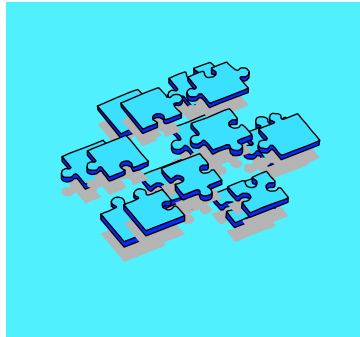


SHARE: S3933

IP Network Design Considerations: Diagnosis and Best Practices in z/OS

Mike Fox (mjfox@us.ibm.com)
Gwendolyn J. Dente (gdente@us.ibm.com)



Long Beach, CA, February 22-27, 2004
Thursday, February 26, 2004 at 1:30 pm
Seaside B

© IBM Corporation 2003

Abstract

➤ Prerequisites:

- Knowledge of Basic Routing Concepts for Static IP Routing
- Experience with implementing MVS or OS/3900-based TCP/IP

➤ Abstract: OSPF as part of OMPROUTE is the strategic direction for implementing a dynamic IP routing protocol in CS for OS/390. The link-state algorithms of OSPF bring great advantages to an IP network over those offered by the distance vector algorithms of RIP. As a result, many customers are choosing to move away from static routing and from the dynamic RIP routing of OROUTED to an OSPF implementation with OMPROUTE, available since OS/390 V2R6. To do this can be relatively simple if you understand the basic architecture of OSPF and if you understand how this architecture influences your implementation and coding choices in OS/390 (z/OS).

➤ This session is the third of three dealing with OSPF in CS for OS/390 or z/OS. The first session, 3914, describes the architecture of OSPF and the OSPF features that are implemented in OS/390 or z/OS. The second session, 3915, describes the implementation of OSPF in OMPROUTE in Communications Server for OS/390 or z/OS. This, the third session in the series, 3933, covers advanced topics. It makes recommendations on the integration of OS/390 (z/OS) into non-OSPF networks, and includes some examples of attaching to OEM routers and to zSeries LINUX. The topic highlights typical OSPF "neighbor" and network design problems and shows you how to diagnose OSPF problems with command displays and tracing.

Agenda

1. Diagnostics with OMPROUTE
 1. **OMPROUTE in the TCP/IP Stack**
 2. **Command Operations and Interpretation**
 3. **Trace Sample in OMPROUTE**
2. Connecting with Cisco Routers
 1. **CLAW**
 2. **MPC**
 3. **LCS**
 4. **QDIO**
 5. **MD5 Interoperability**
3. Integration with non-OSPF Networks
 1. **EIGRP**
 2. **Interoperability scenarios**
4. Integration with zSeries LINUX
5. Common Problems
 1. **Setup problems**
 2. **Performance problems**
 3. **Losing neighbors**
 4. **Routing problems**
6. Common Configuration Complaints
 1. **Subnet Mask**
 2. **Too Many Host Routes**



Warning:
Fast-moving
presentation!

1. This presentation moves at a brisk pace. Do not get discouraged -- we know you may not be able to absorb everything during the time allocated to this topic, but that is why we have included thorough notes!! The notes make it easy for you to review this presentation at your leisure at a later point in time.



1. Diagnostics with OMPROUTE

© IBM Corporation 2001, 2002, 2003

Tracing OMPROUTE

```
F OMPROUTE,TRACE=2
```

```
F OMPROUTE,TRACE=0
```

```
EZZ7800I OMPROUTE starting
EZZ7889I 00 ACTIVE COMP=SYSTCPRT SUB=NM2AOM
EZZ7845I Established affinity with NM2ATCP
EZZ7817I Using default OSPF protocol 89
EZZ7838I Using configuration file: /etc/omproute.conf.nm2y
...
EZZ7895I Processing console command - TRACE=2
EZZ7877I -- OSPF Packet Received -- Type: Hello
EZZ7878I OSPF Version: 2 Packet Length: 44
EZZ7878I Router ID: 10.0.1.1 Area: 1.1.1.1
EZZ7878I Checksum: ef9b Authentication Type: 0
EZZ7878I Hello_Interval: 10 Network mask: 255.255.255.0
EZZ7878I Options: E
EZZ7878I Router_Priority: 1 Dead_Router_Interval: 40
EZZ7878I Backup DR: 0.0.0.0 Designated Router: 0.0.0.0
...
EZZ8062I Subnet 10.0.0.0 defined
EZZ8062I Subnet 9.0.0.0 defined
```

```
> OTHER OPTIONS:
```

```
> omproute -t2 -d4
```

```
> CTRACE (CTIORA00)
```

```
> DEBUG Trace
```

```
>> F OMPROUTE,DEBUG=2
```

```
> F OMPROUTE,DEBUG=0
```

© IBM Corporation 2001, 2002, 2003

1. This detailed trace with formatted entries was created with the console command: "F OMPROUTE,TRACE=2."
2. The output is written to the destination defined in the STDENV member: "OMPROUTE_DEBUG_FILE."
3. This destination is dynamically created at OMPROUTE startup, and it is retained at the subsequent initializations of OMPROUTE.
4. You use the STDENV member OMPROUTE_DEBUG_FILE_CONTROL to specify the size and quantity of debug files maintained by OMPROUTE. When space is exhausted, files wrap.
5. The detailed trace is terminated with the MVS Console command: "F OMPROUTE,TRACE=0."
6. The UNIX variations of these "traceon" and "traceoff" commands are expressed with the parameter "-t2" and "-t0."
7. Level 2 support may request that you run a CTRACE with special characteristics (CTRACE file is CTIORA00; Component Name is SYSTCPRT) to capture interaction between OMPROUTE and the IP Stack.
8. IBM Service may request a Debug trace, which you may enable at OMPROUTE startup via the "dn" parameter (where "n" is any number between 0 and 4).

OMPROUTE Trace Details (1)

```

1  EZZ7800I OMPROUTE starting
   EZZ7889I 00  ACTIVE COMP=SYSTCPRT SUB=USER1464
   EZZ7845I Established affinity with TCPCS8
   EZZ7817I Using defined OSPF protocol 89
   EZZ7838I Using configuration file: /u/user146/omproute/omproute.conf
2  EZZ7883I Processing interface from stack, address 9.169.100.18,
   name CTC2, index 2, flags 451
...
4  EZZ7910I Sending multicast, type 1, destination 224.0.0.5 net 0
   interface CTC1
   EZZ7879I Joining multicast group 224.0.0.5 on interface 9.67.100.8
5  EZZ7913I State change, interface 9.67.100.8, new state 16,
   event 1
...
7  EZZ7908I Received packet type 1 from 9.67.100.7
...
8  EZZ7919I State change, neighbor 9.67.100.7, new state 4, event 1
9  EZZ7919I State change, neighbor 9.67.100.7, new state 8, event 3
   EZZ7934I Originating LS advertisement: typ 1 id 9.67.100.8
   org 9.67.100.8
10 EZZ7919I State change, neighbor 9.67.100.7, new state 16,
   event 14

```

Hello Packet

Interface State
(Pt-to-Pt)

Neighbor States

Router Link LSA

The trace and its description are taken from the IBM CS IP Diagnosis Guide, Chapter 25, for z/OS V1R4, z/OS V1R2, and OS/390 V2R10.

© IBM Corporation 2001, 2002, 2003

- Here's where the information you learned in Topic 1 of this series of OSPF presentations starts coming in handy! Topic 1 was an OSPF Tutorial.
- 1 OMPROUTE initializing (trace level 1 was specified at startup)
- 2 OMPROUTE learns of TCP/IP stack interfaces
- 4 OSPF Hello packet sent out OSPF interface
- 5 OSPF Interface transitions to state "point-to-point"
- 7 OSPF Hello packet received from OSPF neighbor
- 8 OSPF neighbor transitions to state "Init"
- 9 OSPF neighbor transitions to state "2-Way"
 - EZZ7934I = Type 1 LSA: Router Links
 - Describes all router's interfaces into an area and their states (Intra-area destinations)
- 10 OSPF neighbor transitions to state "ExStart"

OMPROUTE Trace Details (2)

```

11 EZZ7910I Sending multicast, type 2, destination 224.0.0.5 net
    0 interface CTC1
12 EZZ7908I Received packet type 2 from 9.67.100.7
13 EZZ7919I State change, neighbor 9.67.100.7, new state 32, event 5
14 EZZ7910I Sending multicast, type 3, destination 224.0.0.5 net 0
    interface CTC1
    EZZ7908I Received packet type 2 from 9.67.100.7
15 EZZ7908I Received packet type 4 from 9.67.100.7

16 EZZ7928I from 9.67.100.7, new LS advertisement: typ 1 id
    9.67.100.7 org 9.67.100.7
...
17 EZZ7910I Sending multicast, type 4, destination 224.0.0.5 net
    0 interface CTC1
...
18 EZZ7919I State change, neighbor 9.67.100.7, new state 128,
    event 6
19 EZZ7908I Received packet type 5 from 9.67.100.7
20 EZZ7910I Sending multicast, type 5, destination 224.0.0.5
    net 0 interface CTC1
...
21 EZZ7949I Dijkstra calculation performed, on 2 area(s)

```

2: DB Description
Packets

3: Linkstate
Request Packet
4: Linkstate
Update Packet

The trace and its description are taken from the [OS/390 V2R10.0 IBM CS IP Diagnosis Guide, Chapter 25.](#)

© IBM Corporation 2001, 2002, 2003

- 11 OSPF Database Description packet sent out OSPF interface
-
- 12 OSPF Database Description received from OSPF neighbor
-
- 13 OSPF neighbor transitions to state "Exchange"
-
- 14 OSPF Link State Request packet sent out OSPF interface
-
- 15 OSPF Link State Update packet received from OSPF neighbor
-
- 16 Link State Advertisements from received Update packet are processed
-
- 17 OSPF Link State Update packet sent out OSPF interface
-
- 18 OSPF neighbor transitions to state "Full"
-
- 19 OSPF Link State Acknowledgment packet received from OSPF neighbor
-
- 20 OSPF Link State Acknowledgment packet sent out OSPF interface
-
- 21 OSPF Dijkstra calculation is performed
-
- NOTE: Several Redbooks include discussions of OSPF protocols.
 - The IBM Communications Server for z/OS V1R2 TCP/IP Implementation Guide, Volume 4, Connectivity and Routing, SG24-6516-02, contains a chapter on OMPROUTE that includes a discussion of the various states that neighbors experience as the OSPF protocol manages the network.
 - The IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide, Volume 1, Configuration and Routing, SG24-5227-02, contains a chapter on OMPROUTE that includes a discussion of the various states that neighbors experience as the OSPF protocol manages the network.



2. Connecting to CISCO Routers

© IBM Corporation 2001, 2002, 2003

Information: IBM & CISCO Interoperability

- Redbook on OSPF, EIGRP, MNLB, Sysplex Distributor:
 - Networking with z/OS and Cisco Routers: An Interoperability Guide (SG24-6297)
 - IBM Communications Server for z/OS V1R2 Implementation, Volume 4: Configuration and Routing (SG24-6516)
- Whitepaper on OSPF with IBM & CISCO:
 - "OSPF Design and Interoperability Recommendations for Catalyst 6500 and OSA-Express Environments"
http://www-1.ibm.com/servers/eserver/zseries/networking/pdf/ospf_design.pdf
- CISCO WebSite:
 - http://www.cisco.com/warp/public/100/omproute_mainframe.html

CLAW Coding with CISCO

OMPROUTE.CONF for Claw to CISCO

```

OSPF_Interface
Name = CISCO2
IP_Address = 9.67.156.66
Destination_Addr = 9.67.156.65
Attaches_To_Area = 0.0.0.0
MTU = 4092
Retransmission_Interval = 5
Transmission_Delay = 120
Hello_Interval = 30
Dead_Router_Interval = 120
Cost0 = 25
Router_Priority = 2
Subnet = Yes
Subnet_Mask = 255.255.255.248;
    
```



Consult APAR PQ48766 for information about CLAW implications for OSPF and RIP!

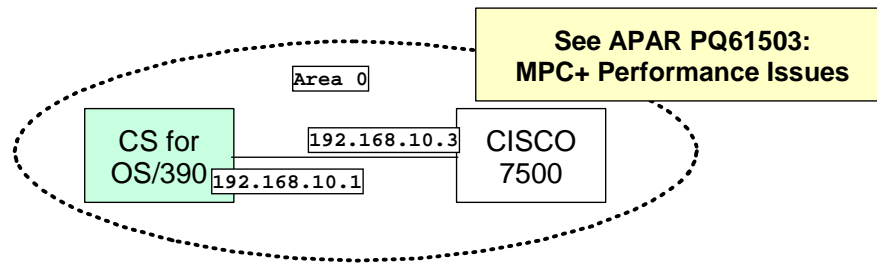
```

interface Channel6/0 CISCO Claw to MVS001
description CLAW Connection to MVS001 and MVS069
ip address 9.67.156.65 255.255.255.248
ip directed-broadcast
ip mtu 4092
ip ospf network point-to-multipoint
load-interval 30
no keepalive
claw D900 10 9.67.156.66 MVS001B C7507A PACKED PACKED broadcast
claw D900 12 9.67.156.67 MVS069B C7507B PACKED PACKED broadcast
end
    
```

© IBM Corporation 2001, 2002, 2003

1. Traditionally OS/390 or z/OS views the CLAW connection as point-to-point. The CIP-attached CISCO router views the CLAW connection as point-to-multipoint. As a result, the router will send information about a network or subnet (for example, 192.168.1.0/24) to the dynamic routing protocol (RIP or OSPF) in OS/390 or z/OS. OS/390 or z/OS assumes that there is only one other HOST in this subnet besides itself: for example, 192.168.1.1/32 might represent the mainframe side of the CLAW connection and 192.168.1.2/32 might represent the router side of the CLAW connection. If this is truly the case, the CLAW link coding without the new P2MP parameter will allow OS/390 or z/OS to learn successfully about the RIP or OSPF network. If, however, the entire network design is such that OS/390 or z/OS begins to learn about other hosts in the same subnet, the routing protocol on the mainframe will discard knowledge of those other hosts and the knowledge of the network breaks down. So, for example, if the mainframe learns through the routing protocols over either this CLAW connection or other connections that there are other HOSTS in the network with addresses 192.168.1.3, 192.168.1.4, and so on, the routing protocol at the mainframe will discard this information, and the IP routing table will not learn the true nature of the network design. This is the case that requires the use of the P2MP keyword on the LINK statement for the CLAW device.
2. What should the CLAW implementer at the mainframe do? Obviously there are cases in which the current coding need not be changed, but it does not hurt to add the P2MP keyword even in these cases to avoid any possibility that in future other hosts in the same subnet or network will be introduced. The interface type (either P2P or P2MP) will be reflected in the output from the command 'D TCPIP,OMPR,OSPF,IF.' If you have coded P2MP, the output will reflect P2MP regardless of the number of hosts learned by the protocol. If you have coded nothing on the LINK statement, the output will reflect P2P.
3. Summary: Prior to APAR PQ48766, OS/390 OSPF perceives a CLAW interface as point-to-point, whereas CISCO codes it to be point-to-multipoint. Note that Cisco, depending on the level of IOS, may not have a CLAW configuration parameter for P2P; it can be configured using the P2MP and BROADCAST parameters to treat CLAW like P2P for compatibility with our CLAW. However, we recommend applying APAR PQ48766 to avoid this workaround.
 1. It may be necessary to implement APAR PQ48766 (closed July 30, 2001) in order to code the new parameter "P2MP" for potential problems with OSPF and RIP over a CLAW interface to a CISCO router.
 2. P2MP
 1. Treat this CLAW link as a point-to-multipoint link.
 2. The default is point-to-point.
 3. Point-to-multipoint RIP neighbors with which omproute will exchange routing information are learned via RIP_INTERFACE NEIGHBOR statements or upon receipt of a RIP Update from the Same-subnet neighbor.
 4. At z/OS V1R2, the APAR PQ51213 added the P2MP parameter to the CLAW LINK definition to overcome the problems with APAR PQ35265, which made the CLAW interface look both P2P and P2MP.
 5. NOTE: The "P2MP" LINK coding (APAR PQ48766) is independent of the "PACKED" feature also represented on this page. The two features can be used independently of each other. (That is, you need not code "PACKED" in order to implement the LINK coding "P2MP.")
4. *****
- 5.; CISCO CLAW DEFINITIONS in OS/390 TCP/IP
- 6.; *****
- 7.;
8. DEVICE CIP2A CLAW D20 MVS001B C7507A PACKED 15 15 4096 4096
9. LINK CISCO2 IP 0 CIP2A
10. If you don't want to use the PACKING feature, you must change PACKED PACKED to TCPIP TCPIP in the CLAW statements.
11. In the TCP/IP profiles, you must change PACKED to NONE.
12. At the time of this writing, only Cisco model 7200-series routers with Channel Port Adapters (ECPAs or PCPAs) and 7500-series routers with CIP cards support CLAW in packed mode. (Verify CISCO Microcode levels.) The CLAW statement within the Cisco router must be respecified to enable packing (refer to Cisco configuration instructions for more information), and the MTU for the channel (CIP) interface on the router must be set to 4092 via the IP MTU command. Failure to reduce CIP MTU to 4092 will result in unpredictable behavior, including potential hangs of the CIP card.
13. If PACKED operation is specified, TCP/IP sets READ and WRITE buffer sizes to 4096, regardless of user-specified values. This enables the packing of small packets into 4K frames.
14. Consult INFO APARS I112353, I112361, and I112494 for more information on PACKING. Also see New Function APAR PQ41205 for V2R8 and V2R10. This new function is intended for configurations in which the average datagram size carried across the channel is less than 1812 bytes. Customers accessing 1500-byte Ethernet on the other side of the router will benefit with this change.
15. At z/OS V1R2, 60K CLAW packing is available.

