.



In this topology, router RH is the Rendezvous Point (RP). RA is directly connected to the source server, and RE is directly connected to a multicast client that sends a “Join” to receive a group.

After the multicast tree is established, the client is able to receive data from the server for that group.

Q: What are Forwarding Cache Record (FCR) entries in the PIM mroute table?

The initial multicast forwarders for the Linux and NetBSD operating systems were designed to support Dense-Mode multicast protocols, in which the receivers knew the sender's IP and vice-versa.

However, PIM Sparse-Mode supported (*,G) entries, in which the receivers were not required to be aware of the sender, as long as the multicast data was destined to group “G”. To support (*,G) forwarding using existing multicast forwarders supporting (S,G) forwarding of the operating systems, FCR entries were used.

For example, S1 and S2 are sending multicast traffic for group G, and receiver R1 sends a (*,G) join: FCR entries will be created for both (S1,G) and (S2,G). If S2 stops sending traffic, (*,G) is still maintained while the FCR entry for (S2,G) is deleted.

Q: What is the function of the generation identifier (genid) in PIM Hello messages?

The generation identifier contains a randomly generated 32-bit value that is regenerated each time PIM forwarding is started, or restarted, on the interface, including when the router itself restarts. When a Hello message with a new generation identifier is received from a neighbor, any old Hello information about that neighbor is discarded and updated with new Hello message information. This may cause a new designated router (DR) to be chosen on that interface.

Q: How many Rendezvous Points (RPs) can ZebOS-XP support? Is there any limitation?

There is no limit to the number of static and candidate RPs that ZebOS-XP can support.

Q: How should I enable PIM-DM on VLAN interfaces?

1. Assign an IP address on the VLAN interface.
2. Enable PIM-DM using the `ip pim dense-mode` command in VLAN interface mode.
3. Verify using the `show ip pim dense-mode interface` command.

Q: While configuring PIM, an interface named “Interface pimreg” is created. What is the use of this interface?

This interface is used by designated routers (DRs) and rendezvous points (RPs). On the DR side, the register packets are encapsulated and transmitted to the RP by this interface. On the RP side, the receiving and decapsulation of these packets happens through this interface. This interface is created by the Linux stack.

Q: The ZebOS-XP implementation of PIM-SM is running on our router. Will it be able to inter-operate with another router running another vendor's implementation of PIM-SM? Are there any inter-operability issues with ZebOS-XP PIM-SM?

Since our implementation is based on standards, it will inter-operate with any other implementation based on the standard.

Q: Can multiple static-RPs be configured? If yes, what algorithm is used for selection?

Yes, multiple static RPs can be configured. The selection algorithm is explained in the *PIM Command Reference* under the `ip pim rp-address` command.

Q: Can the same RP be configured with multiple group ranges? If yes, how is the deletion handled?

Yes, a single static-RP can be configured for multiple group ranges using Access Lists. However, the same IP address cannot be used as a parameter with two `pim ip rp-address` commands. Deletion of the RP-address is handled by removing the static-RP from all the existing group ranges and re-computing the RPs for existing TIB states if required.

For further details, see the `pim ip rp-address` command in the *PIM Command Reference*.

Q: Can the static-RP and BSR mechanisms be used simultaneously? If yes, what is the selection policy?

Yes, static-RP and BSR mechanism can be used simultaneously. If the RP is available from the BSR, it is selected first. If not, then the statically configured RP is selected.

SNMP

Q: How do you enable/disable some MIBs in SNMP during initial configuration?

There are options to enable/disable MIBs in SNMP while running the initial configuration script. For example, the option `--with-mib-modules="smux",` is to enable SNMP with SMUX during configuration of ZebOS-XP. The "mibII/ interface" options can be either enabled or disabled to use values from NSM.

Q: Does the SNMPC viewer return NSM's implementation of mibII/interfaces and mibII/system_mib, or net-snmp's default module?

When NSM is running, SNMP returns NSM 's implementation of mibII/interfaces and mibII/system_mib. When NSM is not running, it returns net-snmp's default values. To specifically override the default values from Linux, use the -I -tcp,ip,system_mib option while running ./snmpd.

Q: Which SNMP MIBs are supported by ZebOS-XP?

The MIB files are located under the respective protocol directory. For example, for BGP, you can find the MIB file in:

- `/bgpd/BGP4-MIB.txt`

The corresponding SNMP files are located under the protocol directory, as `protocolname_snmp.h` and `protocolname_snmp.c`. For example, for BGP, you can find the SNMP files in:

- `/bgpd/bgp_snmp.h`
- `/bgpd/bgp_snmp.c`.

