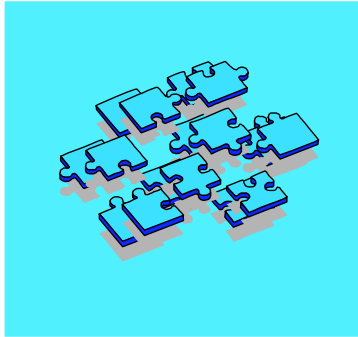


NSTC: N43

IP Network Design Considerations: OSPF Configuration & Design in z/OS CS

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



Orlando, September 23 - September 27, 2002

Wednesday, September 25: 15:45 - 17:00

Thursday, September 26: 14:00 - 15:15

© IBM Corporation 2002

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 second of two dealing with OSPF in CS for OS/390 or z/OS. The first session, Session Number S3914, describes the architecture of OSPF and the OSPF features that are implemented in OS/390 or z/OS. This, the second session on OSPF in CS for OS/390 or z/OS, describes the implementation of OSPF in OMPROUTE in Communications Server for OS/390 or z/OS. It makes recommendations on the integration of OS/390 (z/OS) into an OSPF network, shows you how to code for several network design options, and introduces you to tools such as traces and command displays to understand the OSPF environment in z/OS.

> Acknowledgements: Many thanks to Alfred Christensen and Alan Packett of the IBM Design and Development groups in Raleigh for their suggestions and several visuals. Many thanks to Mike Law of IBM Integrated Test Services for the CISCO examples.



Agenda

1. **Basics of Operating OMPROUTE**
2. **Basics of Configuring OSPF in OMPROUTE**
3. **Network Design: OSPF in CS**
 1. **General Design Guidelines**
 2. **Sample Configurations for Various Network Designs**
 3. **Summary of Hints for OMPROUTE & OSPF**
4. **Diagnostics with OMPROUTE**
 1. **OMPROUTE in the TCP/IP Stack**
 2. **Command Operations and Interpretation**
 3. **Trace Sample in OMPROUTE**
5. **Enhancements for OSPF in z/OS V1R2**
6. **A. Appendices**
 - A.1 **Cisco Examples**
 - A.2 **More Command Output Examples**
 - A.3 **Miscellaneous: Common startup errors, RACF considerations, SNMP, Multipath, etc.**
 - A.4 **Bibliography**





1. Basics of Initializing OMPROUTE

Attend N64: High Availability Connections to the Mainframe
(Speaker: Mike Blair, Cisco Systems)
10AM -- Scotland B



© IBM Corporation 2001, 2002

OMPROUTE Procedure: STDENV

```
//OMPROUTE PROC
//OMPROUTE EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
// PARM=(' POSIX(ON) ',
// 'ENVAR("_CEE_ENVFILE=DD:STDENV")/')
//STDENV DD PATH='/u/usernnn/envcs2',
// PATHOPTS=(ORDONLY)
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

```
# This is /u/usernnn/envcs2
RESOLVER_CONFIG=// 'SYS1.TCPPARMS(TCPDATA2) '
OMPROUTE_FILE=/u/usernnn/config.tcpcs2
OMPROUTE_DEBUG_FILE=/tmp/logs/omproute.deh
OMPROUTE_DEBUG_FILE_CONTROL=1000,5
export TZ=EST4EDT
```

```
Autolog
OMPROUTE
EndAutolog
Port
520 UDP OMPROUTE NoAutolog
```



1. This cataloged procedure runs the OMPROUTE daemon as a BPXBATCH program.
2. The STDENV DD card points to a file containing the environment definitions ("OMPENV"). This file may be an HFS or an MVS dataset, as in this example.
 1. The STDENV file contains the location and name of the OMPROUTE configuration file, of the resolver configuration file (TCPDATA), possibly an OMPROUTE debug file, or even other environment variables, such as the time zone that will provide correct timestamp data.
 2. The OMPROUTE configuration file may be placed in an MVS dataset or in an HFS file. If it is placed in an MVS dataset, ensure that there are no leading blanks or trailing numbers. Otherwise the Configuration file (and any other file coded in an MVS dataset) may not be found.
 3. Note an important point if you have a STDENV definition that resides in a Fixed Block (FB) dataset. The _CEE_ENVFILE variable has a restriction with FB datasets. It does not strip trailing blanks from the end of the variable. Thus, you have the problem that ishell can't view files in the HFS, point STDENV to an HFS file or to a Variable Blocked MVS file; in this case the file will be created without blanks.
 4. Note how we point to the TCPIP.DATA file of the stack with which this OMPROUTE procedure should be associated.
 1. If there are multiple stacks running in an MVS image, each stack would have its own OMPROUTE configuration.



2. Basics of Configuring OSPF in OMPROUTE



© IBM Corporation 2001, 2002

Syntax Rules for OMPROUTE

- The configuration file is specified using the **OMPROUTE_FILE** environment variable.
- The configuration file may be coded within an **MVS** dataset or an **HFS** file.
- The file is not case-sensitive
- Every configuration statement ends with a semicolon (";")
- Comments must not appear within a configuration statement; they are preceded by a semicolon in any column
- Statements may begin in any column
- Sequence numbers may not appear in the data set or file.



1. See previous section on setting up environment variables for OMPROUTE to see more detail on how to specify the OMPROUTE configuration file

OSPF Configuration File Sample

```
RouterID=10.138.165.9
```

Specify RouterID as static VIPA if multiple interfaces in CS.

```
;
```

```
Area
```

```
Area_Number=1.1.1.1
```

```
Stub_Area = NO
```

```
;
```

```
AS_Boundary_Routing
```

```
Import_RIP_Routes = YES
```

```
Import_Static_Routes = YES
```

This CS in OS/390 stack is an ASBR:

It is importing (redistributing) external routes from RIP and from the Static Configuration in the GATEWAY of the IP PROFILE.

```
;
```

```
ROUTESA_CONFIG
```

```
COMMUNity="FRED"
```

```
AGENT=161
```

```
ENABLEd=YES
```

This enables the OMPROUTE OSPF Subagent.

```
;
```



1. Assuming you have more than one interface in a CS node, always code the Router_ID. We recommend coding it with the value of a static VIPA so that it does not by chance default to a dynamic VIPA which could move to another node and generate confusion in the OSPF network.
2. If you are an ASBR, you may choose to Import routes from other protocols. Even if you are importing the Default Route from another protocol, you must specify AS_Boundary_Routing parameters.
3. The ROUTESA_CONFIG statement permits the definition of keywords to interface the OMPROUTE Subagent with SNMP. The OMPROUTE subagent provides an alternative to DISPLAY commands for displaying Open Shortest Path First (OSPF) protocol configuration and state information. The subagent implements the Management Information Base (MIB) variables defined in Request for Comments (RFC) 1850 for the OSPF MIB.

Sample Omproute Configuration File (1)

```

Area Area_Number=1.1.1.1;
Area Area_Number=0.0.0.0;
;Sample OSPF_INTERFACE (Point-to-Point -- CTC, CLAW, ETC.)
  OSPF_Interface
    IP_Address=9.67.113.80
    Name=CTCD00
    Subnet_Mask=255.255.255.252
    Destination_Addr=9.67.113.81
    Hello_Interval = 2
    Dead_Router_Interval = 8
    MTU=4096
    Attaches_To_Area=1.1.1.1
    Cost0=5;
;Sample RIP_INTERFACE (Point-to-Point -- CTC, CLAW, ETC.)
  RIP_Interface
    IP_Address=9.67.114.80
    Name=CTCA00
    Subnet_Mask=255.255.255.252
    Destination_Addr=9.67.114.81
    RIPV2=Yes;
;Sample INTERFACE (Point-to-Point -- CTC, CLAW, ETC.)
  Interface
    IP_Address=9.17.214.10
    Name=CTCB00
    Subnet_Mask=255.255.255.252
    Destination_Addr=9.17.214.11;
  
```

V2R10 Change:
255.255.255.252

V2R10 Change:
255.255.255.252

V2R10 Change:
255.255.255.252

Specify Link Name from IP Profile
Specify Subnet Mask to avoid default of standard Class Mask.

Specify MTU to avoid default
Area defaults to 0.0.0.0
Specify reasonable cost

IBM Corporation 2001, 2002

1. The softcopy for this file is in SEZAINST(EZAORCFG) and in DOC APAR II11555.
2. In earlier versions of OMPROUTE (prior to V2R10), it was possible to use a subnet mask of 255.255.255.254. Earlier versions of the DOC APAR II11555 reflected that subnet mask. Take note that beginning with V2R10, such a mask is invalid. Disregard any documentation that suggests a mask of 255.255.255.254 as a means of minimizing address consumption. APAR PQ50921 has been taken against the samples delivered with OS/390 V2R10 and z/OS V1R2 to reflect this change: "Due to improved validation of outbound packets per RFC 1860, the smallest subnet mask value allowed is 255.255.255.252."
3. CS/390 OSPF perceives a CLAW interface as point-to-point and a CISCO router codes it as point-to-multipoint. Note that Cisco does not have a configuration parameter for P2P for CLAW; it must be configured using the P2MP *AND* BROADCAST parameters to treat CLAW like P2P for compatibility with the IBM CLAW.
 1. For example, a CISCO router considers a broadcast network to be point-to-multipoint; a non-broadcast network is point-to-point. Accordingly the CISCO router will send in Link State Advertisements according to RFC 2328, section 12.4. If you have problems connecting properly or receiving appropriate route updates, you may have a mismatch in the configurations between CS/390 and the router.
4. APARs PQ35265 and PQ48766 describe how to correct these scenarios. Please note new CLAW syntax for LINK in the IBM stack: P2MP. See examples in the Appendix of the presentation.
5. Generally speaking, this is how you use the different types of interface coding:
 1. Use OSPF_INTERFACE if the OSPF protocol will be communicated with other router(s) over the interface
 2. Use RIP_INTERFACE if the RIP protocol will be communicated with other router(s) over the interface
 3. Use INTERFACE if neither the OSPF nor the RIP protocol will be communicated with other router(s) over the interface
6. For these interfaces, the DESTINATION_ADDR parameter should be specified to allow for the creation of a host route to the address at the remote end of the interface.
7. Consult APAR PQ35265 for information on connecting to routers using OSPF protocols.
8. REMEMBER: HELLO INTERVAL and DEAD ROUTERINTERVAL must be the same at both ends of a connection!

Sample Omproute Conf: Pt-Multipt (2)

```
;Sample OSPF_INTERFACE (MPC, IUT, XCF):
```

```
OSPF_Interface
  IP_Address=9.27.13.81
  Name=XCFD00
  Subnet_Mask=255.255.255.0
  MTU=4096
  Attaches_To_Area=1.1.1.1
  Hello_Interval = 2
  Dead_Router_Interval = 8
  Cost0=7;
```

Specify Hello_Interval and
Dead_Router_Interval and ensure
that the same values are used at
other ends of the connection.