MPC+ Coding with CISCO



OMPROUTE.CONF for MPC+ to CISCO

CISCO MPC+ MVS

```

; CIP 3C
OSPF_INTERFACE
  IP_address=192.168.10.1
  Name=LNCIP3C
  Cost=3
  Hello_Interval=2
  Dead_Router_Interval=8
  Subnet_mask=255.255.255.0
  MTU=4000
  Router_Priority=0
;

```

```

interface Channel1/0
  cmpc 0100 00 TG00 READ
  cmpc 0100 01 TG00 WRITE
!
interface Channel1/2
  ip address 192.168.10.3 255.255.255.0
  no ip directed-broadcast
  ip route-cache flow
  ip ospf network point-to-point
  ip ospf hello-interval 2
  no ip mroute-cache
  no keepalive
  tg TG00 ip 192.168.10.1 192.168.10.3 broadcast

```

© IBM Corporation 2001, 2002, 2003

1. MPC+ can be perceived by OS/390 or z/OS as either point-to-point or point-to-multipoint. If the RIP or OSPF protocol determines that there are more than two hosts in the (sub)network represented by the MPC+ connection, the interface will be perceived as point-to-multipoint. If, on the other hand, the RIP or OSPF protocol determines that the only two members of the (sub)network are the mainframe and router sides of the connection, that is, that there are only two members of the (sub)network, the interface will be perceived as point-to-point. In OSPF the interface type is reflected in the output from the command "D TCPIP,,OMPR,OSPF,IF.
2. More examples of MPC+ to a CISCO -- EMIF vs. non-EMIF configuration is irrelevant for considerations about coding "ospf network point-to-point" or "ospf network point-to-multipoint." The sample below works equally well in either an EMIF or non-EMIF environment.

```

interface Channel6/0
  description CPMC Channel to MVS074
  no ip address
  no ip redirects
  ip directed-broadcast
  load-interval 30
  no keepalive
  cmpc 8A11 70 TGMVS074 READ
  cmpc 8A11 71 TGMVS074 WRITE
end

```

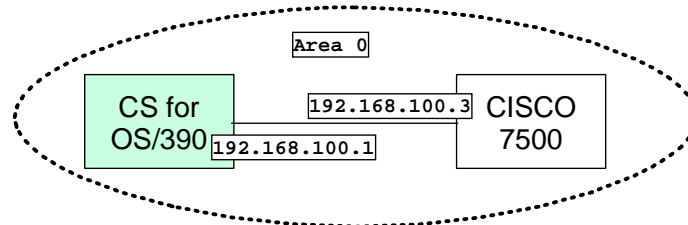
```

interface Channel6/2
  description CMPC Channel
  ip address 9.67.157.198 255.255.255.248
  ip ospf network point-to-multipoint
  no keepalive
  tg TGMVS074 ip 9.67.157.193 9.67.157.198 broadcast
end

```

1. You may discover that an ESCON director in the middle of a connection requires that you code point-to-multipoint on the Cisco side of the connection, regardless of whether your mainframe attachment is EMIF or non-EMIF.

CISCO with LAN LCS Environment



```

; OMROUTE.CONF for LCS
OSPF_INTERFACE
  IP_address=192.168.100.1
  Name=LNKOSA1C
  Subnet_mask=255.255.255.0
  Demand_Circuit=no
  Attaches_To_Area=0.0.0.0
  MTU=1500
  Retransmission_Interval=5
  Transmission_Delay=1
  Router_Priority=1
  Hello_Interval=2
  Dead_Router_Interval=8
  Cost0=3
;

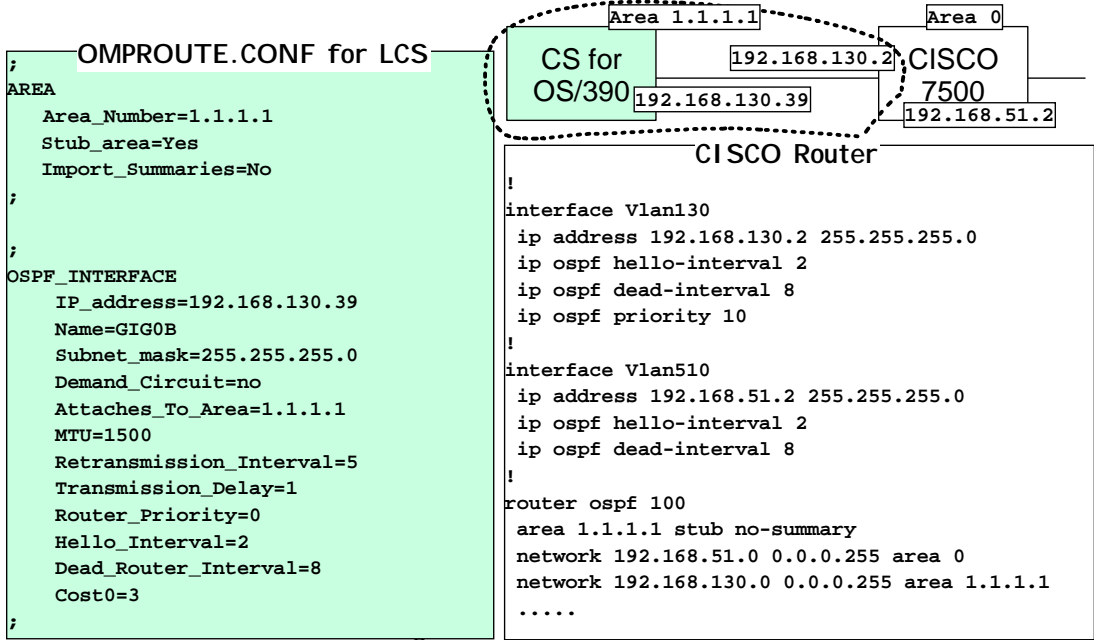
```

```

CISCO Router
!
interface FastEthernet6/0/0
  ip address 192.168.100.3 255.255.255.0
  no ip directed-broadcast
  ip route-cache flow
  ip route-cache distributed
  ip ospf hello-interval 2
  full-duplex
!
!
router ospf 100
  network 192.168.100.0 0.0.0.255 area 0
  .....
!

```

CISCO Example with OSA-E QDI O in Totally Stubby Area



© IBM Corporation 2001, 2002, 2003

1. Many examples of Cisco interfacing with CS in an OSPF network may be found in a Whitepaper on OSPF with IBM & CISCO:
 1. "OSPF Design and Interoperability Recommendations for Catalyst 6500 and OSA-Express Environments"

http://www-1.ibm.com/servers/eserver/zseries/networking/pdf/ospf_design.pdf
2. The parameter "Import_Summaries=NO" on the CS side is optional because the ABR (i.e., the Cisco router) has already indicated that "no-summary" should be sent to CS.

MD5: IBM & Cisco, 'Old Way'

```

USER1:/u/user1: >pwtokey -p HMAC-MD5 -u auth 'ABCDEFGHJKLMNPO'
Display of 16 byte HMAC-MD5 authKey:
baf081c4f34ee7cb810e264d606c3394

```

IBM: Standard MD5 Key

```

interface Ethernet0
 ip address 48.51.32.224 255.255.240.0
 ip ospf message-digest-key 4 md5 ABCDEFGHJKLMNPO
 ip ospf hello-interval 30
 ip ospf retransmit-interval 60
end

```

Cisco: Non-Standard

```

OSPF_Interface
Name = LO6CETH
IP_Address = 48.51.32.156
Subnet_Mask = 255.255.240.0
Hello_Interval = 30
Retransmission_Interval = 60
Dead_Router_Interval = 120
Authentication_Type = Md5
Authentication_Key_ID = 4
Authentication_Key = 0x4142434445464748494A4B4C4D4E4F50

```

ABCDEFGHJKLMNPO
in ASCII Hexadecimal

© IBM Corporation 2001, 2002, 2003

1. IBM provides a utility, "pwtokey," that can be used to generate an MD5 authentication key. For example, given the character string "ABCDEFGHJKLMNPO," the "pwtokey" utility may be run as follows to generate a valid MD5 key:

1. USER1:/u/user1: >pwtokey -p HMAC-MD5 -u auth 'ABCDEFGHJKLMNPO'
2. Display of 16 byte HMAC-MD5 authKey:
3. baf081c4f34ee7cb810e264d606c3394

2. As long as the participating routers follow the MD5 convention, the OSPF_Interface in OMPROUTE may be coded to reflect the generated key as follows:

```

OSPF_Interface
Name = LO6CETH
IP_Address = 48.51.32.156
Subnet_Mask = 255.255.240.0
Hello_Interval = 30
Retransmission_Interval = 60
Dead_Router_Interval = 120
Authentication_Type = Md5
Authentication_Key_ID = 4
Authentication_Key = 0xbaf081c4f34ee7cb810e264d606c3394

```

3. However, be aware that some routers -- Cisco being one of them -- may generate keys for MD5 authentication using nonstandard methods, so that care must be taken to ensure that Cisco and OMPROUTE define the same key. The character value entered on the Cisco side of a network connection must be converted to HEXADECIMAL in ASCII format on the IBM side of the connection and must not have been generated with the "pwtokey" utility. The following example illustrates this:

1. The Cisco box needs to be configured like the following:

```

interface Ethernet0
 ip address 48.51.32.224 255.255.240.0
 ip ospf message-digest-key 4 md5 ABCDEFGHJKLMNPO
 ip ospf hello-interval 30
 ip ospf retransmit-interval 60
end

```

2. Here is the configuration in z/OS OMPROUTE:

```

OSPF_Interface
Name = LO6CETH
IP_Address = 48.51.32.156
Subnet_Mask = 255.255.240.0
Hello_Interval = 30
Retransmission_Interval = 60
Dead_Router_Interval = 120
Authentication_Type = Md5
Authentication_Key_ID = 4
Authentication_Key = 0x4142434445464748494A4B4C4D4E4F50

```

1. This key is ASCII HEX for ABCDEFGHJKLMNPO

4. The key id is a one-byte constant that identifies the key for MD5 authentication. You should really consider it to be part of the key, and it must match on both sides of a connection. Some platforms (including Cisco) support multiple keys. The key id identifies which key is being used. IBM supports only one key, but we provide the key id for compatibility, permitting it to be defined to match what Cisco is using, for example.

MD5 Authentication, IBM & Cisco

APAR PQ73021 or z/OS V1R5

```
interface Ethernet0
  ip address 48.51.32.224 255.255.240.0
  ip ospf message-digest-key 4 md5 ABCDEFGHIJKLMNOP
  ip ospf hello-interval 30
  ip ospf retransmit-interval 60
end
```

OSPF_Interface

Name	= LO6CETH
IP_Address	= 48.51.32.156
Subnet_Mask	= 255.255.240.0
Hello_Interval	= 30
Retransmission_Interval	= 60
Dead_Router_Interval	= 120
Authentication_Type	= Md5
Authentication_Key_ID	= 4
Authentication_Key	= A"ABCDEFGHIJKLMNQP"

© IBM Corporation 2001, 2002, 2003

1. Prior to V1R5 OMPROUTE only accepts OSPF MD5 keys on the OSPF_INTERFACE AUTHENTICATION_KEY statement as hexadecimal strings. The function available with APAR PQ73021 for V1R2 and V1R4 adds support for ASCII text MD5 keys similar to the MD5 text keys used with Cisco routers.
2. Beginning with z/OS V1R5, this method will be included in the base code, with no APAR needed.
3. Further notes: The standard method to code MD5 authentication keys is with a 16-byte hexadecimal string beginning with 0x (0x plus 32 hexadecimal characters). In some cases, pwtokey can be used to generate hexadecimal MD5 keys (refer to z/OS Communications Server: IP System Administrator s Commands). An additional method, which provides compatibility with Cisco, Extreme, and other vendor routers that use a Cisco-compatible CLI interface, is to code the MD5 key as an ASCII string specified in double quotes prefixed with an A. For example, to be compatible with the Cisco key definition of "ABCDEFGHIJKLMNQP," you would code as indicated in the visual above, prefixing the key value with an "A" for ASCII.



3. Integration with Non-OSPF Networks

© IBM Corporation 2001, 2002, 2003

OSPF/EIGRP/BGP Interoperability

EIGRP is a proprietary routing protocol supported by Cisco routers. BGP is a standards-based protocol supported by many different vendor routers.

- CS for z/OS does not support EIGRP or BGP.
- However, OSPF and EIGRP|BGP interoperate effectively using the appropriate routers as the ASBRs.

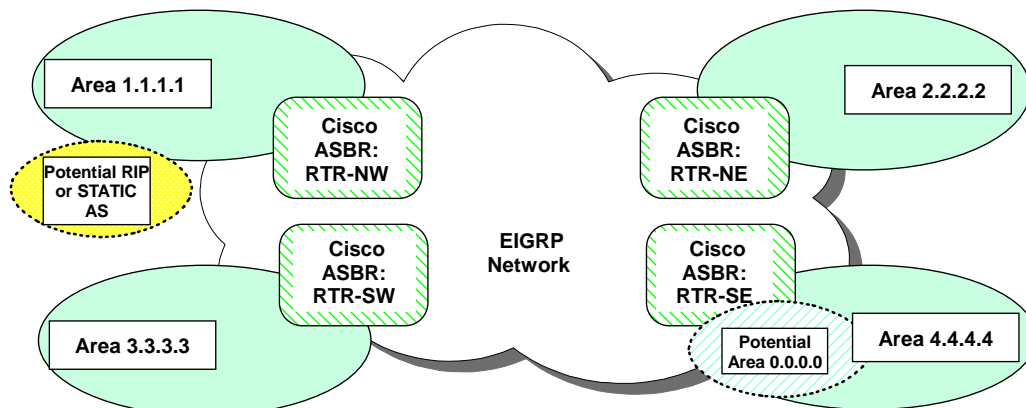
Though EIGRP and BGP are used in the following examples, in fact the backbone autonomous system could be any non-OSPF protocol.

- How to configure the routers to interoperate EIGRP, BGP, and OSPF is beyond the scope of this presentation. See Redbooks SG24-6297 & SG24-6516.

© IBM Corporation 2001, 2002, 2003

1. This information is useful for customers who have an existing Cisco EIGRP backbone and want to attach z/OS sysplexes or mainframe complexes to them, and use dynamic routing.
2. This information is also useful to customer wanting to implement "filtering" between OSPF areas by designing them as separate Autonomous Systems interconnected by ASBRs running EIGRP or BGP.
3. Note that OMPROUTE does not support EIGRP or BGP; however it may interoperate with routers acting as ASBRs that do support these protocols.
 1. Cisco supports the proprietary routing protocol named EIGRP.
 2. Many other router vendors, including Cisco, provide routers that support BGP.