Q: Which RMON MIB does ZebOS-XP support? (HC-RMON with 64 bit counters or RFC 2819?)

For SNMP operations in RMON, ZebOS-XP uses RFC 2819 as the basis for all of the counters. For interface-related counters, ZebOS-XP supports a wide range. These consist of counters defined in RFC 2020, section 3.3.6, and in RFC 2819.

For example, in our implementation of the `hal_if_counters` function we support brdc_pkts_rcv and brdc_pkts_sent as defined in the MIB-related entry for the interface in RFC 2020

Apart from this we also support counters like: pkts_65_127_octets and pkts_128_255_octets which are not part of a standard interface MIB structure, but are part of RMON related MIBs.

All counters in `hal_if_counters` are 64 bit counters. The value of these counters depends on the underlying hardware. Possibly, there are some extra counters supported by ZebOS-XP but not supported by the hardware, and vice versa.

Q: What is the default SNMP agent in ZebOS-XP?

SMUX. AgentX requires explicit enabling.

Q: Can you explain the SNMP callback function parameters?

The parameters of the get/getnext SNMP callbacks are as follows:

```
struct variable *v
```

Entry in the variableN array for the object.

Note: The name field of this structure has been completed into a fully qualified OID by pretending the prefix is common to the whole array.

`oid *name` OID from the request.

`size_t *length` Length of the OID.

`int exact` Flag to indicate whether this is an exact request (get/set) or an “inexact” request (getnext).

`size_t *var_len`

Length (in bytes) of the answer being returned.

`WriteMethod` Pointer to the set function (WriteMethod) for this variable.

`u_int32_t vr_id`

Virtual Router ID.

The parameters of the Set SNMP callbacks are as follows:

`int action` This variable will be used by SMUX, and it will have a value of action, such as, “commit” or “free”.

`u_char *var_val` Value of the variable to be set, such as, an IP address of “10.2.2.3”.

`u_char var_val_type`

Type of value, for example, integer, char, or IP addresses.

`size_t var_val_len`

Length of the value to be set, for example, if it is an integer, its length should be `sizeof(int)`.

`u_char *statP` This variable is also used by the SMUX which will have a return value of the callback function.

`oid *name` OID from the request.

`size_t length` Length of this OID.

`struct variable *v`

Entry in the variableN array for the object.

`u_int32_t vr_id`

Virtual Router ID.

Q: The IMI daemon has read/write access to ZebOS.conf. Is there a similar access from the SNMP path?

All ZebOS-XP modules act as a subagent to an SNMP agent. We can integrate with any third-party SNMP agent that supports the SMUX or AgentX protocol.

Platforms/Partners

Q: Is provider bridging (IEEE 802.1ad) supported in IPNET stack?

Yes.

Q: What is ipout.mod?

The ipout.mod module is built only after building ZebOS-XP with IPNET. The exact path of the outcome is ipcom-vxworks-r6_0_17/obj/ipout.mod.

Q: Can we control the ingress network flow on IPNET stack?

Yes, this can be achieved in IPNET stack. DiffServ and metering is supported in IPNET, which can be used to limit the rate of certain traffic flows.

To configure the DiffServ parameter, ioctl calls can be used.

Q: To which version of IPNET can the iproute2 patch be applied?

The patch should be applied to the iproute2 program, not IPNET. The iproute2 patch provided in ipcom-lkm-r6_0_20 is applied to iproute2, version 2.4.7.

Q: To which software partners is your code pre-integrated and tested?

Enea Element High-Availability middleware and Wind River Interpeak IPNET stack.

Q: Is your solution pre-integrated with the MontaVista package?

Yes. ZebOS-XP is supported on MontaVista.

Q: Is your solution pre-integrated with the MontaVista package applicable to both the CGL and the PRO addition?

Yes.

Q: Are there any redundant components that exist in both your package and the MontaVista package?

No.

Q: What added value do we get from the MontaVista integration with your solution compared to standard Linux?

The advantages are on the operating system side and in certain networking modules. ZebOS-XP is dependent on the TCP/IP stack, which is the same for both.

Q: Do you support a pre-integrated solution with Wind River? If so, which types?

We support PNE (for VxWorks) and LNE (for Linux) from Wind River.

Q: What is the partnership with MontaVista?

IP Infusion Inc. is in partnership with MontaVista, where we have access to all releases of MontaVista for supporting IP Infusion Inc. products.