```
;Sample RIP_INTERFACE (MPC, IUT, XCF):
```

```
RIP_Interface
  IP_Address=9.46.33.181
  Name=MPCA01
  Subnet_Mask=255.255.255.0
  MTU=4096
  RIPV2=Yes;
```

```
;Sample INTERFACE (MPC, IUT, XCF):
```

```
Interface
  IP_Address=9.77.13.49
  Name=XCFB00
  Subnet_Mask=255.255.255.0
  MTU=4096 ;
```



1. Even though XCF and IUTSAMEH links are Pt-to-Multipoint networks that are often considered NBMA networks, you do not need to specify neighbors.
 1. Neighbor IP addresses are exchanged with partners during link activation.

Sample Omproute Config: NBMA (3)

```
;Sample OSPF_INTERFACE (Hyperchannel, ATM, X.25):
```

```
OSPF_Interface
  IP_Address=9.37.84.49
  Name=HCHE00
  Subnet_Mask=255.255.255.0
  Attaches_To_Area=0.0.0.0
  MTU=1024
  Router_Priority=2
  Non_Broadcast=Yes
  DR_Neighbor=9.37.84.53
  No_DR_Neighbor=9.37.84.63;
```

On Non-Broadcast Multiaccess network, identify neighbors and specify whether they are eligible to be the DR.

```
;Sample RIP_INTERFACE (Hyperchannel, ATM):
```

```
RIP_Interface
  IP_Address=9.37.104.79
  Name=ATME00
  Subnet_Mask=255.255.255.0
  MTU=1024
  RIPV2=Yes
  Neighbor=9.37.104.53
  Neighbor=9.37.104.85;
```

```
;Sample INTERFACE (Hyperchannel, ATM):
```

```
Interface
  IP_Address=9.77.13.49
  Name=ATMB00
  Subnet_Mask=255.255.255.0
  MTU=1024 ;
```



1. Note the coding of the following parameters for OSPF in an NBMA network:

1. Router_Priority=2
 1. Designates eligibility to be a DR if it is non-zero
 2. Default value is 1, so not necessary to code this to be eligible to be a designated router
2. Non_Broadcast=Yes
3. DR_Neighbor=9.37.84.53
4. No_DR_Neighbor=9.37.84.63
 1. 9.37.84.63 may not be the DR even if it says it is eligible to become one by having specified a Priority of non-zero.

Sample Omproute Config: Broadcast (4)

```
;Sample OSPF_INTERFACE (Broadcast: token-ring, ethernet, fddi):
; (See notes for information about OSA express and Hipersockets)
```

```
OSPF_Interface
  IP_Address=9.59.101.5
  Name=TR1
  Subnet_Mask=255.255.255.0
  Attaches_To_Area=0.0.0.0
  MTU=1500
  Hello_Interval = 2
  Dead_Router_Interval = 8
  Cost=2
  Router_Priority=1;
```

Specify Hello_Interval and Dead_Router_Interval and ensure that the same values are used at other ends of the connection.

A non-zero value for Router_Priority: router is eligible to become a Designated Router.

```
;Sample RIP_INTERFACE (Broadcast: token-ring, ethernet, fddi):
```

```
RIP_Interface
  IP_Address=9.29.107.3
  Name=TR2
  Subnet_Mask=255.255.255.0
  MTU=1500
  RIPV2=Yes;
```

```
;Sample INTERFACE (Broadcast: token-ring, ethernet, fddi):
```

```
Interface
  IP_Address=9.77.13.49
  Name=ETHB00
  Subnet_Mask=255.255.255.0
  MTU=1500 ;
```



1. OSA Express and Hipersockets do not support broadcast but they do support multicast. Since multicast is the only type of "broadcast" that OSPF does, these interfaces are considered broadcast interfaces for OSPF purposes. For RIP purposes, you should use RIP v2 over these types of interfaces since RIP v1 uses broadcast (which is not supported by these link types) and not multicast (which is).
2. If you define a HiperSockets network interface with a definition similar to the one on this page, ensure that you have allowed the HiperSockets connection to be DR-eligible by coding or defaulting Router_Priority=<non-zero>.

Sample Omproute Config: Static Vipa (5)

; Sample VIPA interface for no OSPF in use on this router:

```
Interface
  IP_Address = 10.138.165.9
  Name = VLINK1
  Subnet_Mask = 255.255.255.248
  Destination_Addr = 10.138.165.9;
```

VIPA is either "Interface" or "OSPF_Interface"

; Sample VIPA if at least one OSPF configured in file:

```
OSPF_Interface
  IP_address = 10.138.165.9
  Name = VLINK1
  Subnet_mask = 255.255.255.248
  Destination_Addr=10.138.165.9
  Attaches_To_Area=1.1.1.1;
```

MTU size is irrelevant, but ...
Destination_Addr is optional.
Cost is irrelevant.

MTU = 65535

Remember to apply APARs PQ25823 and PQ29859 if you are running code earlier than V2R8 and require "host routes" for your VIPA addresses. VIPAs that belong to the same subnet but that are installed in several stacks require this support in OMPROUTE.



1. If there are no OSPF_Interfaces coded in this OMPROUTE, the VIPA is configured as an "Interface."
 1. If there is at least one OSPF_Interface in this OMPROUTE, the VIPA is configured as an "OSPF_Interface"
2. Information on coding for VIPA, both static VIPA and Dynamic Vipa (DVIPA) is included in SEZAINST(EZAORCFG) and in DOC APAR II11555 as well as in the appropriate IP Configuration Guide.
3. Consult FLASH 3190 at www.ibm.com/support/techdocs for information on VIPA and OMPROUTE.
4. Apply the PTFs referred to by APAR PQ25823 if you require "host routes" for your VIPA configuration. PQ25823 is in the base of V2R8 and beyond. Please consult also PQ29859.
 1. OSPF in OMPROUTE will send host routes for all interfaces except for Broadcast networks. You may suppress subnet broadcasting with PT-PT links with a parameter on the OSPF_Interface statement. (This allows host routes only for the PT-PT links.)
5. MTU size is not required for VIPA. However, the OSPF router LSA (Type 1 LSA) must fit into one datagram whose size is limited by the largest MTU of all the interfaces. The router LSA cannot be spread across several datagrams - it must be a single message that can then be fragmented. A 64K MTU on a VIPA works around the limitation of a router LSA being too large to fit into the frame size allowed for OMPROUTE. (In any case, 64K is the largest packet size allowed by IP.)
 1. This is really only a consideration for boxes that have huge numbers of interfaces (for example, many Dynamic VIPAs) who are having trouble sending their router LSA because it cannot fit in the maximum MTU size of the real interfaces.
6. The original documentation on VIPA definitions recommended the use of Destination_Addr. In fact, this parameter is optional.

Samples for Dynamic XCF, HiperSockets, and VIPA Dynamics

PROFILE.TCPIP for Dynamic XCF

```
IPConfig DynamicXCF 192.168.2.2 255.255.255.0 1
```

Dynamic XCF in OMPROUTE
(Can also define the
HiperSockets Interface)

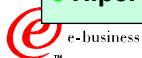
```
OSPF_Interface
IP_address = 192.168.2.*
MTU=4096
Subnet_mask = 255.255.255.0
Router_Priority=1; (default)
```

VIPA Dynamics in OMPROUTE