EIGRP + OSPF as Non-Backbone

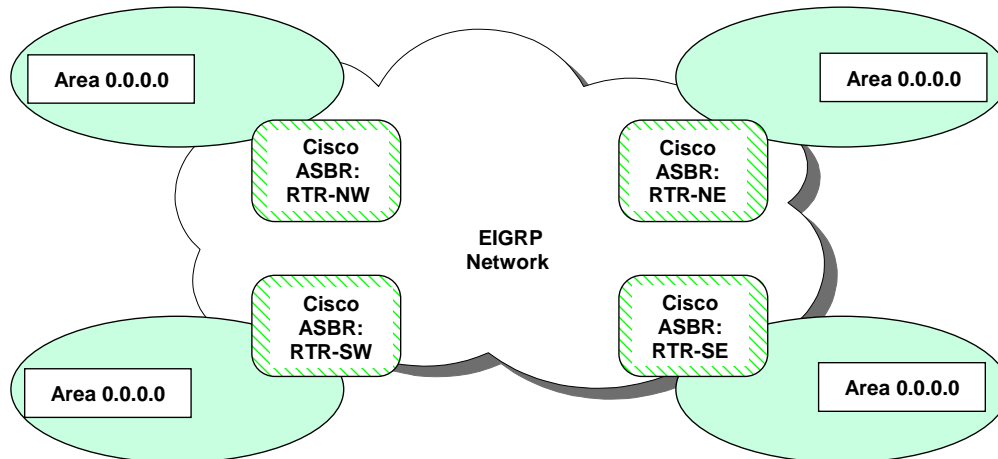


- **OSPF areas may be attached as non-backbone to an EIGRP Autonomous System.**
 - **All routing between OSPF areas occurs through ASBRs and the EIGRP AS.**
 - **Cannot connect a second area to any of the OSPF areas unless second area is a backbone area (Area 0).**
 - **Permits strategy to insert non z/OS routers as Area 0 at a later date between ASBR and the z/OS non-backbone area or to convert EIGRP to Area 0.**
 - **Permits strategy to connect another AS to any of the OSPF areas.**
 - **A Direct Route between two OSPF areas (e.g., Area 1 to Area 2) does not exist because there is no required backbone area (Area 0) between them.**

© IBM Corporation 2001, 2002, 2003

1. ASBRs will be configured to ship only default routes into the OSPF areas.
2. Since there will be relatively small routing trees that need to be maintained within the OSPF areas, a Stub Area or Totally Stubby configuration in the z/OS area is not necessary. This allows the attachment of RIP or Static Autonomous Systems to any of the OSPF areas because any router in them may be an ASBR.
 1. Remember: ASBRs cannot import external routes into an OSPF Stub Area.

EIGRP + OSPF as Backbone

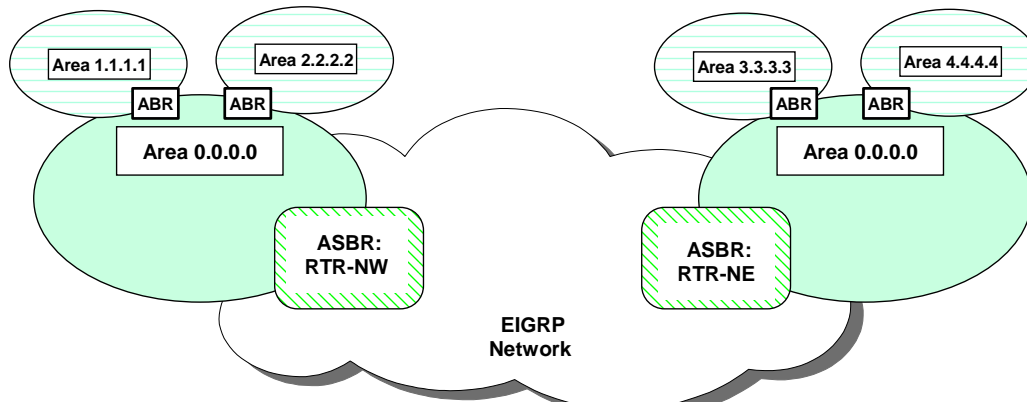


- **OSPF areas may be attached as backbone areas to an EIGRP Autonomous System.**
 - **All routing between OSPF areas occurs through an ASBR and the EIGRP AS.**
 - **The Area 0s are isolated from each other via ASBR connections and are unaware of the existence of another "Area 0." (Ergo: No conflict with multiple Area 0s.) Permits use of EIGRP "redistribute" statements for filtering!**
 - **A Direct Route between two OSPF areas (e.g., Area 1 to Area 2) does not exist; a virtual link cannot be built between OSPF areas across a separate AS (EIGRP).**

© IBM Corporation 2001, 2002, 2003

1. OSPF by its very architecture provides for no filtering algorithms as does RIP. This is because OSPF requires that every router in an OSPF area must have the same image of the area topology as every other router. Filtering would prevent the building of identical link-state databases within an area. However, there are means within OSPF to "filter" out advertisements across areas and between autonomous systems. You heard about these methods in the presentation about OSPF architecture:
 1. Creating Stubby Areas
 2. Creating Totally Stubby Areas
 3. Creating an ASBR that does not "import" (or "redistribute") certain types of advertisements into OSPF
 4. Creating an ABR that suppresses the advertisement of a network through the RANGE statement.
2. By placing EIGRP between two OSPF Autonomous Systems and allowing a Cisco Router to be the ASBR between OSPF and EIGRP, you can take advantage of the Cisco "redistribute" statement to filter OSPF advertisements between the two OSPF AS's.

EIGRP + OSPF: Multiple Areas

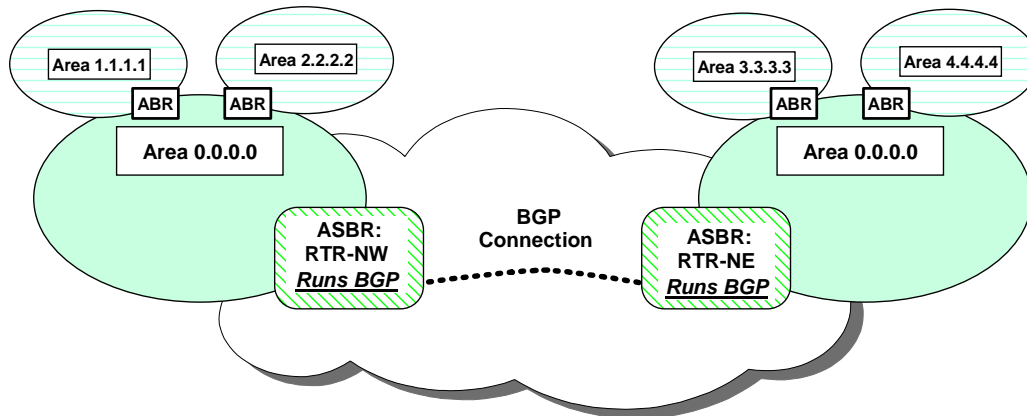


- An OSPF AS with multiple areas may be attached to an EIGRP Autonomous System.
 - All routing between OSPF areas in an AS occurs through ABRs.
 - No need for the OSPF areas within the AS to communicate with each other via the EIGRP AS.
 - The Area 0s are still isolated from each other via ASBR connections and are unaware of the existence of another "Area 0." (Ergo: No conflict with multiple Area 0s.)
 - Area 0's may be physically interconnected to bypass the EIGRP network.

© IBM Corporation 2001, 2002, 2003

1. This is a variation of the theme introduced on the previous page.

OSPF and BGP



- This diagram is a variation on the previous diagram.
 - The previous diagram depicted a Cisco ASBR running both OSPF and EIGRP, to provide connectivity across Autonomous Systems and to provide even "filtering" capability between the OSPF Autonomous Systems.
 - This diagram depicts any ASBR capable of running both OSPF and BGP, also providing connectivity across Autonomous Systems and "filtering" capability between the OSPF Autonomous Systems.

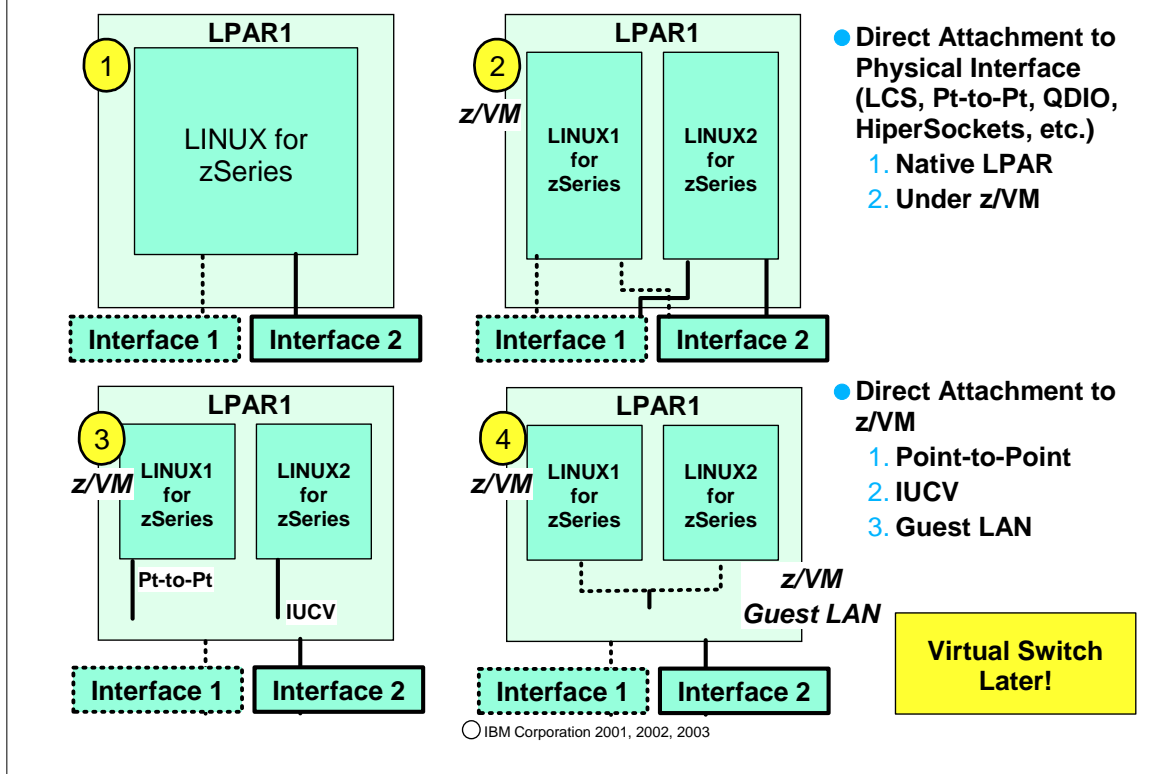


4. Integrating LI NUX on zSeries into a Network with z/OS



© IBM Corporation 2001, 2002, 2003

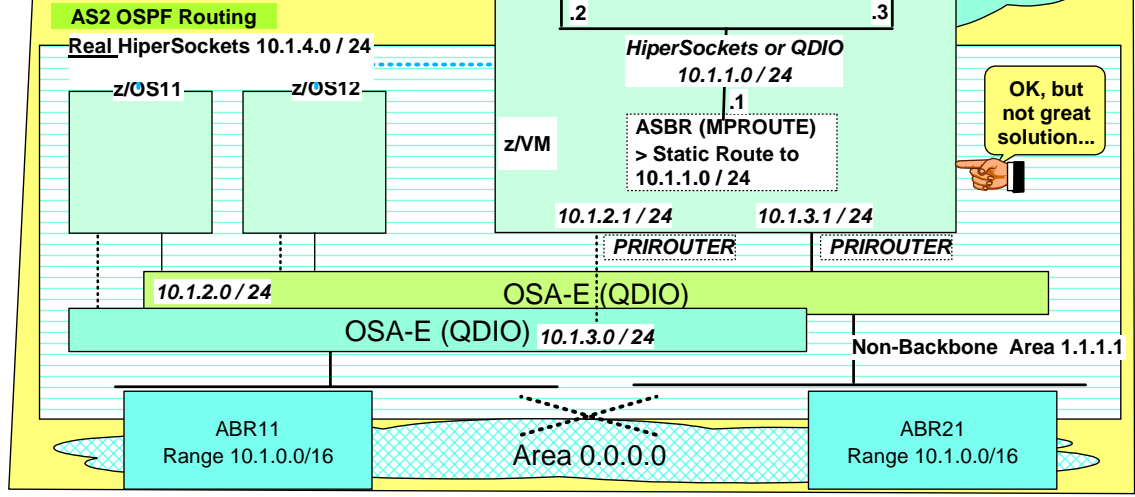
Physical Network Design for zSeries Linux



- zSeries LINUX may be attached to a multitude of physical interfaces when it resides standalone in an LPAR or a physical zSeries footprint. (See Visual 1.) In this case, the LINUX operating system and TCP/IP stack "own" and control the physical interfaces.
- zSeries LINUX may have dedicated physical interfaces of many types when it is operating as a Guest under z/VM. (See Visual 2.) In this case, the z/VM operating system has dedicated the physical interfaces to the LINUX operating system, which together with its TCP/IP stack "owns" and controls the physical interfaces.
- z/VM can support LINUX guests by connecting to them in any of several ways.
 - If using Pt-to-Pt or IUCV connections, the z/VM system can use PROXYARP to respond to requests for addresses that share the same IP subnet as the physical QDIO attachment of the z/VM system itself. This sometimes facilitates routing in the network, particularly when there is a dearth of addresses that can be handed out to the new LINUX guests. Alternatively, PROXYARP can be eliminated if a different IP subnet has been assigned to each of the IUCV and Pt-to-Pt interfaces. (See Visual 3.) In this case, the physical interfaces are owned and controlled by z/VM; the "virtual" Pt-to-Pt and IUCV interfaces are owned on one end by the LINUX guests and on the other end by z/VM.
 - Guest Virtual LAN support is the recommended way to attach Guest LINUX systems, as future support will be invested here versus with Pt-to-Pt or IUCV connections. (See Visual 4.) In this case, the physical interfaces are owned and controlled by z/VM; the "virtual" connections to the Guest LAN are owned on one end by the LINUX guests and on the other end by z/VM.
 - z/VM V4R2 supports Guest Virtual Lans that emulate Hipersockets connectivity. Because this is an emulation, the actual machine on which z/VM is running need not be a zSeries platform. HOWEVER, z/VM V4.2 Guest LANs cannot use broadcast or multicast mode. Therefore, at z/VM V4.2, MPROUTE cannot communicate with z/VM Guest LINUXes over the Guest LAN using OSPF or RIPv2 protocols. (Recall that OSPF and RIPv2 generally require multicast.)
 - NOTE: DRNEIGHBOR Coding could be employed to use OSPF over non-multicast interfaces, but this solution is not optimal in light of other, more elegant solutions for LINUX integration. Therefore, DRNEIGHBOR coding will not be addressed in this section.
- Starting with z/VM V4.3 Guest Virtual LANS may also be defined to emulate QDIO mode LANs and to use multicast and broadcast over Guest LANs. So, at this level of VM V4R3, MPROUTE with Multicast can be used over the Guest LAN interfaces.
- At z/VM V4.4 more capability has been added to Guest LAN support. One new feature in z/VM V4.4 is the use of the Virtual Switch. You will see this in a later visual.
- An OSA-E on a zSeries can support 80 LINUX Guests that are directly attached (3 UCB addresses per LINUX Guest). An OSA-E can support 60 z/OS Guests that are directly attached to the OSA-E (4 UCB addresses per attachment).
- An OSA-E on a G5/G6 platform supports far fewer z/VM Guests that are directly attached (perhaps 6 z/VM Guests).

z/VM as ASBR for LINUX Guests

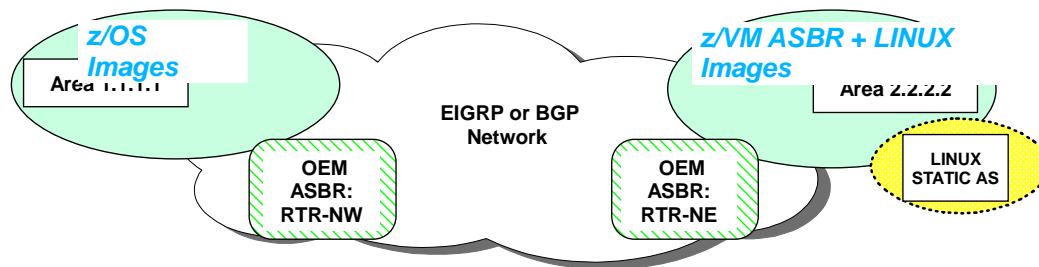
- 2 Autonomous Systems
- Static Routing for LINUX
- Non-Backbone Area Configuration
- HiperSockets Between VM and z/OS
- LANs and HiperSockets Between VM and z/OS
 - Reach LINUX via External Routes to 10.1.1.0 over OSA-E or HiperSockets