Q: How do we locate the source code for the different modules?

Please refer to the builds.log file under the `ZebOS` directory to see the files included for every module.

For PAL related files:

PAL-MV = ZebOS/pal/linux

PAL-MV-IP = ZebOS/pal/linux

PAL-VXW = ZebOS/pal/vxworks

PAL-VXW-IP = ZebOS/pal/vxworks_ipnet2

For virtual routers, please check the files listed for zos-vr-xxx (bgp, ospf, rip).

Current and Planned Features

Q: Can you provide a road map?

The list of high-level features being addressed as part of future releases of ZebOS-XP is:

- Next Generation Transport Technologies, for example: PBT, PBB-TE, T-MPLS
- Network Operation and Maintenance, for example: Ethernet OAM Y.1731
- Multicast support for Next Generation services, for example: IGMP snooping over VPLS, P2MP
- Network Resiliency: BFD, BGP Graceful Restart for MPLS VPN
- High Availability for ZebOS-XP protocols, OSPF, RSVP-TE

Q: Do you support OSPF interaction with BGP, as detailed in RFC 1745?

Yes.

Q: Does ZebOS-XP support, or scheduled to support, VRF aware services for features such as ARP, ping, trace route, FTP, TFTP, or RFP?

The implementation for static ARP entries is not associated with any VRF and is associated with the default FIB. However, support for this type of VRF aware service is being considered.

Q: When will ZebOS-XP support RFC 5283 (LDP Extensions for Inter-Area Switched Paths), section 6.2?

Inter-Area LSP is part of the ZebOS 7.7.1 release. Please refer to the Customer Support Web page for information regarding the release date.

Q: Do you have a roadmap for SNMPv3 support?

SNMPv3 is a function of the SNMP Master agent. ZebOS-XP currently integrates with agents that already support SNMPv3. ZebOS-XP has support for SNMP v1/v2c/v3.

Q: Do your products complement the standard Linux, so that together, they can be defined as Carrier-Grade Linux (CGL 3.2/4.0) compliant?

Carrier-Grade Linux (http://www.linuxfoundation.org/news-media/announcements/2011/04/linux-foundation-releases-carrier-grade-linux-50-specification) compliance is planned for a future release.

Build

Q: When we insmod hsl.o, there are a lot of unresolved symbols in hsl.o, with prefix "bcmx_". How do we resolve them?

The unresolved symbols "bcmx_" encountered are available in the modules linux-kernel-bde.o, linux-uk-proxy.o and linux-bcm-diag-full.o. The above mentioned modules should be inserted before starting ./bcm.user.proxy:

```
$ insmod linux-kernel-bde.o debug=2 forceirq=2
$ insmod linux-uk-proxy.o
```



```
$ insmod linux-bcm-diag-full.o debug=2
```

Q: What is the purpose of the enable_tcp_message option and what will happen if this option is disabled?

The `--enable-tcp-message` option is used during compilation. By default, this option is disabled. The following further describes this option:

- `--enable-tcp-message`: TCP/IP sockets are used for interprocess communication between daemons.
- `--disable-tcp-message`: UNIX domain sockets are used for interprocess communication between daemons.

Q: What is the purpose of the msoft-float option?

This option turns on or off software emulation of floating-point operations and must only be turned off when the underlying hardware does *not* support floating-point operations.

Q: What is the meaning of HAVE_VRRP_LINK_ADDR?

For VRRP with link-address, the Virtual IP address is added to the interface that allows users to connect to the current Master VRRP router, regardless of whether or not it is the IP.

Please refer to the VRRP Configuration Guide for more details.

Q: How do I clean up the Layer-2 module object files?

Do `make clean-layer2` to clean up the layer2_module.o files. Doing `make clean` instead of `make clean-layer2` will not clean up the Layer-2 object files, only the protocol modules.

Q: What is the use of the --enable-l2lern option?

This option is used to enable MAC access lists in bridges. Refer to `nsn/nsm_mac_acl_cli.c` for the commands. When this option is disabled, software learning takes place, and when it is enabled, hardware learning takes place in the bridge.

Q: When you modify one of the files under /pal/linux/imi/, is it necessary to run the config.sh file again to make the daemons?

When you make changes in `Zebos/pal/linux/imi/pal_*.c` files, you need to do a `make clean`, and then do a `make nsm imi imish ...` under the `platform/linux` folder for the changes to take effect. There is no need to run `config.sh` again.

Index