```
OSPF_Interface
IP_address = 10.138.165.80
Name = dummy1
Subnet_mask = 255.255.255.240;
```

Dynamic XCF in OMPROUTE

- EZASAMEMVS or IUTSAMEH Connection (DR not necessary; ignored)
- XCF Data Connection (DR not necessary; ignored)
- HiperSockets Connection (DR required)



© IBM Corporation 2001, 2002

1. DYNAMIC XCF links are coded in the PROFILE.TCPIP with a statement similar to the following:
 1. IPConfig DynamicXCF 192.168.2.2 255.255.255.0 1
 2. This very definition can also generate a HiperSockets DEVICE/LINK if HiperSockets have been enabled in the IOCPgen of a zSeries platform and no competing HiperSockets DEVICE/LINK has been defined on the same HiperSockets CHPID.
2. To avoid having to determine what the names of the dynamic XCF links will be, it is better to define them with wildcard IP addresses. In this fashion, the syntax checker does not require that the "Name" be specified.
 1. The subnet mask and metric you specified on the DynamicXCF PROFILE statement is for ORouteD use only. You must add an Ospf_Interface statement to your OMPRoute configuration file if you want OSPF advertisements to be exchanged over the XCF network.
 2. The XCF and IUTSAMEH connections that are created by means of the DYNAMICXCF definition do not require Designated Router (DR) election because they are a special type of Point-to-Multipoint connection.
 3. However, HiperSockets represent a broadcast network -- therefore HiperSockets connections, which may also be built with DYNAMICXCF, require DR election. Since we usually code OMPROUTE to include a wildcard definition covering all DYNAMICXCF types, and that definition for OSPF_Interface defaults to a Router_Priority of 1, DR election can take place for HiperSockets as required, but it is ignored for the other types of DYNAMICXCF-generated interfaces. It is important NOT to code Router_Priority of 0, which disables DR election, on this wildcarded DYNAMICXCF interface definition because it will prevent HiperSockets from participating in an OSPF network. You should either allow Router_Priority to default with this type of OSPF_Interface definition or you should code it with a non-zero value.
3. For VIPA dynamics, the "Name" field is replaced with the assigned name once the VIPA is activated. A "Name" entry is required because you have defined the full, four-octet IP address. The subnet mask allows you to include the entire range of addresses if you used the VIPARANGE form of VIPA dynamics.
 1. If you are configuring dynamic VIPA links, you should use a full IP address for the Dynamic VIPA, or, if you have specified a VIPARANGE, use a full IP address for one of the entries in the range and then apply the subnet mask that will allow all other entries in the range to be recognized. When a full IP address is specified, a dummy link name entry is required. The dummy link name will be replaced with the real link name once the device is activated.

Static Routes with OSPF: IP Profile

```
GATEWAY
 9.82.1.107      =      TR1      2000      HOST
 9              =      TRL216DL  2000      0.255.255.128 0.82.84.0
 9.82.1.62     9.82.84.72 TRL216DL  2000      HOST

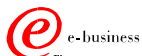
BEGINRoutes
ROUTE 9.82.1.107 HOST = TR1 mtu 2000
ROUTE 9.82.84.0/28 = TRL216DL mtu 2000
ROUTE 9.82.1.62 HOST 9.82.84.72 TRL216DL mtu 2000

ENDRoutes
```



The BEGINROUTES statement is recommended over the GATEWAY statement. Future enhancements (IPv6, etc.) will NOT support GATEWAY!

If you need static routes in your OMPROUTE environment, use the GATEWAY or V2R10 BEGINROUTES statement in your PROFILE.TCPIP and specify the AS_Boundary_Routing statement! Also specify "IMPORT_Static_Routes=YES."



© IBM Corporation 2001, 2002

1. The GATEWAY statement is used to define static routes for OMPROUTE. (In V2R10 you may also use an alternative PROFILE section called "BEGINROUTE.")
2. The GATEWAY section of the TCP/IP Profile defines routes to the network. In contrast, the OMPROUTE Configuration File defines interfaces to the network. All interfaces, regardless of the routing protocol they may use, should be coded in the OMPROUTE Configuration File.
3. This ensures that OMPROUTE does not build default interface definitions with values that may not be valid for your network. (Examples: Subnet Masks, MTU sizes, etc.)
4. If any of the static routes flow over interfaces that do not use OSPF at all, ensure that you have also coded an INTERFACE statement in the OMPROUTE configuration file for these interfaces.
5. If a route that has been configured in the GATEWAYS or BEGINROUTES section of the PROFILE is also learned by OMPROUTE, the GATEWAYS or BEGINROUTES definition takes precedence (V2R10 and earlier releases).
6. At z/OS V1R2 CS it will be possible to designate Static Routes as "replaceable."
7. BEGINROUTES is recommended for users on V2R10 or later releases. It is easier to code and read, and future enhancements such as IPv6 will not be supported on GATEWAY.
8. If you specify "Import_Static_Routes=Yes," the static routes will be imported as 5 External LSAs (Type 5 LSAs) into the ASBR and will be "broadcast" through all OSPF Areas with the exception of Stub Areas. If you wish to conceal a "Static Autonomous System" from the rest of the OSPF areas, do NOT import these routes. In this fashion the destinations defined with Static Routes will be known only to the ASBR itself.

z/OS V1R2: Replaceable Static Routes

```
BEGINRoutes
```

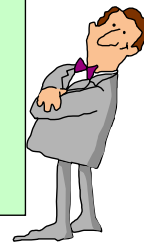
```
ROUTE 9.82.1.107 HOST = TR1 mtu 2000 REPL
ROUTE 9.82.84.0/28 = TRL216DL mtu 2000 REPL
ROUTE 9.82.1.62 HOST 9.82.84.72 TRL216DL mtu 2000 REPL
```

```
ENDRoutes
```

REPLACEABLE static route:

A new type of static route introduced in V1R2

It can be replaced by a dynamic route learned by OMPROUTE, but is remembered by the IP stack so it can be reinstated if dynamic routes are lost.



© IBM Corporation 2001, 2002

1. Replaceable static routes can only be specified on BEGINROUTES statements. GATEWAY does not support this.
2. Unlike basis static routes, OMPROUTE can replace replaceable static routes. OROUTED cannot. OMPROUTE is the strategic routing daemon, and OROUTED is to be phased out.
3. The TCP/IP stack always maintains knowledge of replaceable static routes, even when they are not being used because of being superseded by dynamic routers. When dynamic routes are lost, replaceable static routes are installed by TCP/IP. In this manner they can act as last-resort, backup routes.

Default Routes with OSPF

TCP/IP Profile at TCPA

```

BEGINRoutes
ROUTE 9.82.1.62 HOST 9.82.84.72 TRL216DL mtu 2000
ROUTE DEFAULT          9.82.1.103 TR1      mtu 2000
ENDRoutes
                    
```

OMPROUTE.CONF File at TCPA

```

AS_Boundary_Routing
...
Import_Static_Routes = YES;
                    
```

RTTABLE at TCPA

```

D TCPIP, TCPA, OMPR, RTTABLE
TYPE DEST NET MASK COST .. NEXT HOP(S)
STAT 0.0.0.0      0   0   9.82.1.103
                    
```

OMPROUTE.CONF File at TCPA

```

Default_Route
Name = TR1
Next_Hop = 9.82.1.103;
                    
```

RTTABLE at TCPA

```

D TCPIP, TCPA, OMPR, RTTABLE
EZZ7847I ROUTING TABLE 789
TYPE DEST NET MASK COST .. NEXT HOP(S)
DFLT 0.0.0.0      0   0   9.82.1.103
                    
```

OMPROUTE.CONF File at TCPA

```

AS_Boundary_Routing
...
Originate_Default_Route = YES
Default_Route_Cost=1
Default_Forwarding_Address=...;
                    
```

RTTABLE at TCPB (2 Views: Non-Stub and Stub)

```

D TCPIP, TCPB, OMPR, RTTABLE
EZZ7847I ROUTING TABLE 789
TYPE DEST NET MASK COST .. NEXT HOP(S)
SPE2 0.0.0.0      0   1   192.168.251.1
                    