© IBM Corporation 2001, 2002, 2003

1. Note in our diagram how the z/OS images and z/VM itself are attached to two different OSA-Es, which are presumably attached to the rest of the network in such a way as to provide redundancy. z/VM and z/OS are also attached to each other via HiperSockets.
2. Please imagine that the ABRs in the network provide redundancy to reach any of the depicted LANs. (There were too many lines to draw, so we appeal to your imagination for this.)
3. This diagram depicts zSeries LINUX guests under z/VM. z/VM is acting as a router to the guests. It communicates with the LINUX guests over a Guest LAN using address 10.1.1.1.
 1. If using Guest Virtual LAN support, the z/VM system cannot use Proxy Arp. Remember that External Routes (LSA Type 5) are not sent into Stub Areas; this is why our configuration has z/VM and z/OS in a non-backbone area that is allowed to receive LSA Type 5s.
 2. Since z/VM itself can support the OSPF dynamic routing protocol via the MPROUTE code (ported from z/OS Communications Server's OMPROUTE), we can configure z/VM as a node in the OSPF AS. (z/VM supports MPROUTE starting with z/VM V4R2.) And yet, note how z/VM is communicating with the LINUX guests using STATIC routing protocols. Since z/VM must be able to advertise the static routes to the rest of the Non-Backbone Area 1.1.1.1, including to the z/OS images, as well as to the ABRs that interconnect Area 1.1.1.1 with Area 0.0.0.0, z/VM is configured as an Autonomous System Boundary Router (ASBR).
 3. Note how the z/VM LPAR has still been designated the PRIROUTER for the shared OSAs; otherwise the packets destined for the LINUX guests in subnet 10.1.1.0/24 would be rejected. (OSAs must either recognize the full host address of received packets or be allowed to recognize unknown host addresses in order to pass the packets up to the IP stack for further routing.)
 4. This solution is all right, but ... depending on the size of the Autonomous System and the number of other types of AS's, the routing tables could be very large. It might be better to come up with another network design that minimizes the routing table at each node and the amount of computation required to build a routing table.
4. So, in summary, our configuration shows:
 1. Two Autonomous Systems
 1. OSPF + Static Routes
 2. Area 1.1.1.1 (a non-backbone area able to receive LSAs for external routes)
 1. z/VM = ASBR
 2. z/OS = Intra-Area Routers
 3. Areas 1.1.1.1 and 0.0.0.0
 1. ABRnn = Area Border Routers

Separating z/VM + LINUX from z/OS



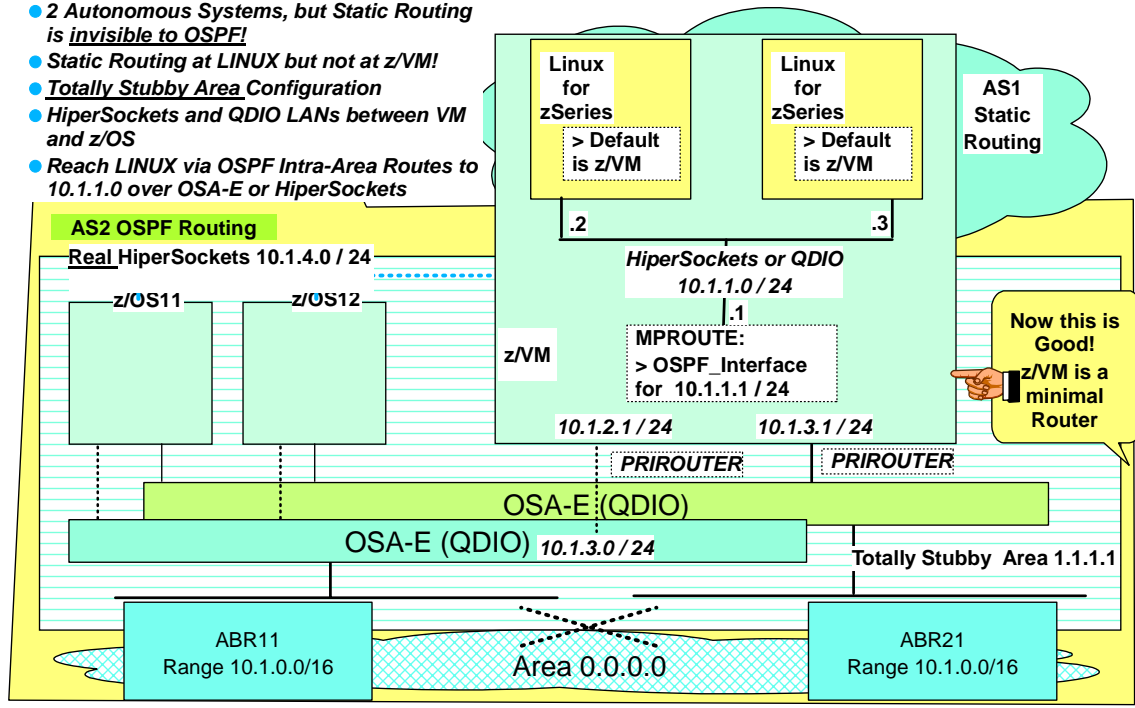
- **OSPF areas may be attached as non-backbone to an Autonomous System of another protocol (e.g., EIGRP or BGP)**
 - **All routing between OSPF areas occurs through ASBRs and the intermediate AS.**
 - **Permits strategy to connect another AS to any of the OSPF areas.**
 - **For example, LINUX Static AS**
 - **Minimizes Routing Table size at z/OS, z/VM, and LINUX.**

© IBM Corporation 2001, 2002, 2003

1. In this configuration we have z/VM with LINUX in one Area and z/OS in another. We have also placed each area in a separate OSPF Autonomous System (AS).
 2. The non-zSeries ASBRs will be configured to ship only default routes into the OSPF areas.
 3. Since there will be relatively small routing trees that need to be maintained within the OSPF areas, a Stub Area or Totally Stubby configuration in the z/OS area is not necessary. This allows the attachment of a Static Autonomous System for LINUX Guests to the z/VM OSPF areas because any router in them may be an ASBR. It also allows the z/VM system to be an ASBR in its own right (as you saw in the previous visual) and to interface with the non-zSeries ASBR, which can perform extra filtering to reduce the size of the routing tables at z/OS, z/VM, and LINUX.
1. Remember: ASBRs cannot import routing destinations into OSPF Stub Areas.

LINUX Guests: z/VM & z/OS in Stub Area

- 2 Autonomous Systems, but Static Routing is *invisible to OSPF!*
- Static Routing at LINUX but not at z/VM!
- Totally Stubby Area Configuration
- HiperSockets and QDIO LANs between VM and z/OS
- Reach LINUX via OSPF Intra-Area Routes to 10.1.1.0 over OSA-E or HiperSockets

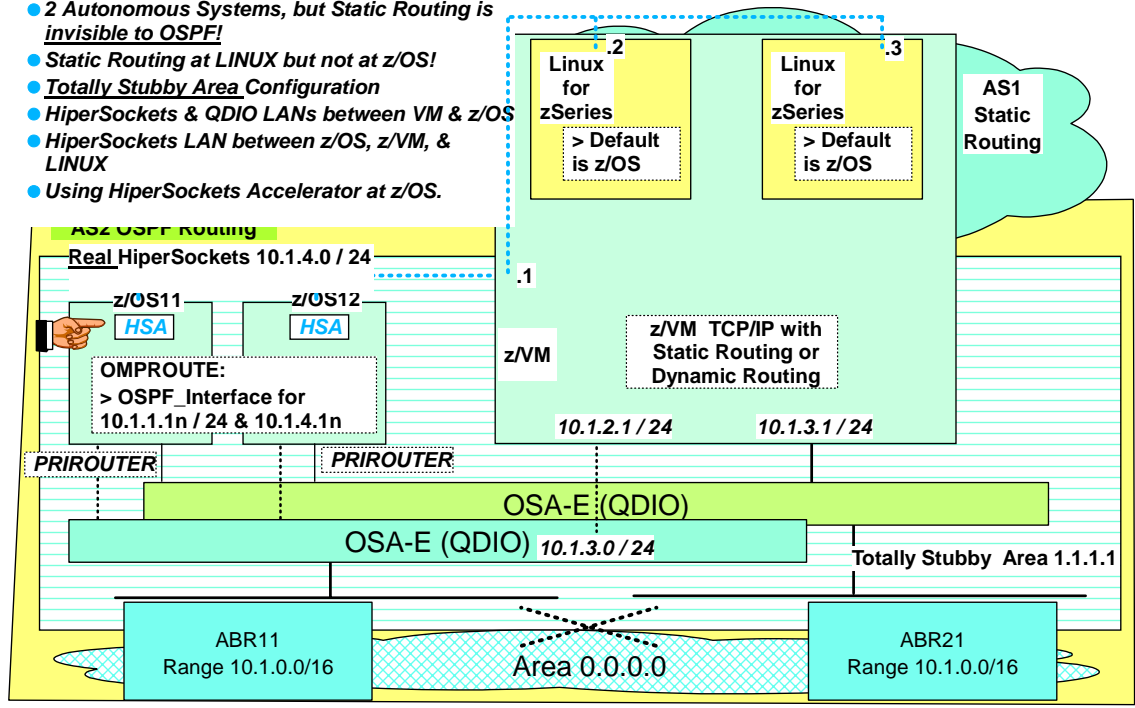


© IBM Corporation 2001, 2002, 2003

1. Note in our diagram how all of the z/OS images and z/VM itself are attached to two different OSA-Es, which are presumably attached to the rest of the network in such a way as to provide redundancy. z/VM and z/OS are also attached to each other via HiperSockets.
2. Again, please imagine that the ABRs in the network provide redundancy to reach any of the depicted LANs. (There were too many lines to draw, so we appeal to your imagination for this.)
3. This diagram depicts the zSeries LINUX guests under z/VM. Again, z/VM is acting as a router to the guests - however its routing duties are not heavy. It communicates with the guests over a Guest LAN. The advantage to this configuration is that z/VM and the z/OS images reside in a Totally Stubby Area, communicating with each other via Intra-Area Routes and with Default Routes to reach the backbone area and any other areas or autonomous systems beyond the two Area Border Routers (ABRs).
4. The LINUX guests are using static routing that points to the z/VM router as the default router over the HiperSockets or QDIO Guest LAN. However, the z/VM attachment to the Guest LAN is defined as an OSPF_Interface so that the z/VM node can advertise the Guest LAN IP subnet address throughout the Totally Stubby Area 1.1.1.1 and to the ABRs (ABR11 and ABR21). Although the z/VM interface to the Guest LAN sends out periodic HELLOs, the LINUX Guests will not respond since they have not joined the appropriate OSPF multicast group. In fact, even if z/VM is at a level that does not support multicast over the Guest LAN, thus preventing the OSPF_Interface from even receiving HELLO responses (were they possible), the MPROUTE code will still send out LSAs about the subnetwork 10.1.1.0 over the HiperSockets and the QDIO interfaces.
5. Note how the z/VM LPAR has still been designated the PRIROUTER for the shared OSAs; otherwise the packets destined for the LINUX guests in subnet 10.1.1.0/24 would be rejected. (OSAs must either recognize the full host address of received packets or be allowed to recognize unknown host addresses in order to pass the packets up to the IP stack for further routing.)
6. So, in summary, our configuration shows:
 1. Two Autonomous Systems
 1. OSPF + Static Routes (an AS hidden from OSPF)
 2. Area 1.1.1.1 (a Totally Stubby Area)
 1. z/VM and z/OS communicate with each other via intra-area routes
 2. z/VM and z/OS communicate with the rest of the network by using the Default routes sent to them by the ABRs (ABR11 and ABR21), which have connections to the Totally Stubby Area and to Area 0.
 3. Area 0.0.0.0 (backbone area)
 1. Used to reach other OSPF Areas in the same Autonomous System or to reach other ASs that are not depicted.

LINUX Using z/OS as a Router in Stub

- 2 Autonomous Systems, but Static Routing is invisible to OSPF!
- Static Routing at LINUX but not at z/OS!
- Totally Stubby Area Configuration
- HiperSockets & QDIO LANs between VM & z/OS
- HiperSockets LAN between z/OS, z/VM, & LINUX
- Using HiperSockets Accelerator at z/OS.



© IBM Corporation 2001, 2002, 2003

1. This diagram is quite similar to the previous one -- with a major exception: z/OS is acting as the "minimal" router for the z/LINUX images instead of z/VM, as was the case in the previous diagram. Of course, there are a few other differences as well.
2. This diagram depicts the zSeries LINUX guests under z/VM. However, in this scenario, z/VM is NOT A ROUTER. z/OS takes over the ROUTING role and uses HiperSockets Accelerator function to reduce the CPU requirements for z/OS. z/OS communicates with the guests and with z/VM over a HiperSockets connection -- we are not using a Guest LAN in this configuration, since each LINUX owns a connection to the HiperSockets LAN as does z/VM. We still have the Stubby Area we had before, but now z/OS can be designated the PRIROUTER for the shared OSAs.
 1. Note: The use of "HSA" at the z/OS nodes to indicate "HiperSockets Accelerator" can cause confusion because HSA" also stands for the "Hardware System Area." We use "HSA" here only because of space limitations.
3. Note in our diagram how all of the z/OS images and z/VM itself are attached to two different OSA-Es, which are presumably attached to the rest of the network in such a way as to provide redundancy. z/VM and z/OS are also attached to each other via HiperSockets.
4. Again, please imagine that the ABRs in the network provide redundancy to reach any of the depicted LANs. (There were too many lines to draw, so we appeal to your imagination for this.)
5. If a client in the network beyond the ABRs need to reach z/VM, he can do so by means of the OSAs, since z/VM's TCP/IP will download its own addresses for the HiperSockets connection and the QDIO attachments into the OSAs.
6. You might still configure z/VM with a TCP/IP stack that is using MPROUTE, as you saw in the previous diagram. In such a case, z/VM s part of the Totally Stubby Area, but it can advertise connections to the HiperSockets and QDIO LANs. Alternatively, you can choose to use static routing at z/VM so as to essentially eliminate dynamic routing overhead completely.
7. Of course, if static routing is being used at z/VM you have the option of making the ABRs into ASBRs and allowing them to import the static host routes at z/VM and advertise them into the backbone. However, this is probably not necessary since the subnet routes are already being advertised into the backbone because of the OSPF protocols at z/OS. Once the network packets reach the OSA LANs, the ARP process will ensure that z/VM is reached over the OSA attachments. The z/LINUX guests are reached over the z/OS images that are using the HiperSockets Accelerator function and acting as minimal routers themselves.

Extra Pages on New z/VM V4.4 Virtual Switch

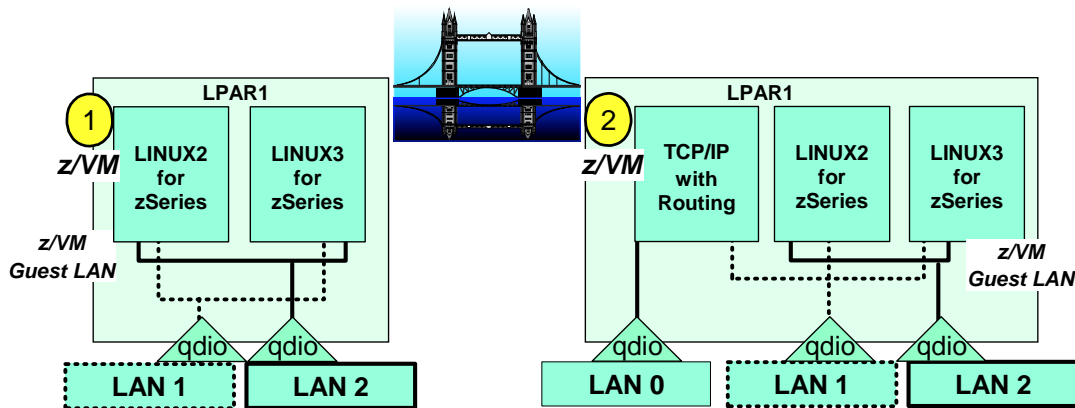


Depending on time we may treat
this short section as an
"Appendix" and skip to Page 33!