```

```

TYPE DEST NET MASK COST .. NEXT HOP(S)
SPIA 0.0.0.0      0   6   192.168.251.1
                    
```

4 Default Destination automatically sent into Stub Areas by ABRs or ASBRs.

IBM Corporation 2001, 2002

1. Default routes may be learned by OSPF in several ways:

1. You may learn of a default route because you, as an AS Boundary Router, have imported a default static route that was coded in the GATEWAY section of your IP Profile. This static route was NOT learned due to the receipt of any LSA; therefore, it will not be seen in the Link State Database.
2. You may know of a default route because of an OMPROUTE Configuration File statement: "Default_Route" This default will not be present in the Link State Database.
 1. If you have coded both DEFAULT in the GATEWAY section of the PROFILE and "Default_Route" in the OMPROUTE Configuration file, the the GATEWAY coding supersedes the "Default_Route" coding.
 2. If you code more than one "Name" and "Next_Hop" pair, the last one defined will take effect.
3. Your OMPROUTE OSPF autonomous system may learn of a default route when another OSPF router sends either an External route in a Link State Advertisement (Type 5 LSA) . This default route was learned due to the receipt of an LSA; therefore, it WILL be present in the Link State Database. "SPE2" indicates that this is an "External Type 2" route.
 1. "D TCPIP, NM2aTCP, OMPR, OSPF, external" shows these Type 5 LSAs.
 2. EZZ7928I from 192.168.251.1, new LS advertisement: typ 5 id 0.0.0.0 org 10.0.0.1
4. If you are a Stub Area, your OMPROUTE OSPF autonomous system may learn of a default route when another OSPF router sends a Summary route in a Type 3 LSA. This default route was learned due to the receipt of an LSA; therefore, it WILL be present in the Link State Database. "SPIA" stands for an "inter-area destination." The default route will be sent to a Stub Area either from an ABR or from an ASBR. In either scenario, the route appears to the Stub Area as a Type of "SPIA," and its cost is governed by the value of "Stub_Default_Cost." You will see examples of this when we configure Stub Areas.
 1. D TCPIP, NM2bTCP, OMPR, OSPF, database, areaid=1.1.1.2
 2. EZZ7934I Originating LS advertisement: typ 3 id 0.0.0.0 org 10.0.0.1 (TRACE RECORD)
5. Default Route advertisements with a cost greater than an already learned default route are ignored.
6. Remember a note from a previous page:
 1. If a route that has been configured in the GATEWAYS or BEGINROUTES section of the PROFILE is also learned by OMPROUTE, the GATEWAYS or BEGINROUTES definition takes precedence (V2R10 and earlier releases).
7. The "Default_Forwarding_Address" is the address to which packets should be forwarded for the default route. If not specified, the packets follow normal OSPF behavior and are forwarded to the advertising router. If specified, they are forwarded to the provided address instead of to the advertising router. This allows OMPROUTE to advertise a default route that uses some other router, presumably one that is not running OSPF and therefore unable to advertise a default route on its own behalf.



3. Network Design for OSPF in CS for z/OS



© IBM Corporation 2001, 2002

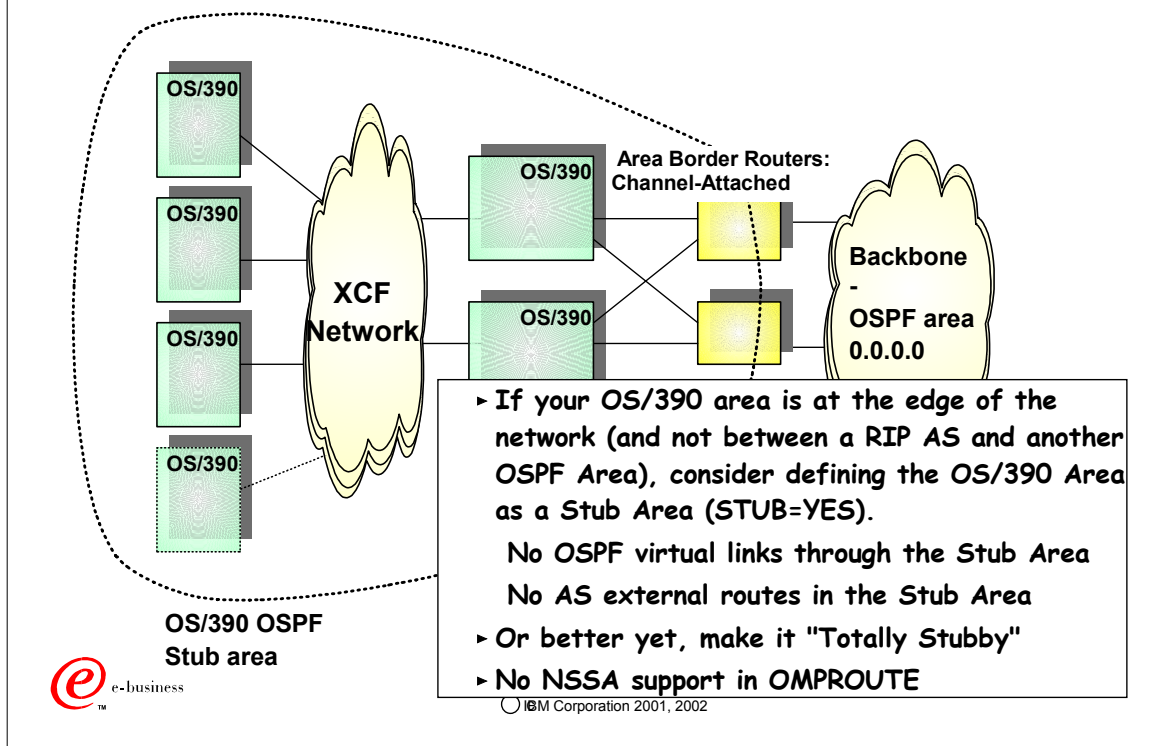


3.1. General Network Design Guidelines



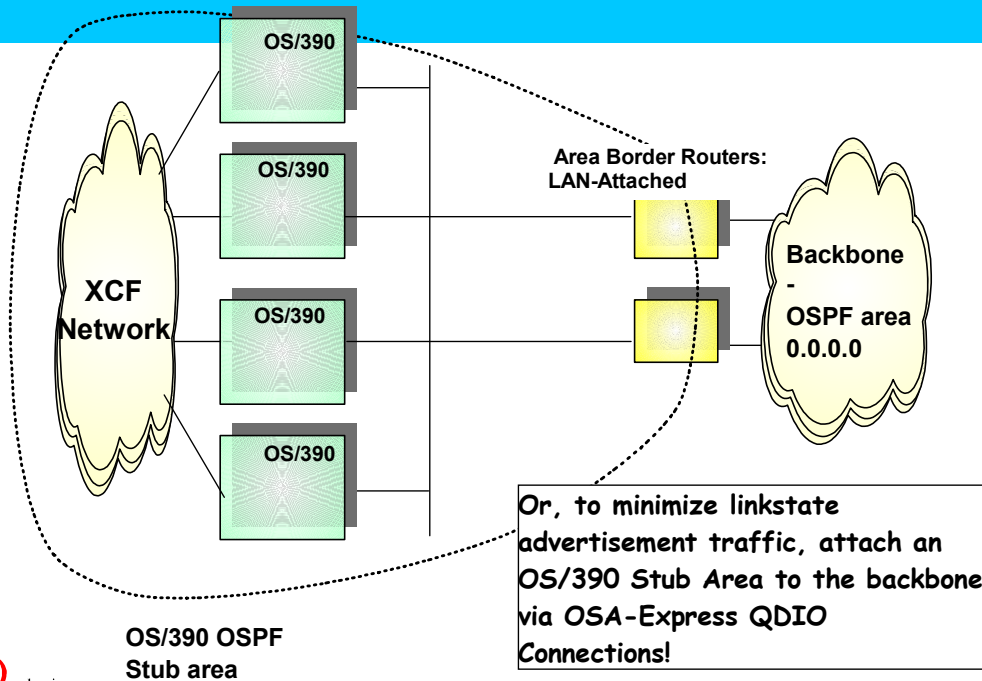
© IBM Corporation 2001, 2002

OS/390 in an OSPF Stub Area



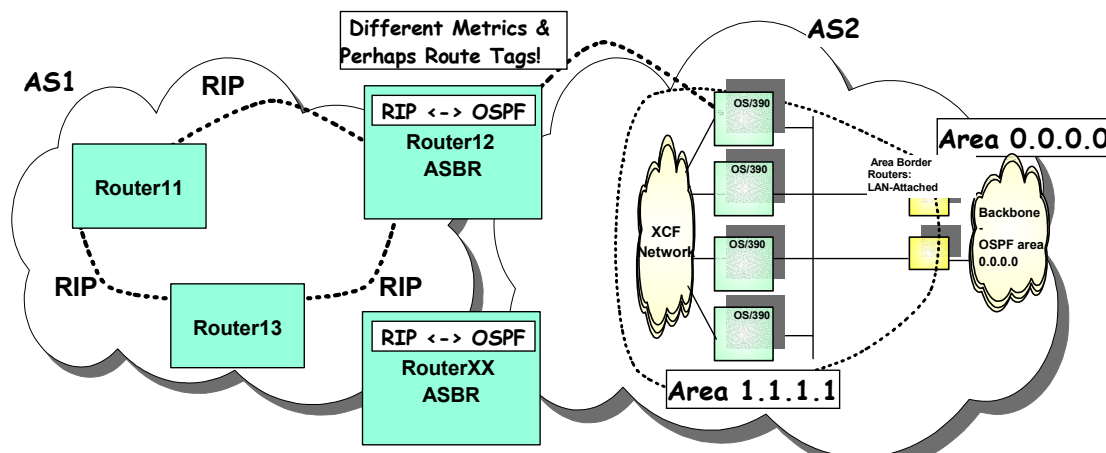
1. OSPF design options allow an installation to limit the amount of network topology that OS/390 or z/OS needs to be aware of and to process through the OSPF area concept.
2. This allows us to limit the amount of CPU required to process required OSPF information updates.
3. If your OS/390 cluster is at the edge of the network, you can reduce the amount of routing information in the OS/390 environment by defining the OS/390 sysplex as an OSPF stub area where the border routers attach the stub area to the backbone area.
 1. No OSPF virtual links through the stub area
 2. No AS external routes
4. You might even consider making CS a "Totally Stubby Area" that does not even receive Summary LSAs (except for the Default Route in a Summary LSA).
5. OMPROUTE does NOT support the "not so stubby area (NSSA)" concept described in RFC 1587
6. The following is a summary of what you may already know about different types of areas and router roles.
7. Area Border Router
 1. An area border router summarizes all OSPF links into other areas.
 2. External Link State Advertisements flow freely across all OSPF areas, except for Stub Areas..
 1. (External Links are RIP, Static, and Direct.)
 3. An Area Border Router can interface with a Stub Area.
 1. All OSPF links and summaries can flow into a Stub Area.
 2. The Area Border Router sends summary of OSPF links into the Stub Area.
 3. External Link State Advertisements cannot flow into a Stub Area.
 4. The Area Border Router says, "I'm not telling you the external links, so here's a default route for you if you need to reach an external route."
8. Stub Areas
 1. Stub Areas can have multiple default routes from different attached Area Borders. (One for each.)
 2. A Stub Area is defined with STUB=YES in the Area definition.
 3. A Stub Area can be defined as a "Totally Stubby Area" in CS for OS/390 with "Import_Summaries=NO." Only Default Routes are accepted.
 4. A stub area can be adjacent to the backbone or not adjacent to the backbone, but adjacent to a border area.
9. Connectivity to Backbone OSPF Areas and Virtual Links
 1. Every area must be adjacent to the backbone. If non-adjacent physically, there must be a virtual link to the backbone.
 2. Stub Areas can be the end-points of a virtual link, but they cannot be on the intermediate path of a virtual link.

OS/390 in an OSPF Stub Area with OSA-E



1. In the previous network, we saw point-to-point connections to channel-attached routers. Separate IP networks are used for each channel attachment, thus increasing the number of LSAs that must be advertised throughout the area.
2. In a broadcast network such as what you see on this page, fewer LSAs need to be advertised to establish adjacencies and to set up routing trees.

OS/390 in a Non-Backbone Area



- If your OS/390 area is not at the edge of the network
- You may define Area 1.1.1.1 as a non-backbone area so that it may receive RIP routes (i.e., "external routes") into the OSPF environment.
- **You must design for Multiprotocol Route Cost Comparisons!**
- With redundant ASBRs, configure for route tags to prevent routing loops.



1. Here you see that RIP AS1 is connected to the OSPF AS2 by means of an Autonomous System Boundary Router and the OS/390 area lies between the RIP AS and the Backbone Area.
2. The ASBR (Router 12) may be a CS/390 IP stack.
3. In this multiprotocol scenario, RIP routes with their metrics are imported into OSPF; OSPF routes with their Costs are imported into RIP.
4. Since metrics and costs mean different things, you must establish a route precedence methodology to determine which type of route should be preferred in the case of multiple equal-cost routes of different flavors.
5. It goes beyond the scope of this tutorial to examine these issues. For more information about coding for route precedence with the OSPF statement "Comparison," please consult the IP Configuration Guide and the Information APARs on RETAIN regarding "route precedence" and "Type 1" and "Type 2" external routes. Also the IP Configuration Guide for V2R10 and later releases has discussions of route precedence in a multiprotocol environment.
6. If you have redundant ASBRs, as you see here with Router12 and RouterXX, configure them to include route tags so that the receiving AS does not attempt to re-distribute the same routes back into the AS from which they came.

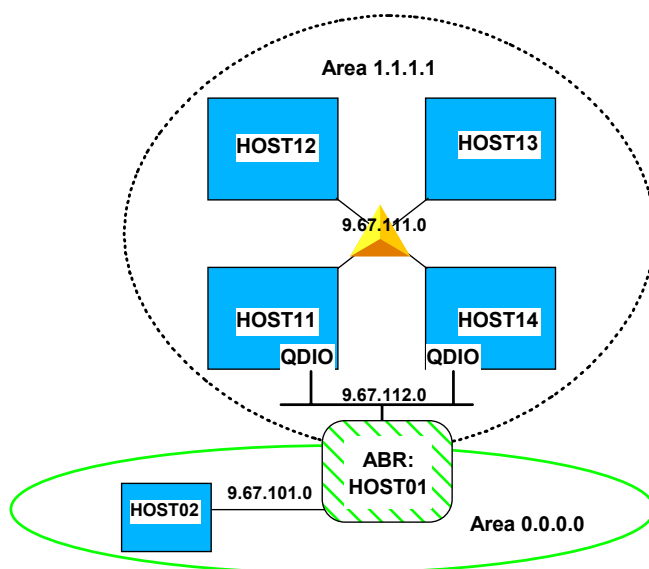


3.2. Samples for Various OSPF Network Design Options in CS

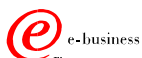


© IBM Corporation 2001, 2002

Sysplex As Non-Backbone Area: #1a



Two non-backbone Area Hosts connected only via XCF connections
 Two non-backbone Area Hosts connected also via OSA-E in QDIO mode
 One backbone interior router (HOST02) connected to ABR



© IBM Corporation 2001, 2002

1. These simple diagrams do NOT necessarily depict "best practices." For example, we often recommend the use of Private IP addresses for interior links like XCF connections, because the best practice would be to implement VIPA addresses as Public Addresses in each of the LPARs. With VIPA as our target IP addresses, there would be no need to "waste" a set of Public Addresses on the XCF connections.
2. Nevertheless, to minimize the amount of coding we need to show you in a limited space, we depict only the physical adapter addresses and not any assumed VIPA addresses.

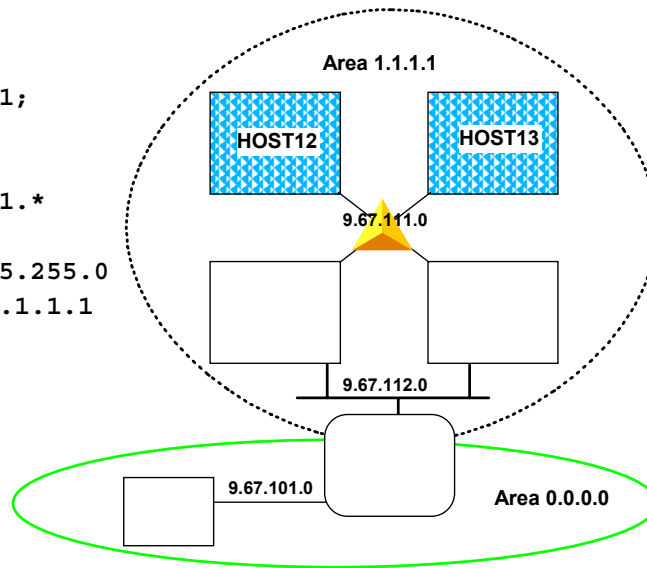
OMPROUTE Configuration: Hosts 12 & 13