© IBM Corporation 2001, 2002, 2003

Physical Network Design for VSWITCH

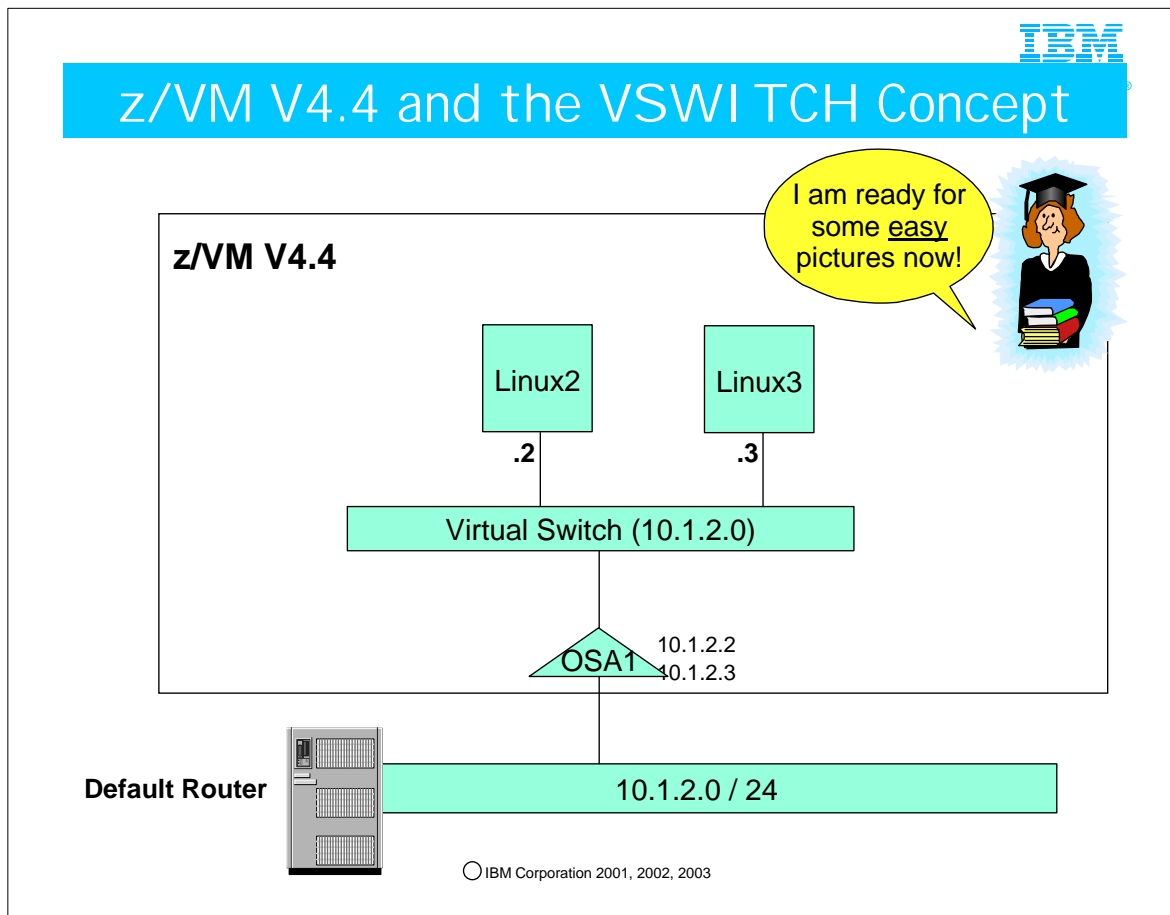


- Virtual Switch ("VSWITCH") in z/VM V4.4
 - VM Control Program bridges to the Adapter
 - Guest Machines are on same subnet as QDIO Adapter of Guest LAN.
 1. Direct Attachment to Physical QDIO Interface
 - Under z/VM with no internal layer 3 routing machine required
 2. Direct Attachment to Physical QDIO Interface
 - One Guest Machine acts as a Layer 3 Router

© IBM Corporation 2001, 2002, 2003

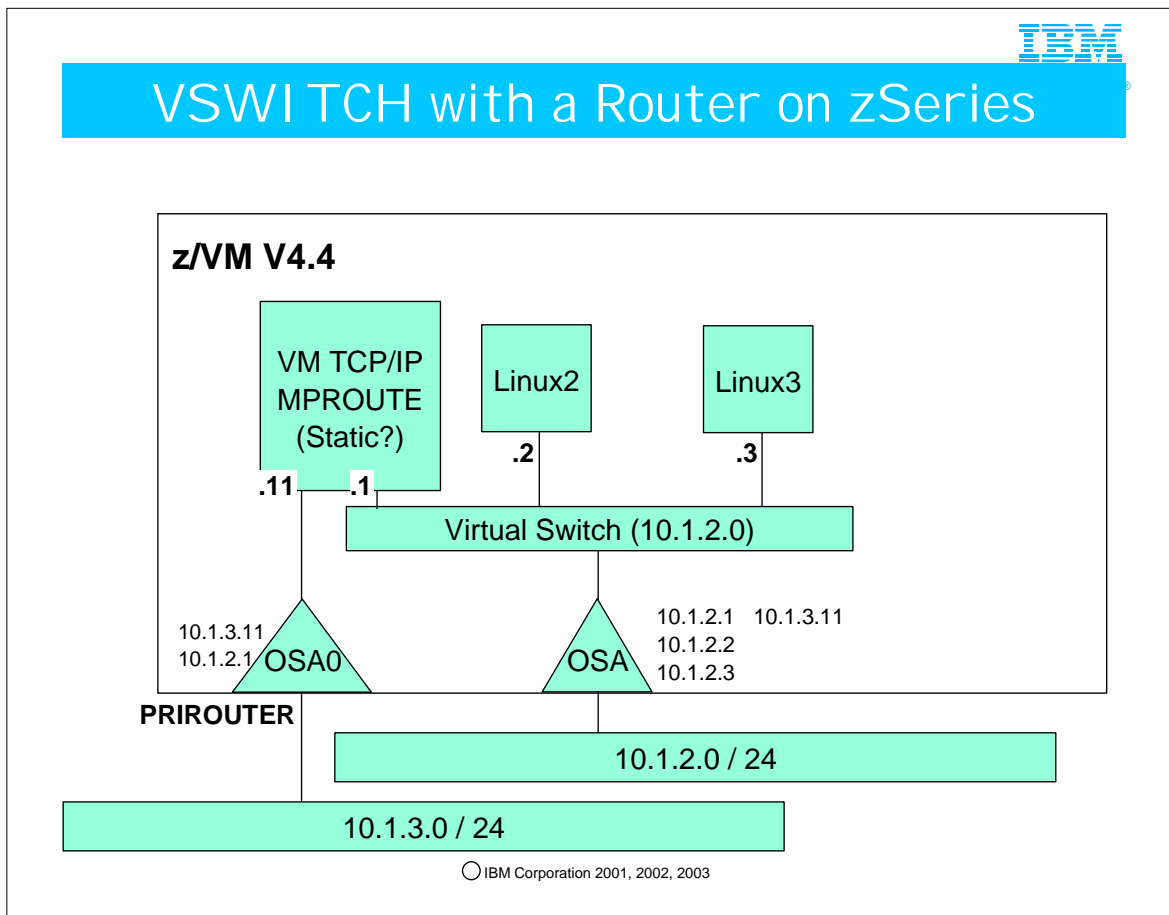
1. The "Virtual Switch" (or "VSwitch"), a layer-2 routing technique, was announced together with other new functions for z/VM V4.4 in Announcement Letter 203-128. The Announcement Letter, dated May 13, 2003 with availability August 15, 2003, is entitled:
 1. "IBM z/VM V4.4 Improves Virtualization Capabilities for Linux on zSeries"
2. What is a "Virtual Switch"?
 1. In a nutshell, the Virtual Switch allows you to connect guests over a Guest LAN that resides in the same IP subnet as the real QDIO adapter. (HiperSockets Guest LANs cannot take advantage of the Virtual Switch configuration.) VM's Control Program (CP) bridges the Guest LAN to the real QDIO LAN. The z/VM Control Program and the associated Virtual Switch controller (i.e., a copy of the z/VM TCP/IP stack itself) push the Guest LAN addresses into the QDIO adapter. Which addresses are pushed down is a function of the operating system that the VM Guest is using. z/VM and z/OS Guests are similar to each other in this respect; zSeries LINUX is different. We will look briefly at these differences on a later set of visuals.
 2. The z/VM Virtual Switch employs transparent bridging to enable the switch to dynamically determine and maintain node connectivity so that the LAN administrator has less network maintenance to perform. The z/VM Virtual Switch also adds 802.1q VLAN support, although a discussion of this particular feature goes beyond the scope of this presentation.

z/VM V4.4 and the VSWITCH Concept



1. We will get back to our look at LINUX in an OSPF network later. But for now, we need to look at some simple pictures to understand the Virtual Switch.
2. Here you see two LINUX Guests using the z/VM V4.4 feature called "Virtual Switch."
3. The virtual switch allows transparent bridging to the OSA adapter (and also introduces VLAN 802.1q capability to the VSWITCH connections).
4. Note how the addresses on the VSWITCH belong to the same IP Subnet as the LAN Segment to which the OSA is attached.
5. Note how the all the addresses known to the VSWITCH are downloaded into the OSA adapter. Thus, any packets from the network that are destined for addresses known to the OSA Adapter will be forwarded to those addresses; the OSA will not discard packets to known IP destination addresses on the zSeries platform.
6. Since the Guest Machine images appear attached to the OSA itself, these images can bypass the z/VM routing function and connect directly to an external network over a QDIO OSA-E. Any other node besides z/VM -- a downstream router, for example -- can function as the default router for the Guest Machine, thus reducing the CPU load on z/VM. The Guest Machine (e.g., LINUX, z/OS, z/VM) configuration with VSWITCH looks like a flat network, since there is no z/VM router in between the Guest and the external network. Every node serviced by the virtual switch has an adapter coupled directly to the virtual switch's LAN.
7. For more information about Virtual Switch, see www.vm.ibm.com. Here are some excerpts from the z/VM V4.4 Announcement Letter 203-128, "IBM z/VM V4.4 Improves Virtualization Capabilities for Linux on zSeries". It is here for your reference only. :
8. "Network Consolidation using the Virtual Switch: z/VM V4.4 further enhances virtualization technology by introducing a virtual IP switch that is capable of bridging a z/VM Guest LAN to an associated real LAN connected by an OSA-Express adapter. The z/VM virtual switch is designed to help eliminate the need for virtual machines acting as routers to provide IPv4 connectivity to a physical LAN through an OSA-Express adapter. Further, it helps eliminate the need to define a separate routable subnet for the exclusive use of the members of a Guest LAN. Using the virtual switch, the convenience of a Guest LAN is maintained while allowing the guests to be assigned IP addresses in the real LAN subnet."
9. "Virtual routers consume valuable processor cycles to process incoming and outgoing packets requiring additional copying of the data being transported. The virtual switch helps alleviate this problem by moving the data directly between the real network adapter and the target or originating guest data buffers."
10. "Centralized network configuration and control of the virtual switch within CP helps allow the z/VM Guest LAN administrator to more easily grant and revoke access to the real network and to manage the configuration of Guest LAN VLAN segments. While the z/VM system can be a member of multiple VLANs, the z/VM Guest LAN administrator can control which guests belong to which real VLAN, without requiring additional network adapters or switch port configuration. If a guest does not support IEEE 802.1q, z/VM will transparently join the virtual network interface into the desired VLAN."

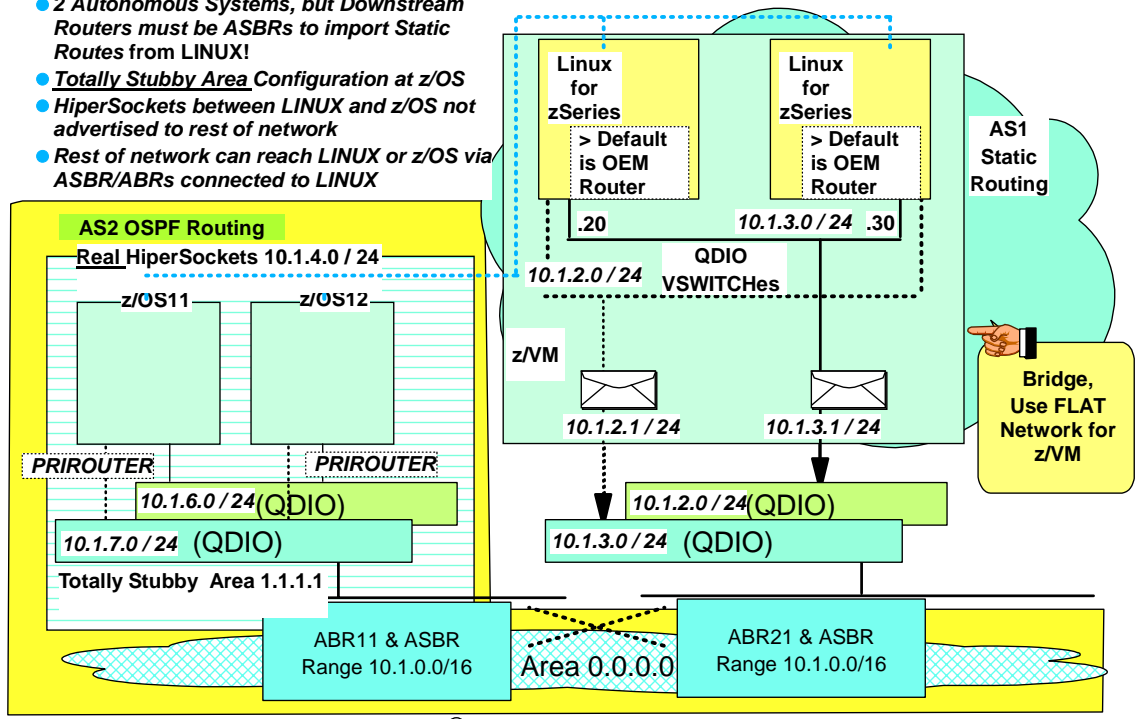
VSWITCH with a Router on zSeries



- Here we have added a z/VM TCP/IP machine attached both to an OSA-E in its own right and also to the VSWITCH at 10.1.2.0. The TCP/IP stack could be using static routing or dynamic routing. The advantage to using any kind of routing stack in front of network adapters is that you may use multipath outbound. The VSWITCH direct bridging cannot perform multipath over multiple OSA adapters; it can perform multipath only over the Guest LAN if the guest has two interfaces to the guest LAN. Note which addresses have been downloaded into OSA0:
 - z/VM's TCP/IP sends all addresses known to itself down to the attached OSA: 10.1.3.11 and 10.1.2.1.
 - No LINUX machine has direct connectivity to the OSA0, so none of the LINUX addresses are downloaded onto that QDIO adapter.
- Note which addresses have been downloaded into OSA1:
 - z/VM's TCP/IP sends all addresses known to itself down to the attached VSWITCH: 10.1.3.11 and 10.1.2.1.
 - LINUX2 and LINUX3 send their addresses on LAN Segment 10.1.2.0 to OSA1: 10.1.2.2 and 10.1.2.3.
- Packets coming in over LAN Segment 10.1.2.0 will reach the LINUX Guests directly, as the OSA knows and accepts these destination addresses, since they have been downloaded as previously described. The packets could even reach the z/VM TCP/IP stack at address 10.1.2.1; again, this address is known to the OSA card.
- Packets coming in over LAN Segment 10.1.3.0 will reach the z/VM TCP/IP stack at address 10.1.3.11. Such packets could be routed out of z/VM TCP/IP over the VSWITCH into the LINUX Guests if z/VM were made PRIROUTER on OSA0.
- In case of a complete VSWITCH failure, Linux can still get out through VM TCP/IP because the Guest LAN portion of the Virtual Switch continues to operate. Likewise and in the reverse direction, packets needing to reach the LINUX guests can use z/VM's TCP/IP as a router. Of course, z/VM must be the PRIROUTER for the OSA so that packets to destinations unknown to OSA0 (i.e., LINUX2 and LINUX3) may be accepted by the OSA0 and forwarded through z/VM TCP/IP using z/VM's routing protocol.
 - Note: Although our diagram depicts z/VM as the router for this configuration, in fact, z/OS could serve this purpose.
- Just in case you were wondering if you could attach both the z/VM TCP/IP stack to the same OSA as the one to which the VSWITCH is attached, the answer is "yes."
 - You might even add a second OSA and a second VSWITCH to which VM TCP/IP and a different set of Guest LANs are attached.
 - For more information about these types of configurations, please consult the literature on designing with VSWITCHes under z/VM V4.4.