```

Area
  Area_number=1.1.1.1;

OSPF_Interface
  IP_address=9.67.111.*
  Name=DYNXCF
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=55296;

```



1. This configuration file can be used on both of the hosts that are attached only to the XCF.
2. Area 1.1.1.1 needs to be defined.
3. By using a wildcard definition for the XCF interface, the OSPF_INTERFACE statement can be used unchanged on all of the hosts attached to the XCF. Note that the interface name is a dummy on wildcard interfaces
4. Because these two hosts only have one interface (the XCF) it is not necessary to define the ROUTER_ID -- it will be the address of the XCF interface since OMPROUTE picks one of the configured interfaces for router id and that is the only interface available.
5. If you follow a "best practices" scenario, you would code at least one VIPA address in each z/OS image and define it as the Router_ID.

OMROUTE Configuration: Hosts 11 & 14

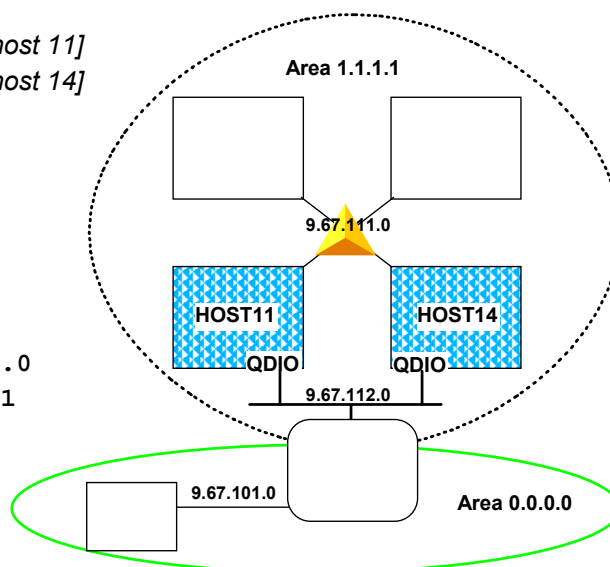
```
RouterId = 9.67.111.11; [host 11]
RouterID = 9.67.111.14; [host 14]
```

```
Area
  Area_number=1.1.1.1;
```

```
OSPF_Interface
  IP_address=9.67.111.*
  Name=DYNXCF
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=55296;
```

```
OSPF_Interface
  IP_address=9.67.112.*
  Name=OSAGBE11
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
```

 MTU=8992;



© IBM Corporation 2001, 2002

1. Because this router has two interfaces, we will explicitly define the router id so that we know what it is. To be consistent with the other XCF hosts, the XCF interface is used on as the router id in this example. Wildcard addresses are not allowed on the RouterID definition, so the same configuration file can't be used on both hosts. Except for the routerid statements, however, the configuration files are identical copies (because of the use of wildcard definitions for each interface)
2. Once again, area 1.1.1.1 must be defined.
3. The XCF interface definition is copied from the config files for hosts 12 and 13.
4. The OSA-express is configured as a Gigabit ethernet in this example

OMPROUTE Configuration: Host 01

```

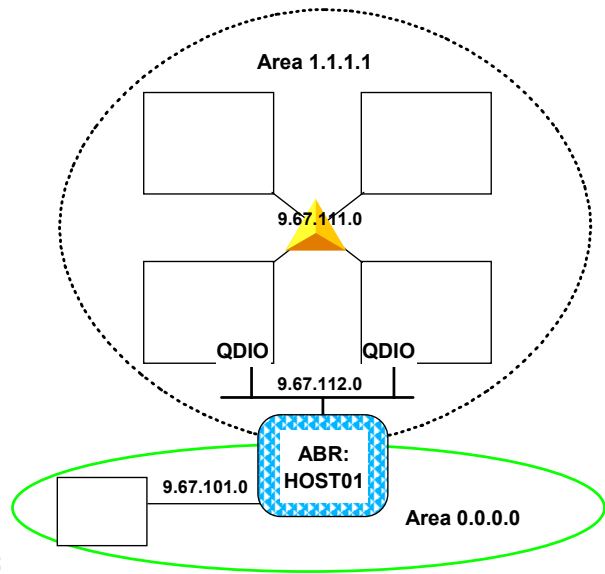
RouterID=9.67.101.1;

Area
  Area_number=1.1.1.1;

Area
  Area_number=0.0.0.0;

OSPF_Interface
  IP_address=9.67.112.*
  Name=OSAGBE11
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=8992;

OSPF_Interface
  IP_address=9.67.101.1
  Name=CTC1TO2
  Subnet_mask=255.255.255.252
  Attaches_to_area=0.0.0.0
  Destination_address=9.67.101.2
  MTU=65527;
  
```



1. This host is the area-border router between areas 0.0.0.0 (the backbone) and 1.1.1.1 (the sysplex's area). The mere presence of more than one Area statement makes this router an ABR.
2. An area is coded statement for each area
3. The OSA Express definition is copied from the other hosts on the network
4. The point-to-point link to host 2 is also defined.
5. There would probably be additional OSPF interfaces on this host in a real configuration.

OMPROUTE Configuration: Host 02

```
RouterID=9.67.101.2;
```

```
OSPF_Interface
```

```
IP_address=9.67.101.2
```

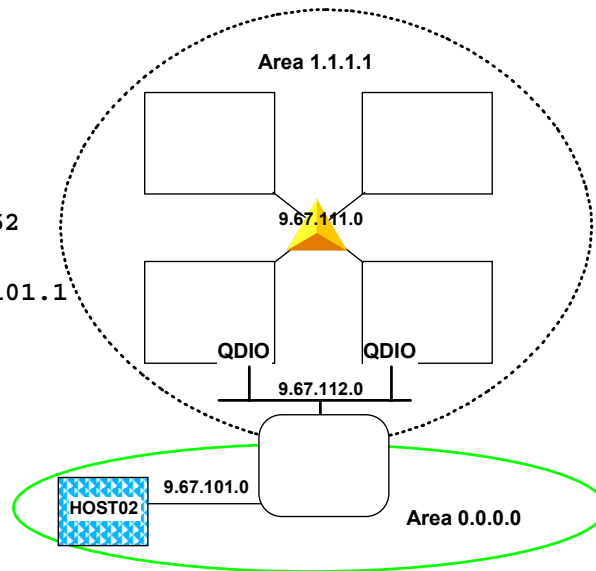
```
Name=CTC2T01
```

```
Subnet_mask=255.255.255.252
```

```
Attaches_to_area=0.0.0.0
```

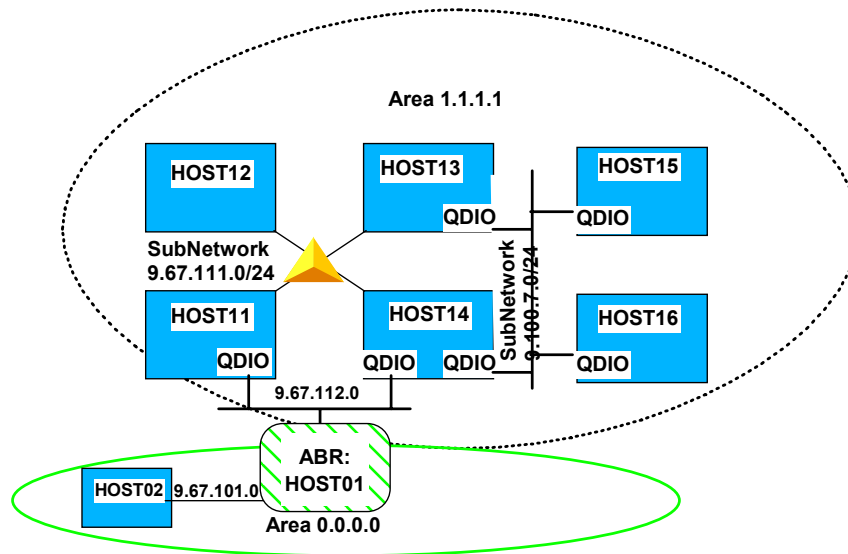
```
Destination_address=9.67.101.1
```

```
MTU=65527;
```



1. The main thing this configuration shows is that it is not necessary to define the backbone area (0.0.0.0) in routers that are in the backbone.

Sysplex As Non-Backbone Area: #1b



A separate subnet (9.100.7.0) that we wish to "hide" from the rest of the Autonomous System.



1. This example is the same as the previous one, except with the addition of the new Gigabit Ethernet subnetwork 9.100.7.0. For purposes of this example, assume that we do not want that subnetwork advertised beyond area 1.1.1.1.

OMPROUTE Configuration: Host 01

```

RouterID=9.67.101.1;

Area
  Area_number=1.1.1.1;

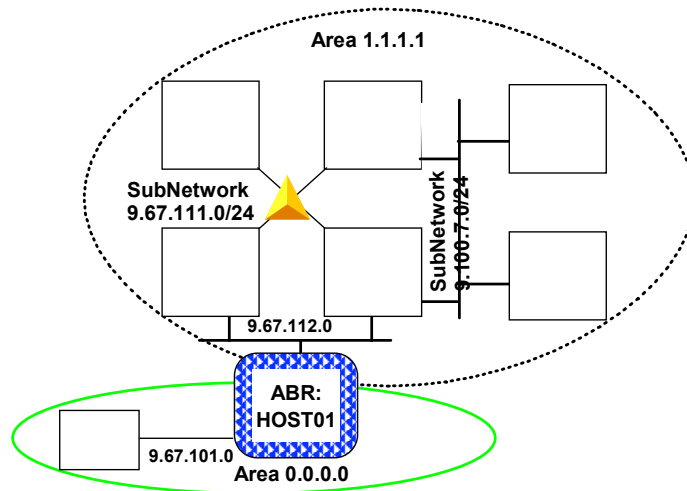
Area
  Area_number=0.0.0.0;

Range
  IP_Address=9.100.7.0
  Subnet_mask=255.255.255.0
  Area_number=1.1.1.1
  Advertise=No;

OSPF_Interface
  IP_address=9.67.112.*
  Name=OSAGBE11
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=8992;

OSPF_Interface
  IP_address=9.67.101.1
  Name=CTC1T02
  Subnet_mask=255.255.255.252
  Attaches_to_area=0.0.0.0
  Destination_address=9.67.101.2
  MTU=65527;

```



1. This is identical to the previous Host 01 configuration except for the addition of the Range statement.
2. In this example, we don't want the the 9.100.7.0 subnet advertised into the backbone area. We accomplish this with the Range configuration statement. First we define the range as all the addresses in the 9.100.7.0/24 subnet that exist in Area 1.1.1.1. Then we add the Advertise=No, which tells the Area-border router not to advertise this range into other areas.
3. This is the only type of "filtering by ip address" support that OMPROUTE has for OSPF.
 1. The RANGE statement may be used either for Route Aggregation or for Filtering.

OMPROUTE Configuration: Host 14

```
RouterID = 9.67.111.14;
```

```
Area
```

```
Area_number=1.1.1.1;
```

```
OSPF_Interface
```

```
IP_address=9.67.111.*
```

```
Name=DYNXCF
```

```
Subnet_mask=255.255.255.0
```

```
Attaches_to_area=1.1.1.1
```

```
MTU=55296;
```

```
OSPF_Interface
```

```
IP_address=9.100.7.*
```

```
Subnet_mask=255.255.255.0
```

```
Attaches_to_area=1.1.1.1
```

```
MTU=8992;
```

```
OSPF_Interface
```

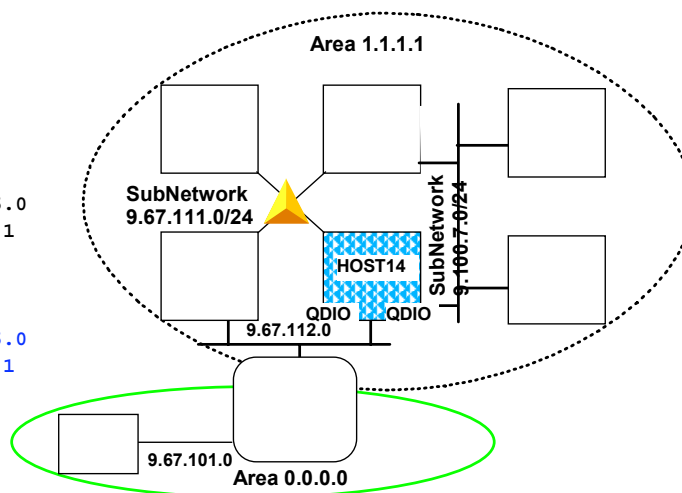
```
IP_address=9.67.112.*
```

```
Name=OSAGBE11
```

```
Subnet_mask=255.255.255.0
```

```
Attaches_to_area=1.1.1.1
```

```
MTU=8992;
```



© IBM Corporation 2001, 2002

1. The only change from the previously shown configuration for this host is the addition of the OSPF_Interface for the Gigabit Ethernet network for subnetwork 9.100.7.0.
2. We can share this Gigabit Ethernet definition with HOST13 because we have used a wildcard in its definition. Therefore, here you see two ways of defining the Gigabit Ethernet interface to OMPROUTE: one uses a wildcard with no NAME parameter; the other uses an explicit IP address with the exact Gigabit Interface Linkname.

OMPROUTE Configuration: Host 13

```
RouterID=9.67.111.13;
```

Area

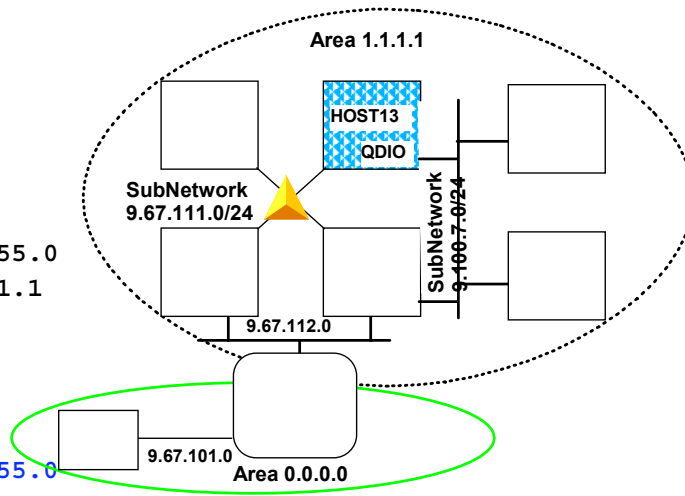
```
Area_number=1.1.1.1;
```

OSPF_Interface