Linux & VSwitch: 2 Subnets, Separate OSAs

- 2 Autonomous Systems, but Downstream Routers must be ASBRs to import Static Routes from LINUX!
- Totally Stubby Area Configuration at z/OS
- HiperSockets between LINUX and z/OS not advertised to rest of network
- Rest of network can reach LINUX or z/OS via ASBR/ABRs connected to LINUX



© IBM Corporation 2001, 2002, 2003

1. Here you see a configuration with z/VM and LINUX residing in a STATIC Autonomous System. The Virtual Switch creates a "Flat Network" look with no router between the VM Guests and the OSA QDIO Adapters themselves. Note how z/VM does not own any IP address on the Guest LAN because it will not be functioning as a router.
2. Again, please imagine that the ABRs/ASBRs in the network provide redundancy to reach any of the depicted LANs. (There were too many lines to draw, so we appeal once again to your imagination for this.)
3. Note that our visual now shows the Virtual Switch bridging directly to the adapter. Also note how we have reconfigured our QDIO adapters, with two attached to z/VM and two different QDIO adapters attached to the MVS images. Finally, note how HiperSockets connections are defined via static routing in the LINUX images and in z/OS. Since these connections will be used ONLY for z/OS-LINUX communications, there is no need for these static routes to be imported into OSPF for broadcasting through the OSPF AS; as a result, our z/OS images may remain in a Totally Stubby Area.
4. The LINUX guests point not to z/VM as their default router, but to some external OEM router. In our visual the router might be ASBR11 or ASBR21, depending on the physical connectivity.
5. With a FLAT Network design you may lose the traffic splitting afforded by dynamic routing over the OSAs to the LINUX guests unless you manage traffic distribution inbound by some means other than dynamic routing with OSPF. Of course, the static routing protocols at the downstream router might allow you to entertain two equivalent paths. Outbound the LINUX Guests may use multipath only on the Guest LAN itself if they have multiple interfaces to it; however, CP does not support Multipath on the physical adapters -- even if they reside on the same subnet.



5. Common Problems

© IBM Corporation 2001, 2002, 2003

Common Errors at V2R6 or Higher

```

S OMPROUTE
$HASP100 OMPROUTE ON STCINRDR
IEF695I START OMPROUTE WITH JOBNAME OMPROUTE IS ASSIGNED TO
USER TCPI1A, GROUP OMVSRP
$HASP373 OMPROUTE STARTED
JOBNAME  PROCSTEP  STEPNAME  CPU TIME      EXCPS      RC
OROUTED  STARTING  OMPROUTE  00:00:00      46         00
EZZ7800I OMPROUTE STARTING
IEE252I MEMBER CTIORA00 FOUND IN SYS1.PARMLIB

ICH408I USER(TCPI1A ) GROUP(OMVSRP )
NAME(#####)
  MVS.ROUTEMGR.OMPROUTE CL(OPERCMDS)
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(CONTROL) ACCESS ALLOWED(NONE )
EZZ7897I USER IS NOT RACF AUTHORIZED TO START OMPROUTE
EZZ7805I OMPROUTE EXITING ABNORMALLY - RC(11)
OMPROUTE --NONE-- *OMVSEX  00:00:00      521      11
$HASP395 OMPROUTE ENDED

```

© IBM Corporation 2001, 2002, 2003

1. OMPROUTE requires RACF authorization:
 1. The User associated with the task must have been authorized to start OMPROUTE.
 2. Although not required, this user may be a Superuser.
2. The error indicates that you need to authorize the userid that starts OMPROUTE with a PERMIT statement:
 1. RDEFINE OPERCMDS(MVS.ROUTEMGR.OMPROUTE) UACC(NONE)
 2. PERMIT MVS.ROUTEMGR.OMPROUTE ACCESS(CONTROL) - CLASS(OPERCMDS) ID(TCPI1A)
 3. SETROPTS RACLIST(OPERCMDS) REFRESH

RACF Worksheet + Definitions

Superusers (Started Tasks):	<pre>ADDUSER TCPIP1A DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/ ')) SETROPTS CLASSACT(STARTED) RACLIST(STARTED) RDEF STARTED TCPIP.* STDATA(USER(TCPIP1A)) RDEF STARTED OMPROUTE.* STDATA(USER(TCPIP1A)) SETROPTS RACLIST(STARTED) REFRESH</pre>
OMPROUTE Initialization	<pre>RDEFINE OPERCMDS(MVS.ROUTE MGR.OMPROUTE) UACC(NONE) PERMIT MVS.ROUTE MGR.OMPROUTE ACCESS(CONTROL) - CLASS(OPERCMDS) ID(TCPIP1A) SETROPTS RACLIST(OPERCMDS) REFRESH</pre>

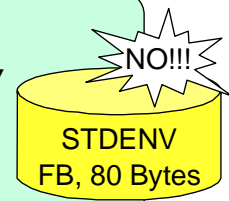
Must reside in APF-authorized library.

© IBM Corporation 2001, 2002, 2003

1. OMPROUTE requires association with a Superuser just as the TCPIP cataloged procedure does.
2. In addition, even this superuser must be permitted to the operator command "MVS.ROUTE MGR.OMPROUTE."
 1. OROUTED at V2R6 also requires special command authorization for "MVS.ROUTE MGR.OROURED."
 1. At V2R5 this additional authorization for OROUTED was unnecessary.

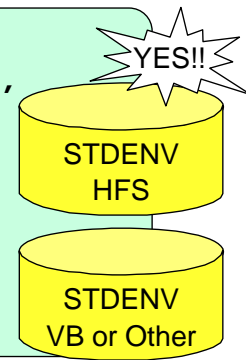
Something Wrong with STDENV

```
//NM1AOMPR PROC
//NM1AOMPR EXEC PGM=OMPROUTE,REGION=0K,TIME=NOLIMIT,
// PARM=( 'POSIX(ON)' ,
//      'ENVAR( "_CEE_ENVFILE=DD:STDENV" , ' ,
//      '"_CEE_RUNOPTS=HEAP(,,,FREE)" ' ,
//      '/ -t1' )
//STDENV DD DSN=SYS1.TCPIP.TCPPARMS(NM1AOENV),DISP=SHR
.....
```



Make sure your STDENV file is not in a fixed-block dataset

```
//NM1AOMPR PROC
//NM1AOMPR EXEC PGM=OMPROUTE,REGION=0K,TIME=NOLIMIT,
// PARM=( 'POSIX(ON)' ,
//      'ENVAR( "_CEE_ENVFILE=DD:STDENV" , ' ,
//      '"_CEE_RUNOPTS=HEAP(,,,FREE)" ' ,
//      '/ -t1' )
//STDENV DD PATH='/etc/omproute.env.nm1a' ,
//      PATHOPTS=(ORDONLY)
//*STDENV DD DSN=SYS1.TCPIP.OMPRENV,DISP=SHR
.....
```



© IBM Corporation 2001, 2002, 2003

1. OMPROUTE is a UNIX application.
 1. UNIX pads fixed-block file lines with blanks to the fixed-block size
 2. In UNIX, blanks are valid filename characters.
2. You are always safer allocating your STDENV file as either an HFS file or as a non-Fixed Block file, unless the files you reference inside the OMPROUTE Environment file (STDENV) are MVS datasets. The next page explains more.
3. Although we don't spend time on this topic in this brief pitch, note how our JCL includes some LE environment variables to free LE storage. In some cases it is necessary to free this storage. The symptoms might be that there are Neighbor state loops, with OMPROUTE having deleted most of the dynamic routes and taking many minutes to recover. Another symptom might be significant growth in OMPROUTE storage usage. (Several other problems might also occur; these are detailed in the RETAIN Informational APAR II12026, "INFORMATIONAL APAR FOR OMPROUTE PROBLEMS.") One of these scenarios is caused by LE Runtime options that indicate storage should be kept rather than freed. The LE default setting for HEAP management is:


```
HEAP((32K,32K,ANYWHERE,KEEP,8K,4K),OVR)
```

 1. This statement allocates 32K heaps and does not free the heap (HANC control block) after it is no longer needed.
 2. To prevent storage growths, the following setting is recommended for OMPROUTE:
 1. HEAP(,,,FREE)
 3. This parameter can be specified on the JCL PARMS as indicated above.
4. NOTE: FREE HEAP is recommended when there are many transient routes (routes that are only temporary, or where omproute has lots of changes in neighboring routers, or when tracing is enabled for a long period of time, etc. It does not necessarily hurt to specify the FREE HEAP option. However, be aware of the disadvantage when trying to diagnose an overlay problem: there is an extremely remote possibility that the storage causing the overlay problem might be freed, making it difficult to diagnose the problem. In such a case you would want to eliminate this LE environment variable before recreating the problem.

Unable to Find Config File or Read Trace!

```

0          10          20          30          40          80
+.....+.....+.....+.....+.....+.../ /...+
STDENV
FIXED 80
OMPROUTE_FILE=/etc/omproute.conf
OMPROUTE_DEBUG_FILE=/tmp/omproute.dbg
  
```



```

•EZZ7822 Could not find configuration file
  - or -
•Trailing blanks in directory names or
  filenames are not supported by edit or
  browse
  - or -
•EDC5129I No such file or directory.
  
```



OMPROUTE is a UNIX application.
 UNIX pads fixed-block file lines with blanks to the fixed-block size
In UNIX, blanks are valid filename characters.

So in this example OMPROUTE would be looking for files named:

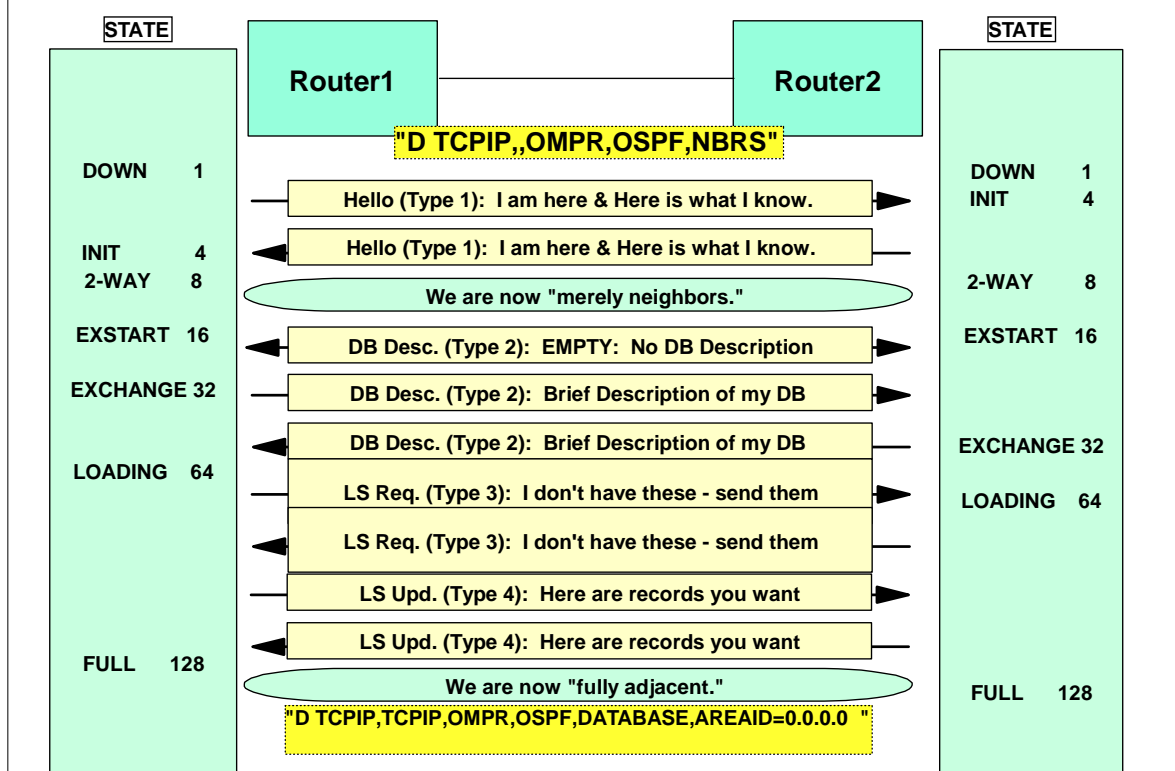
```

"/etc/omproute.conf      "
"/tmp/omproute.dbg      "
  
```

Which probably isn't what you actually named them!

1. If you use a STDENV file and get this error on OMPROUTE startup:
 1. EZZ7822 Could not find configuration file
 2. This means that your configuration file is in the HFS (i.e., it is a UNIX file). UNIX pads fixed-block file lines with blanks to the fixed-block size and thus converts your configuration file name into a name that is padded with blanks. Since the actual configuration file name is not padded with blanks, the file cannot be found!
 3. BEST SOLUTION: Make sure your STDENV file is not in a fixed-block dataset: either a Variable-Blocked or Sequential Dataset or an HFS File.
 4. ALTERNATE SOLUTION #1: If STDENV must remain FB, put your configuration file into an MVS Dataset.
2. If you use a STDENV file and get any of the following errors when you try to read or browse your trace data from the HFS:
 1. Trailing blanks in directory names or filenames are not supported by edit or browse
 2. cat: /var/logs/omproute.debug.nm1a: EDC5129I No such file or directory.
 3. Erno=81x No such file or directory exists; Reason=05620062x
 1. This means that your OMPROUTE_DEBUG_FILE is coded inside an OMPROUTE Standard Environment File that resides in a Fixed Block Dataset in MVS and it is pointing to an HFS file. UNIX dynamically creates the file in the HFS. As explained previously, UNIX pads fixed-block file lines with blanks to the fixed-block size and thus converts your debug trace file into a name that is padded with blanks.
 4. BEST SOLUTION: Make sure your STDENV file is not in a fixed-block dataset: either a Variable-Blocked or Sequential Dataset or an HFS File.
 5. ALTERNATE SOLUTION #1: If STDENV must remain FB, copy the HFS file into an MVS Dataset and browse from there. (This is supported.)