```
IP_address=9.67.111.*
Name=DYNXCF
Subnet_mask=255.255.255.0
Attaches_to_area=1.1.1.1
MTU=55296;
```

OSPF_Interface

```
IP_address=9.100.7.*
Subnet_mask=255.255.255.0
Attaches_to_area=1.1.1.1
MTU=8992;
```



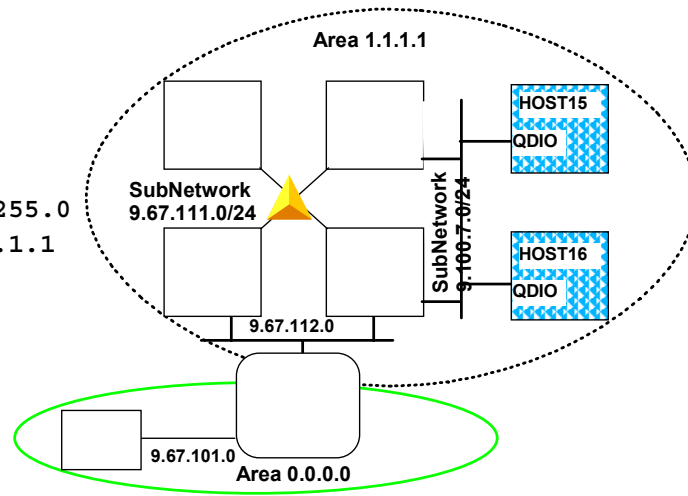
1. Because host 13 now has two interfaces, it is necessary to define the OSPF router id if we want to ensure that we get the value we want.
2. Also, the OSPF_interface for the 9.100.7.0 Gigabit Ethernet network is added. Note once again that we are sharing the same definition for this Gigabit Ethernet definition to the 9.100.7.0 network as the one we used for HOST14.

OMPROUTE Configuration: Hosts 15 & 16

```

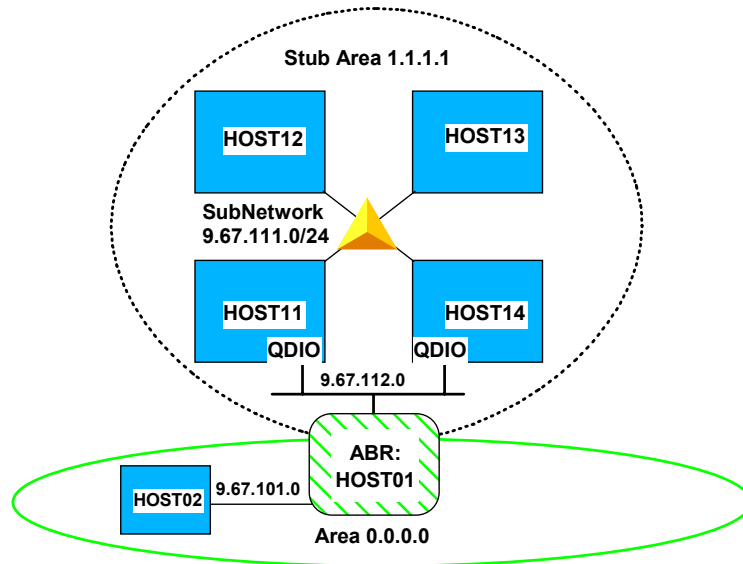
Area
  Area_number=1.1.1.1;

OSPF_Interface
  IP_address=9.100.7.*
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=8992;
  
```



1. Like hosts 12 and 13 in the previous configuration, these hosts have a relatively simple configuration for attachment via Gigabit Ethernet to the OSPF network. Each may use a wildcarded definition.

Sysplex Attached as Stub Area: #2



- Two Stub Area Hosts connected only via XCF connections
- Two Stub Area Hosts connected also via OSA-E in QDIO mode
- One backbone interior router (HOST02) connected to ABR



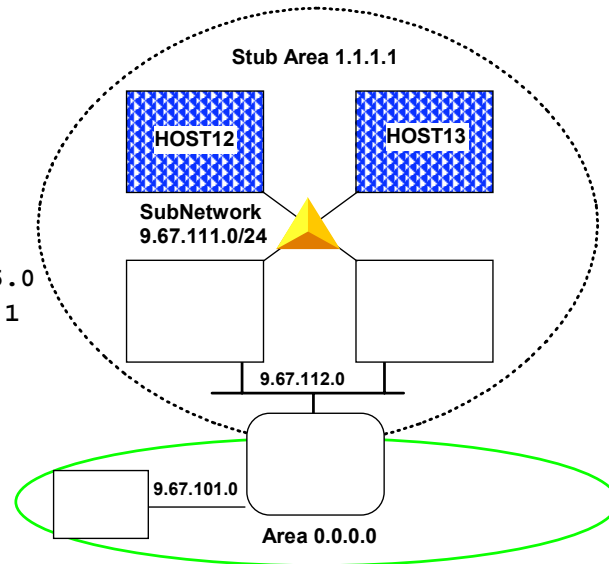
OMPROUTE Configuration: Hosts 12 & 13

Area

```
Area_number=1.1.1.1
Stub_area=yes;
```

OSPF_Interface

```
IP_address=9.67.111.*
Name=DYNXCF
Subnet_mask=255.255.255.0
Attaches_to_area=1.1.1.1
MTU=55296;
```



1. The difference between this configuration and the first configuration for these hosts is the addition of Stub=Yes to the area statement for area 1.1.1.1
2. Because 1.1.1.1 is now defined as a stub area, OSPF external routes (like static or direct routes) will not be imported into it.
3. OMPROUTE does not support Not So Stubby Area (NSSA), which is a method defined by RFC 1537 for limited importation of external routes.

OMPROUTE Configuration: Hosts 11 & 14

```
RouterId = 9.67.111.11; [host 11]
RouterID = 9.67.111.14; [host 14]
```

Area

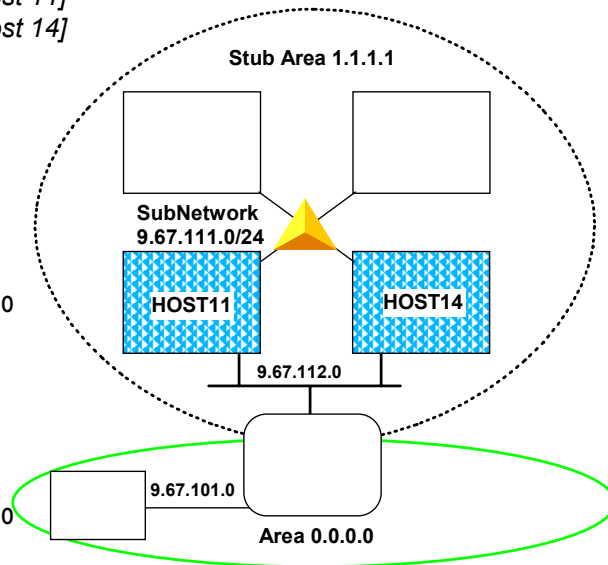
```
Area_number=1.1.1.1
Stub_area=yes;
```

OSPF_Interface

```
IP_address=9.67.111.*
Name=DYNXCF
Subnet_mask=255.255.255.0
Attaches_to_area=1.1.1.1
MTU=55296;
```

OSPF_Interface

```
IP_address=9.67.112.*
Name=OSAGBE11
Subnet_mask=255.255.255.0
Attaches_to_area=1.1.1.1
MTU=8992;
```



1. This configuration is identical to the first configuration for these two hosts except for the addition of Stub=yes on the area statement for 1.1.1.1

OMPROUTE Configuration: Host 01

```

RouterID=9.67.101.1;

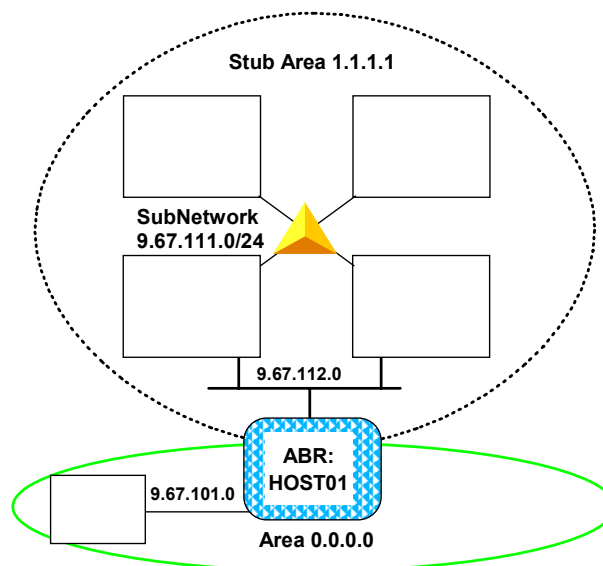
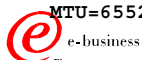
Area
  Area_number=1.1.1.1
  Stub_area=yes
  Stub_default_cost=10;

Area
  Area_number=0.0.0.0;

OSPF_Interface
  IP_address=9.67.112.*
  Name=OSAGBE11
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=8992;

OSPF_Interface
  IP_address=9.67.101.1
  Name=CTC1T02
  Subnet_mask=255.255.255.252
  Attaches_to_area=0.0.0.0
  Destination_address=9.67.101.2
  MTU=65527;