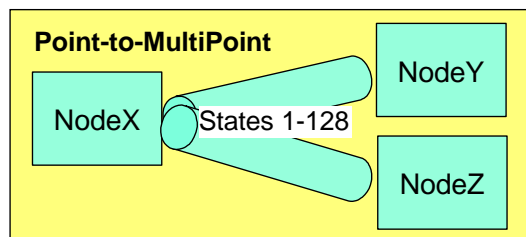
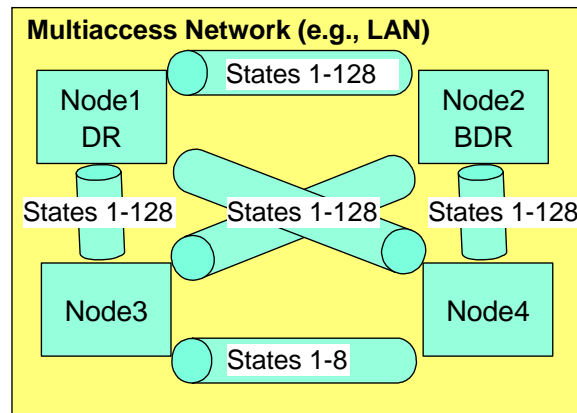
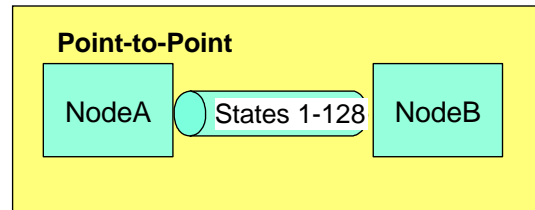
Review: Neighbor States



1. This page reviews what you have already learned if you attended the session on OSPF protocols ("An OSPF Tutorial").
2. We show it to you again to set the stage for examining neighbor adjacency issues in the messages and traces you may be taking.
3. Review: The diagram indicates the state changes that the routers (or CS nodes) go through during each phase of the protocol exchange.
4. Depending on the platform, the command displays for monitoring the routing in the network might show either the verbal status ("DOWN," INIT," etc.) or a number that can be equated with that state (DOWN=1, INIT=4, and so on).
5. 1 (DOWN): Initially the routers are down and have no contact with each other.
6. 2 (Attempt): In NonBroadcast MultiAccess (NBMA) networks a different state may be indicated even when a router is marked down. "Attempt" indicates that no contact has been made but the Hello packet will continue to send the packets to "attempt" to make contact.
7. 4 (Initialize): The Hello packet has been identified/received but no exchange of information has taken place between neighboring routers. Contact between the neighboring routers has been made, however.
8. 8 (2-Way): The Hello packets have been received and acknowledged by both neighboring routers, as indicated by the presence of the router itself in the neighbor's Hello packet. That is, each router sees itself in the neighbor's Hello packet. The designated router (DR) is selected and thereafter follows the selection of the Backup DR.
9. 16 (ExStart): Neighbor routers form adjacencies between themselves. Neighbor routers' communication is more advanced in this state and the routers decide who is the "master" and who is the "slave" and what is the initial DataBase Sequence Number. The transmission of the link state database can begin.
10. 32 (Exchange): The neighbors send their link state database to their adjacent routers. The link state database records describes the characteristics of the database and each must be acknowledged. Link state request packets requesting the neighbor's recent LSA's status may be sent to the neighboring routers. In this state the neighbors are capable of receiving and sending all types of OSPF routing protocol packets.
11. 64 (Loading): The link state request packets are being transmitted and received from neighboring routers requesting the most recent LSAs.
12. 128 (Full): If the neighboring routers are displayed with this state, it means that they are fully adjacent. | The adjacent routers exchange LSAs and appear in their neighbors' router-LSAs.
13. The synchronization of databases can be verified with a command to display the OSPF database; there is a field called CHECKSUM TOTAL, which can be used to compare if 2 routers have synched their DBs.
 1. D TCPIP,TCPIP,OMPR,OSPF,DATABASE,AREAID=0.0.0.0

Neighbor States

Nbr. State #	Meaning
1	Down
2	Attempt
4	Init
8	2-way
16	ExStart
32	Exchange
64	Loading
128	Full



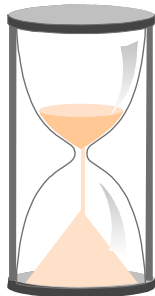
© IBM Corporation 2001, 2002, 2003

1. The Neighbor State Codes are described in RFC 1583.
2. A multiaccess network is a network in which all the attached hosts share one transport medium. Examples of multiaccess networks are LANs, ATM, etc.
3. Although you will frequently hear that the best neighbor state to be in is State 128 (Full Adjacency), in fact, on a Multiaccess network the only Full Adjacency is with the DR and the BDR. In the broadcast network depicted, Node3 and Node4 are perfectly fine with Neighbor State of 8 (2-way communication), as they need not synchronize their databases with each other. They synchronize only with the DR and the BDR. They simply need to know that they are available for communication with each other in the subnetwork.

Performance Problems & Timing

OSPF is a timer-intensive protocol

APAR II12026, "INFORMATIONAL APAR FOR OMPROUTE PROBLEMS."



→ *If dead router intervals are allowed to lapse without OMPROUTE sending and processing packets from neighbors, network routing problems can ensue*

× *adjacencies dropped*

★ *snowballing bad performance as OMPROUTE and neighbors attempt to restore lost adjacencies*

× *routes lost*

APAR PQ51196 (V1R2) added a warning to impending problems:

EZZ7968I Hello interval missed on interface OSAQDIO

© IBM Corporation 2001, 2002, 2003

1. Many performance and other problems are detailed in the RETAIN Informational APAR II12026, "INFORMATIONAL APAR FOR OMPROUTE PROBLEMS.") We include some of these hints and tips in this presentation, but you will want to consult this APAR frequently for information we have not covered and for any new information.
2. Hello and dead router intervals are a tradeoff. Higher values mean slower convergence of router outages. Lower values mean that adjacencies are quicker to be dropped if there is a temporary holdup in packets
3. It takes a lot more processing to restore lost adjacencies that it does to keep them up in the first place. So if you are losing adjacencies due to performance problems, the problem will only get worse as attempts are made to restore them
4. When adjacencies are dropped, a router deletes from its routing table all routes learned from the dropped neighbor and must relearn them either from another source or by re-establishing the lost adjacency. So network connectivity can be severely affected.
5. Message EZZ7968 is issued if OMPROUTE notices that it has not been able to send a required hello packet within two hello intervals. This indicates that the adjacency is in trouble, probably because OMPROUTE is not getting dispatched enough. If you see this message, immediate attention is needed to ensure OMPROUTE gets the necessary processing. Or perhaps tuning is needed to increase the hello and dead router intervals.
6. ADDITIONAL SYMPTOMS you might notice if this scenario occurs:
 1. Storage growth in 4K ECSA - this is queued up protocol traffic which is not being read by OMPROUTE because it was swapped out or not dispatched.
 2. OMPROUTE neighbor state will loop in states 8-16 or 8-16-32
 3. OMPROUTE may determine that DEAD_ROUTER_INTERVAL has elapsed, or that the RIP timeout of 3 minutes has elapsed, and will therefore remove dynamically learned routes from the routing table

OSPF Timers

;Sample OSPF_INTERFACE (Broadcast: ethernet, token-ring, fddi, etc.):

```
OSPF_Interface
  IP_Address=9.59.101.5
  Name=TR1
  Subnet_Mask=255.255.255.0
  Attaches_To_Area=2.2.2.2
  MTU=1500
  Hello_Interval = 10
  Dead_Router_Interval = 40
  DB_Exchange_Interval=40
  Cost0=2
  Router_Priority=0;
```

PQ51195:

Specifying OMPROUTE_OPTIONS=hello_hi coded in the STDENV file will change the way OMPROUTE processes the OSPF Hello packets. These packets will be given a higher priority than other updates and processed by the first available OMPROUTE task ahead of other received packets. Prior to specifying this parameter customers must be cognizant of the impact to their network of processing hello packets out of the received order sequence.

© IBM Corporation 2001, 2002, 2003

1. Hello_Interval

1. This parameter defines the number of seconds between OSPF Hello packets being sent out this interface. This value must be the same for all routers attached to a common network. Valid values are 1 to 255 seconds.

2. Dead_Router_Interval

1. The interval in seconds, after not having received an OSPF Hello, that the neighbor is declared to be down. This value must be larger than the Hello_interval. Setting this value too close to the Hello_Interval can result in the collapse of adjacencies. A value of 4*Hello_Interval is recommended. This value must be the same for all routers attached to a common network. Valid values are 2 to 65535.

3. DB_Exchange_Interval

1. The interval in seconds that the database exchange process cannot exceed. If the interval elapses, the procedure will be restarted. This value must be larger than the Hello_interval. If no value is specified, the DB_Exchange_Interval will be set to the Dead router interval. Valid values are 2 through 65535.

4. STDENV Omproute Environment File coding

1. OMPROUTE_OPTIONS=hello_hi

1. Specifying OMPROUTE_OPTIONS=hello_hi changes the way OMPROUTE processes the OSPF Hello packets. These packets are then given a higher priority than other updates and processed by the first available OMPROUTE task ahead of other received packets. Prior to specifying this parameter, customers must be cognizant of the impact to their network of processing hello packets out of the received order sequence.
2. Note: Specifying OMPROUTE_OPTIONS=hello_hi only helps to keep adjacencies up when OMPROUTE is running and getting flooded with protocol packets. It does not provide any help for the case when adjacencies are not staying up because OMPROUTE is not getting enough cycles (that is, swapped out or running in too low a priority).

How Are the Timers Set?

D TCPIP,,OMPR,OSPF,LIST,ALL

EZZ7831I GLOBAL CONFIGURATION 967

TRACE LEVEL: 1, DEBUG LEVEL: 0, SADEBUG LEVEL: 0

STACK AFFINITY: TCPCS6

OSPF PROTOCOL: ENABLED

...

EZZ7833I INTERFACE CONFIGURATION

IP ADDRESS	AREA	COST	RTRNS	TRDLY	PRI	HELLO	DEAD	DB_EX
9.168.100.3	0.0.0.0	1	10	1	1	20	80	256
9.167.100.13	2.2.2.2	1	10	1	1	20	80	320

EZZ7836I VIRTUAL LINK CONFIGURATION

VIRTUAL ENDPOINT	TRANSIT AREA	RTRNS	TRNSDLY	HELLO	DEAD	DB_EX
9.67.100.8	2.2.2.2	20	5	40	160	480

EZZ7835I NBMA CONFIGURATION

INTERFACE ADDR	POLL INTERVAL
9.168.100.3	120

EZZ7834I NEIGHBOR CONFIGURATION

NEIGHBOR ADDR	INTERFACE ADDRESS	DR ELIGIBLE?
9.168.100.56	9.168.100.3	YES
9.168.100.70	9.168.100.3	NO

Why Are Adjacencies Dropped?

A hated message:

```
EZZ7921I OSPF adjacency failure, neighbor 9.5.1.2, old state 128, new state 1, event 12
```

What happened depends on the event code. The most common events are:

7 -- sequence number mismatch

▶ *Usually means the neighbor is attempting to restart the adjacency*

12 -- no hellos seen recently

▶ *We haven't heard from the neighbor for a full dead router interval, we assume he's down*

15 -- failure to thrive

▶ *Adjacency was trying to come up, but failed to complete within the DB_EXCHANGE_INTERVAL*

© IBM Corporation 2001, 2002, 2003

1. The message indicating lost adjacencies might have been missed in some installations in which the message was being sent only to SYSLOGD and not to the MVS error log. This has been corrected with the following APAR. .
 1. PQ50918 and PQ46720: V1R2 and V2R10
 1. Message EZZ7921I being sent only to SYSLOGD and not to MVS error log when designated router adjacency is lost.
 1. EZZ7921I OSPF Adjacency Failure, neighbor neighbor, old state state, new state state, event event
2. For reason code 7, usually it means the neighbor has not heard from OMPROUTE for a full dead router interval. So the neighbor marks its adjacency with OMPROUTE down, and then attempts to restart the adjacency. Since OMPROUTE still thinks the adjacency is up, it's not expecting an initial hello so considers it to be a sequence number mismatch. This could indicate that, while OMPROUTE is receiving HELLO packets from the neighbor, for some reason the neighbor is not getting ours.
3. For reason code 12, OMPROUTE is not receiving HELLO packets from the neighbor. This may indicate that the neighbor has actually gone down. If that is not the case, then perhaps the dead router interval is too short for this network design and load. Or perhaps OMPROUTE is not getting dispatched in a timely manner so is not processing the neighbor's HELLO packets quickly enough
4. Reason code 15 can indicate a death spiral. Recall that it takes more work to restart an adjacency than it does to maintain one. If OMPROUTE is barely able to keep up with hello timers, because of processor or network load, and a bunch of adjacencies drop, then the work of trying to restore them all will not help, and they may not be able to come back. This reason code is governed by the DB_EXCHANGE_INTERVAL. If an adjacency establishment does not fully complete in that time, it is torn back down and started over.
 1. If the DB_Exchange_Interval is too short, Neighbors will cycle between states 8-16-32...8-16-32...8-16-32 and may never actually reach state 128 (full adjacency) . The cycling is shown in EZZ7919I and EZZ7921I messages indicating event 12. You may need to raise this interval.

How to Track Adjacency Problems (1)

D TCPIP,,OMPR,OSPF,NBRS

EZZ7851I NEIGHBOR SUMMARY 015

NEIGHBOR ADDR	NEIGHBOR ID	STATE	LSRXL	DBSUM	LSREQ	HSUP	IFC
192.168.5.170	192.168.5.170	128	0	0	0	OFF	EZAXCFM2
192.168.11.92	172.16.1.92	128	0	0	0	OFF	TRLSM92A

D TCPIP,,OMPR,OSPF,NBRS,IPADDR=192.168.11.92

EZZ7852I NEIGHBOR DETAILS 019

NEIGHBOR IP ADDRESS: 192.168.11.92

OSPF ROUTER ID: 172.16.1.92

NEIGHBOR STATE: 128

PHYSICAL INTERFACE: TRLSM92A

DR CHOICE: 0.0.0.0

BACKUP CHOICE: 0.0.0.0

DR PRIORITY: 1

NBR OPTIONS: E

DB SUMM QLEN: 0 LS RXMT QLEN: 0 LS REQ QLEN: 0

LAST HELLO: 0 NO HELLO: OFF

LS RXMITS: 1 # DIRECT ACKS: 0 # DUP LS RCVD: 10

OLD LS RCVD: 0 # DUP ACKS RCVD: 1 # NBR LOSSES: 0

ADJ. RESETS: 0

© IBM Corporation 2001, 2002, 2003

1. This example shows some basic neighbor displays.
2. In the second display, note that the neighbor IP address and the OSPF router ID may be different. Recall that the OSPF Router ID identifies the router as a whole, while the Neighbor IP address identifies the IP address of the interface that attaches the router to the common medium.
3. Since the neighbor is fully adjacent yet there are no Designated Routers nor Backup Designated routers assumed, this is not a multiaccess medium. Beginning in z/OS V1R5 this will be made clearer by using "N/A" for irrelevant fields.

How to Track Adjacency Problems (3)

D TCPIP,NM1ATCP,OMPR,OSPF,IFS

EZZ7849I INTERFACES 520

IFC ADDRESS	PHYS	ASSOC. AREA	TYPE	STATE	#NBRS	#ADJS
192.168.5.168	EZASAMEMVS	1.1.1.1	P-2-MP	16	1	1
192.168.5.168	EZAXCFM2	1.1.1.1	P-2-MP	16	1	1
9.82.5.122	VIPL0952057A	1.1.1.1	VIPA	N/A	N/A	N/A
192.168.51.168	TRLSM94A	0.0.0.0	P-2-MP	16	1	1
192.168.31.168	TRLSM93A	0.0.0.0	P-2-MP	16	1	1
192.168.11.168	TRLSM92A	0.0.0.0	P-2-MP	16	1	1
172.18.2.168	CTCC128	1.1.1.1	P-P	1	0	0
9.82.5.121	VLINK2	1.1.1.1	VIPA	N/A	N/A	N/A
9.82.5.120	VLINK1	1.1.1.1	VIPA	N/A	N/A	N/A

REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND

- State is "Interface State" -- not "Neighbor State"
- State "1" = Interface is down
- State "16" = Interface is Point-to-Point

© IBM Corporation 2001, 2002, 2003

1. Interface STATE can be one of the following. (Do not confuse these states with Neighbor States.)
 1. 1 (down)
 2. 2 (backup)
 3. 4 (looped back)
 4. 8 (waiting)
 5. 16 (point-to-point)
 6. 32 (DR other)
 7. 64 (backup DR)
 8. 128 (designated router)
 9. N/A (interface is a VIPA and does not actually come up)
2. #NBRS Number of neighbors.
 1. This is the number of routers whose hellos have been received, plus those that have been configured.
3. #ADJS
 1. Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

How to Track Adjacency Problems (4)

D TCPIP,NM1ATCP,OMPR,OSPF,IFS,NAME=EZAXCFM2

EZZ7850I INTERFACE DETAILS 524

```

                INTERFACE ADDRESS:      192.168.5.168
                ...
                INTERFACE TYPE:         P-2-MP
                STATE:                   16
                DESIGNATED ROUTER:      0.0.0.0
                BACKUP DR:                0.0.0.0
DR PRIORITY:      1  HELLO INTERVAL:    20  RXMT INTERVAL:    5
DEAD INTERVAL:   80  TX DELAY:          1  POLL INTERVAL:    0
DEMAND CIRCUIT: OFF  HELLO SUPPRESS:    OFF  SUPPRESS REQ:     OFF
MAX PKT SIZE:   16384  TOS 0 COST:      1  DB_EX INTERVAL:   80
AUTH TYPE:      NONE
# NEIGHBORS:     1  # ADJACENCIES:      1  # FULL ADJS.:      1
# MCAST FLOODS: 32  # MCAST ACKS:      20  DL UNICAST:        OFF
MC FORWARDING:  OFF
NETWORK CAPABILITIES:
POINT-TO-POINT
EMULATED-BROADCAST
DEMAND-CIRCUITS

```

© IBM Corporation 2001, 2002, 2003

- Interface STATE can be one of the following. (Do not confuse these states with Neighbor States.)
 - 1 (down)
 - 2 (backup)
 - 4 (looped back)
 - 8 (waiting)
 - 16 (point-to-point)
 - 32 (DR other)
 - 64 (backup DR)
 - 128 (designated router)
- Had this been a broadcast connection, the DR and BDR fields would have been filled in with the address of these two nodes. In z/OS V1R5 this display will show N/A for irrelevant fields
 - DESIGNATED ROUTER IP address of the designated router. BACKUP DR IP address of the backup designated router.
- This display shows you exactly with how many nodes the interface has established FULL Adjacencies. This is in contrast with the previous display that contained information only on "neighbors" and on "adjacencies."
- # NEIGHBORS Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.
- # ADJACENCIES Number of adjacencies. This is the number of neighbors in state Exchange or greater.
- # FULL ADJS Number of full adjacencies. This is the number of neighbors whose state is Full (and therefore with which the router has synchronized databases).
- # MCAST FLOODS Number of link state updates that flooded the interface (not counting retransmissions).

How Does OMPROUTE Drop Adjacencies



- Other load on the z/OS machine keeps OMPROUTE from dispatching enough
 - ▶ *taking dumps -- all address spaces marked non-dispatchable during dump processing*
 - ▶ if the dump takes longer than a dead router interval, the adjacencies will fail
 - ▶ *too many other address spaces running higher priority than OMPROUTE*
 - ▶ *OMPROUTE should be one less than TCP/IP's dispatching priority*
- Not enough dispatching priority for OMPROUTE
 - ▶ *either increase OMPROUTE's priority or increase the dead router intervals*
 - ▶ *don't run OMPROUTE as BPXBATCH program*

© IBM Corporation 2001, 2002, 2003

1. If you are unwilling or unable to give OMPROUTE the priority and cycles it needs on the machine, than increase the dead router interval so that adjacencies can survive prolonged OMPROUTE starvation
2. When OMPROUTE runs as a BPXBATCH program, it cannot be made non-swappable; i.e., it cannot be placed in the PPT. Furthermore, running it as BPXBATCH makes the job of setting an appropriate WLM Service Class difficult.

OMPROUTE: High Priority or Service Class

JOBNAME	...	DP	...	U%	Workload	SrvClass
BPXOINIT		FF		4	SYSTEM	SYSTEM
...						
CRON6		FF		4	SYSTEM	SYSOTHER
CSSVTAM		FE		4	SYSTEM	SYSSTC
...						
JESXCF		FF		4	SYSTEM	SYSTEM
JES2		FE		4	SYSTEM	SYSSTC
...						
NM1AOMPR		FE		4	SYSTEM	SYSSTC
NM1ASYSL		FF		4	SYSTEM	SYSOTHER
NM1ATCP		FE		4	SYSTEM	SYSSTC
...						
NM1RSLVR		FE		4	SYSTEM	SYSSTC

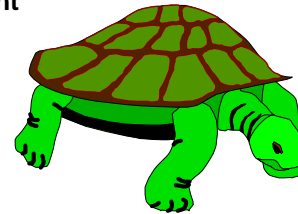
- To change Service Class:
 - Key over SDSF SrvClass column (requires proper WLM Authority), or
 - RESET NM1AOMPR,SRVCLASS=xxxx, or
 - Make change permanent through WLM displays

© IBM Corporation 2001, 2002, 2003

Self-inflicted OMPROUTE Performance Problems

The following things can cause OMPROUTE performance to be unacceptable even when it gets enough cycles

- **Too much tracing**
 - OMPROUTE debug tracing using file I/O and really hurts performance
 - relief is on the way in z/OS V1R5 with CTRACE trace enhancements
- **Too big a route table**
 - Once OMPROUTE has more than 1000-2000 routes, start expecting trouble
 - Try to keep z/OS out of backbone areas. Stub areas are ideal.
- **Too many adjacencies**
 - This may especially be a problem in an XCF environment
 - do XCF interfaces really have to be OSPF?



© IBM Corporation 2001, 2002, 2003

1. In z/OS V1R5 OMPROUTE will be able to write its debug trace to CTRACE, which is much more efficient than the current file I/O method

Get the Most out of OMPROUTE Performance

➤ Use recommended network design practices

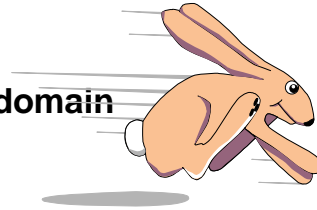
- ▶ *Try to put z/OS and the sysplex into a stub or totally stubby area or isolate areas with BGP or EIGRP*
 - ★ to minimize routing table size and advertisements that have to be processed
- ▶ *Try to avoid OMPROUTE becoming designated router on LANs*
 - ★ to minimize OSPF adjacencies
 - ★ may be unavoidable on Hipersockets LAN since no routers attached
- ▶ *OMPROUTE has full OSPF function but it's unlikely you bought a z/OS box to be a network router*
 - ★ let the routers do the routing work
 - ★ unlike on dedicated routers, OMPROUTE has to share machine resources with your business-critical applications!

➤ Only use debug tracing when necessary

➤ Consider taking XCF links out of the OSPF domain

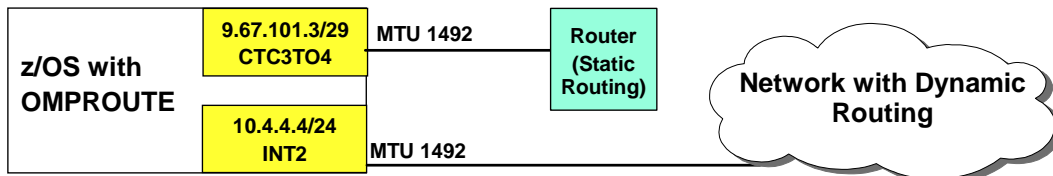
- ▶ if not used for data traffic
- ▶ cuts down significantly on adjacencies

© IBM Corporation 2001, 2002, 2003



1. The sysplex function maintains its own routing table. Therefore it is not necessary for OMPROUTE to advertise XCF links for basic sysplex functions like DVIPA and DDVIPA to work properly. If XCF links will not be used for application data traffic, define them to OMPROUTE using INTERFACE statements to avoid adjacencies being set up over them.
2. Because XCF is a point to multipoint medium, if it is in the OSPF domain it will have fully meshed adjacencies instead of the more efficient designated router model of LANS. Fully meshed adjacencies over XCF can really be a load on your system if there are a lot of LPARs in the sysplex.

Spurious Network Route



Most likely cause: stack interface(s) not defined to OMPROUTE

● **If CTC3TO4 is coded in OMPROUTE.CONF:**

● **Routing Table Display "NETSTAT GATE"**

```
MVS TCP/IP NETSTAT CS V1R5      TCPIP Name: TCPCS3      14:42:08
Known gateways:
NetAddress  FirstHop  Link      Pkt Sz Subnet Mask  Subnet Value
9.0.0.0     <direct> CTC3TO4   1492   0.255.255.248 0.67.101.0
```

● **If CTC3TO4 is not coded in OMPROUTE.CONF:**

● **Routing Table Display "NETSTAT GATE"**

```
MVS TCP/IP NETSTAT CS V1R5      TCPIP Name: TCPCS3      14:54:21
Known gateways:
NetAddress  FirstHop  Link      Pkt Sz Subnet Mask  Subnet Value
9.0.0.0     <direct> CTC3TO4   576    <none>
```

© IBM Corporation 2001, 2002, 2003



1. Stack interfaces may not be defined to OMPROUTE because they simply were not defined, or because of a typographical error in the definition
2. For IPv4, OMPROUTE sets the BSDROUTINGPARMS MTU value and subnet mask on every interface
 1. If an interface exists on the stack but is not defined to OMPROUTE, OMPROUTE sets default values for the MTU and subnet mask, overriding any other BSDROUTINGPARMS definitions
 1. Default MTU size is 576
 2. Default subnet mask is the class mask of the interface's home address
 1. The default subnet mask could cause undesirable results. For example, an undefined interface with home address 9.67.101.3 would cause OMPROUTE to assign a subnet mask of 255.0.0.0. **This in turn would cause OMPROUTE to add a route to the stack's routing table to 9.0.0.0/8 over that interface -- which is probably not was was intended.**
 2. If OMPROUTE is an Autonomous System Boundary Router and is importing direct routes, it would also advertise that 9.0.0.0/8 route to the rest of the OSPF routers, causing all traffic to unknown destinations in the 9 network to be sent to the TCP/IP stack and then forwarded over that interface!
 2. Solution: make sure all stack interfaces are defined to OMPROUTE, even the ones that are not running RIP or OSPF.
 3. Because of these problems, IBM has long recommended that every stack interface be defined to OMPROUTE, regardless of whether or not dynamic routing will be used over it.
3. INTERFACE configuration statement used for interfaces over which no dynamic routing will be done.

Debugging Undefined Stack Interfaces

If you suspect an undefined stack interface, look for the following message in OMPROUTE trace:

→ *-t1 trace level must be activated*

```
EZZ7966I No matching interface statements for 9.67.101.3 (CTC3TO7)
```

Starting with z/OS V1R5, you will be able to display undefined interfaces in OMPROUTE:

```
d tcpip,,omproute,generic,ifs
EZZ8060I IPV4 GENERIC INTERFACES
IFC NAME          IFC ADDRESS      SUBNET MASK      MTU  CFG  IGN
CTC3TO7          9.67.101.3      N/A              N/A  NO  NO
```



6. Common Configuration Complaints

© IBM Corporation 2001, 2002, 2003

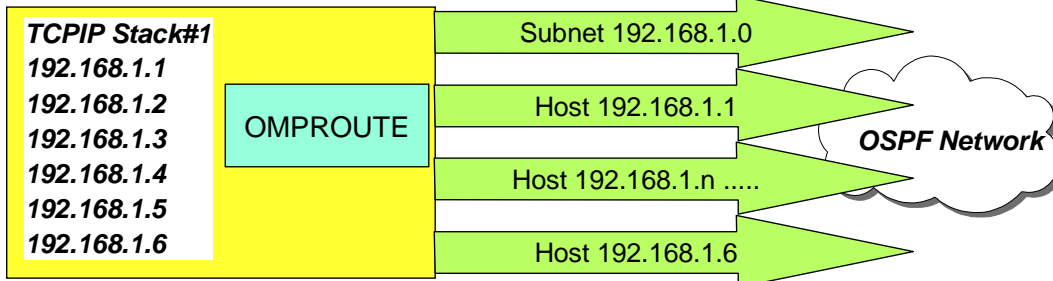
My Router Folks Want a "255" Mask

OMPROUTE.Conf Coding for Dynamic VIPAs

```

OSPF_Interface
IP_address = 192.168.1.*
Subnet_mask = 255.255.255.248
Attaches_To_Area=1.1.1.1
Subnet=No
MTU=65535;

```



Subnet_Mask

The subnet mask of the subnet to which this interface attaches. This value must be the same for all routers attached to a common network.

© IBM Corporation 2001, 2002, 2003

1. Many router administrators are used to being able to code a host route mask for individual IP addresses when they code static routes. The host route mask is 255.255.255.255 or 32 bits long in IP Version 4.
2. z/OS System Programmers are often pressed to code this host route mask in the OMPROUTE configuration file, when, in fact, this is a violation of the RFCs.
 1. It is also unnecessary because many types of interfaces, including point-to-point and VIPA interfaces will automatically have both their host routes and their subnet routes broadcast into the network.

My Router Folks Don't Like Host Routes

```

PROFILE.TCPIP for Dynamic VIPA Block

VIPADYNAMIC
VIPAFINE 255.255.255.248 192.168.1.001
.....
VIPAFINE 255.255.255.248 192.168.1.006
ENDVIPADYNAMIC
    
```

```

OMPROUTE.Conf Coding for Dynamic VIPAs

OSPF_Interface
IP_address = 192.168.1.*
Subnet_mask = 255.255.255.248
Attaches_To_Area=1.1.1.1
Subnet=No
MTU=65535;
    
```

But don't suppress host routes unless you have to!!



Subnet = No (default)	Send Host Routes plus Subnet Routes
Subnet = Yes	Send Subnet Routes <u>ONLY</u>

© IBM Corporation 2001, 2002, 2003

1. Many router networking technical people are overwhelmed if they have to manage routing tables for a plethora of host routes. In some cases it is possible to suppress host route advertisements. In OMPROUTE the parameter SUBNET may be used to suppress host routes on certain types of interfaces if necessary and to broadcast only subnet routes for certain interfaces. However, heed the warning note in the description of this parameter below.
2. Subnet = YES | NO
 1. For an interface to a point-to-point serial line, this option (SUBNET=YES) enables the advertisement of a stub route to the subnet that represents the serial line rather than the host route for the other router's address.
 2. For a VIPA interface, this option (SUBNET=YES) suppresses advertisement of the VIPA host route. Normally CS for z/OS advertises both a host route and a subnet route for owned VIPA interfaces. With this option set to YES, only the subnet route will be advertised. Note that the default is SUBNET=NO, meaning that both host routes and subnet routes will be advertised.
3. **Note: Do not use this option of SUBNET=YES for dynamic VIPAs or for any VIPA whose subnet might exist on multiple hosts. If you do, problems can occur routing to all VIPAs that share the subnet.**

For More Information ...



URL	Content
http://www.ibm.com/servers/eserver/zseries	IBM Enterprise Servers (z900 & S/390)
http://www.ibm.com/servers/eserver/zseries/networking	z900 Networking
http://www.ibm.com/servers/eserver/zseries/networking/technology.html	Networking White Papers and Information
http://www.ibm.com/software/network	Networking & Communications Software
http://www.ibm.com/software/network/commserver	Communications Server
http://www.ibm.com/software/network/commserver/library	CS White Papers, Product Doc, etc.
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.rfc-editor.org/rfcsearch.html	RFCs
http://www.ibm.com/support/techdocs/	Advanced Technical Support (Flashes, Presentations, White Papers, etc.)

Whitepaper on OSPF with IBM & CISCO:
"OSPF Design and Interoperability Recommendations for Catalyst 6500 and OSA-Express Environments" http://www-1.ibm.com/servers/eserver/zseries/networking/pdf/ospf_design.pdf
Redbook on "Networking with z/OS and Cisco Routers: An Interoperability Guide (SG24-6297)"

Bibliography

RFCs 1583 & 2328	OSPF V2 by John Moy: URL http://www.ietf.cnri.reston.va
GG24-3376	TCP/IP Tutorial and Technical Reference
SC31-8513	OS/390 Communications Server: IP Configuration
Huitema, Christian	Routing in the Internet, Prentice Hall
Black, Uyless	IP Routing Protocols: RIP, OSPF, BGP, PNNI & CISCO Routing Protocols (SR23-9498)
Parkhurst, William R.	Cisco Router OSPF Design and Implementation Guide, Parkhurst McGraw-Hill (SR23-8683)
Cisco Systems	CCIE Fundamentals: Network Design and Case Studies, 2nd Edition (SR23-9437)
Doyle, Jeff (Cisco Systems)	CCIE Professional Development: Routing TCP/IP, Volume 1 (SR23-9241)
SG24-5631	SecureWay Communications Server for OS/390 V2R8 TCP/IP Guide to Enhancements
SG24-5227	V2Rn TCP/IP Implementation Guide, Vol. 1 (Configuration and Routing)
SG24-5228	V2Rn TCP/IP Implementation Guide, Vol. 2 (UNIX Apps)
SG24-5229	V2Rn TCP/IP Implementation Guide, Vol. 3 (MVS Apps)
FLASH N3196	TCP/IP Migration Tips & Hints (V2R8) at www.ibm.com/support/techdocs
FLASH N3190	OMPROUTE and VIPA at www.ibm.com/support/techdocs

Bibliography V1R2 Redbooks

SG24-5227-03	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 1: Base and TN3270 Configuration
SG24-5228-03	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 2: UNIX Applications
SG24-6516-00 	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 4: Connectivity and Routing
SG24-6517-00	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 5: Availability, Scalability, and Performance
SG24-6839-00	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 6: Policy and Network Management
SG24-6840-00	Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 7: Security
SG24-5229-01	OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 3: MVS Applications
SG24-6297-00 	Networking with z/OS and Cisco Routers: An Interoperability Guide

© IBM Corporation 2001, 2002, 2003

1. These redbooks are all now available from www.redbooks.ibm.com.
2. The new Volume 4 contains information not only on implementing OSPF in z/OS, but also information on integrating with Cisco Routers, including the integration with EIGRP.
3. Note also that the MVS Applications volume of the redbook series has not been updated since OS/390 V2R7, although there have been enhancements to several of the socket applications discussed there. Notably, the CICS sockets application has received important enhancements in the z/OS V1R2 release. These CICS enhancements are included in one of the presentations in this course.
4. The final redbook mentioned also contains information on using EIGRP and BGP with Cisco.



End

© IBM Corporation 2001, 2002, 2003



End

© IBM Corporation 2001, 2002, 2003