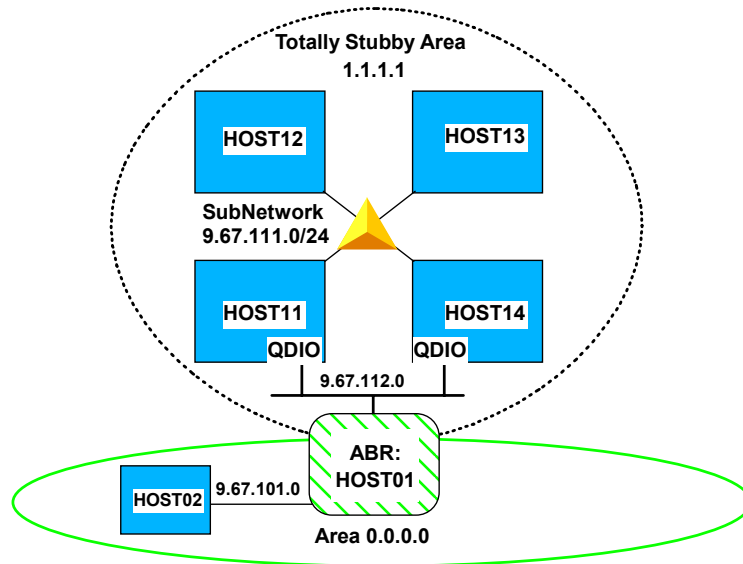
```



© IBM Corporation 2001, 2002

1. This host is the area-border router between areas 0.0.0.0 (the backbone) and 1.1.1.1 (the sysplex's stub area)
 2. An area statement for each area is required
 3. This definition is the same as the first definition for this host except for the addition of the stub area information. The area definition for a stub area at an area-border router is more interesting than it is for a router that resides wholly inside the stub area.
 4. An area border router advertises a default route into each stub area it attaches to. Routers inside the stub area will use the default route to reach OSPF external destinations via the area-border router. The `Stub_default_cost` parameter specifies the OSPF cost that will be assigned to the default route that this router advertises into the stub area.
 5. OMPROUTE will advertise summary LSAs for other areas into stub areas unless commanded not to. Because of this, a default OMPROUTE stub area is not what is called a "Totally Stubby Area." But note that it is also NOT a "Not So Stubby Area" (NSSA) as defined in RFC 1587, because it does not import type 7 LSAs for OSPF external destinations. (OSPF in OMPROUTE for OS/390 or z/OS does not support Type 7 LSAs.) Basically all the optimization you are getting in this case is that OSPF External LSAs are not propagated into the stub area, plus it can't be used as a transit network for virtual links, plus all the area-border routers advertise default routes into the stub area.
1. Assume your ABR is also an ASBR and you have coded the parameters we have seen earlier in the `AS_Boundary_Routing` statement: `"Originate_Default_Route=Yes"` and `"Default_Route_Cost=n."` Even with these parameters, the Stub Area will receive a Type 3 Summary LSA for a default route with -- NOT the `Default_Route_Cost` -- but rather the ("`Stub_Default_Cost`" plus 1). In fact, the `"Originate_Default_Route="` parameter has no effect whatsoever on the generation of the Default Route for Stub Areas. The Stub Default and its cost are derived solely from the OSPF protocol governing such exchanges and by the coding (or default setting) of `"Stub_Default_Cost."`

Sysplex As Totally Stubby Area: #3



- Two Totally Stubby Area Hosts connected only via XCF connections
- Two Totally Stubby Area Hosts connected also via OSA-E in QDIO mode
- One backbone interior router (HOST02) connected to ABR



OMPROUTE Configuration: Host 01

```

RouterID=9.67.101.1;

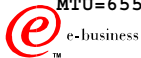
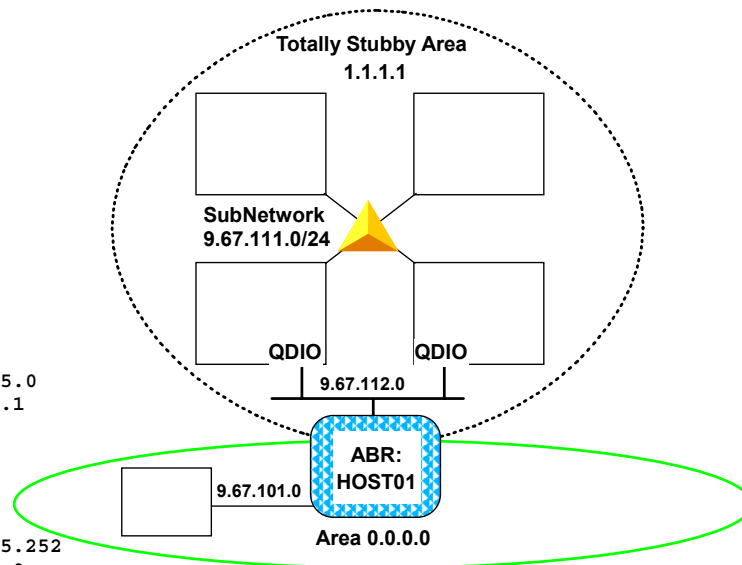
Area
  Area_number=1.1.1.1
  Stub_area=yes
  Stub_default_cost=10
  Import_summaries=no;

Area
  Area_number=0.0.0.0;

OSPF_Interface
  IP_address=9.67.112.*
  Name=OSAGBE11
  Subnet_mask=255.255.255.0
  Attaches_to_area=1.1.1.1
  MTU=8992;

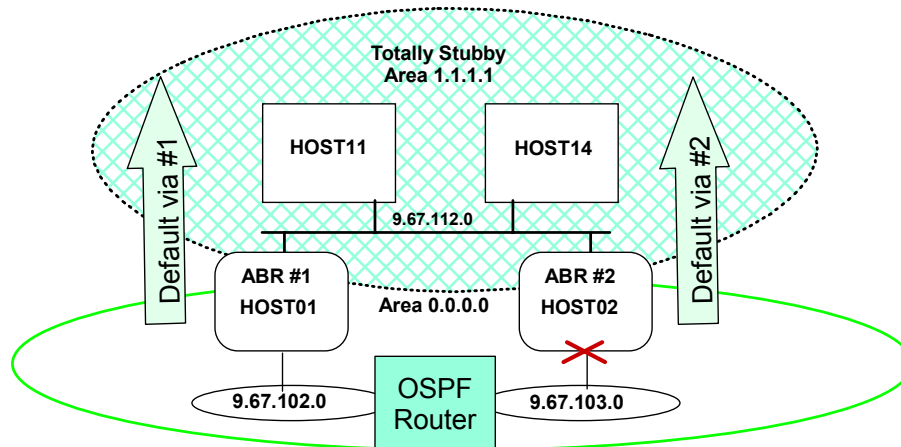
OSPF_Interface
  IP_address=9.67.101.1
  Name=CTC1T02
  Subnet_mask=255.255.255.252
  Attaches_to_area=0.0.0.0
  Destination_address=9.67.101.2
  MTU=65527;

```



1. By adding `import_summaries=no` to the area statement for stub area 1.1.1.1 at the area-border router, we have changed it into a true OSPF Stub area (totally stubby area).
2. Now the ONLY route that is not wholly contained within the stub area that this area-border router will advertise into the stub area is the default route. This can be a significant performance optimization for hosts in the stub area and will greatly simplify their routing tables.
3. The drawback is that if there is more than one area-border router serving the stub area, they will only advertise default routes into the stub area, so hosts inside the totally stubby area cannot pick an area-border router based on destination address. In other words, if there are multiple area-border routers serving a totally stubby area, they must be redundant connections.

How to Interconnect ABRs for Totally Stubby Area? Ex. 1

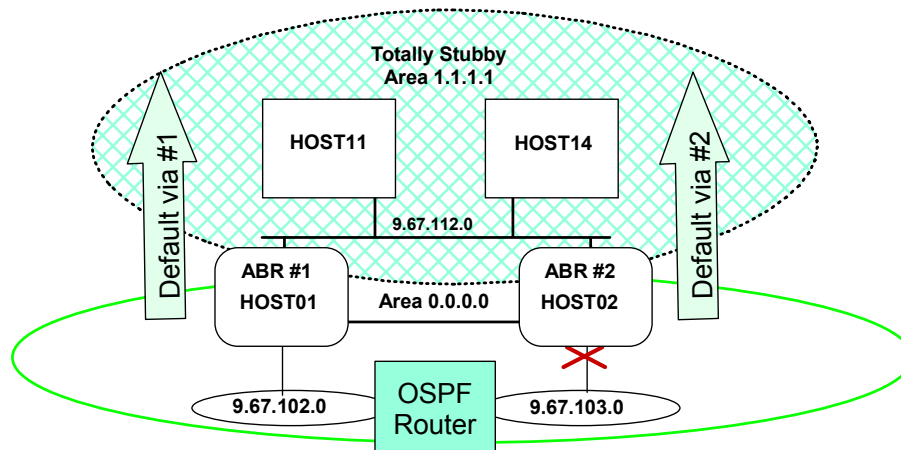


- ABRs interconnected via Link in Totally Stubby Area; two Default Gateways for each HOSTnn.
- LSA about loss of connectivity to subnet 9.67.103.0 via ABR #2 is **NOT** sent into Totally Stubby Area 1.1.1.1 by either ABR #1 or ABR #2, since only default routes are sent into Totally Stubby Areas.
- As far as hosts in the Totally Stubby Area know, all destinations outside the area can be reached via either ABR. They will continue sending packets for 9.67.103.0 to both ABR #1 and/or ABR#2 (depending on multipath settings). The packets sent via ABR #2 will not be delivered
- This is an undesirable way to configure a Totally Stubby Area. It violates the rule that it should never matter which ABR is used to reach any destination outside the area.



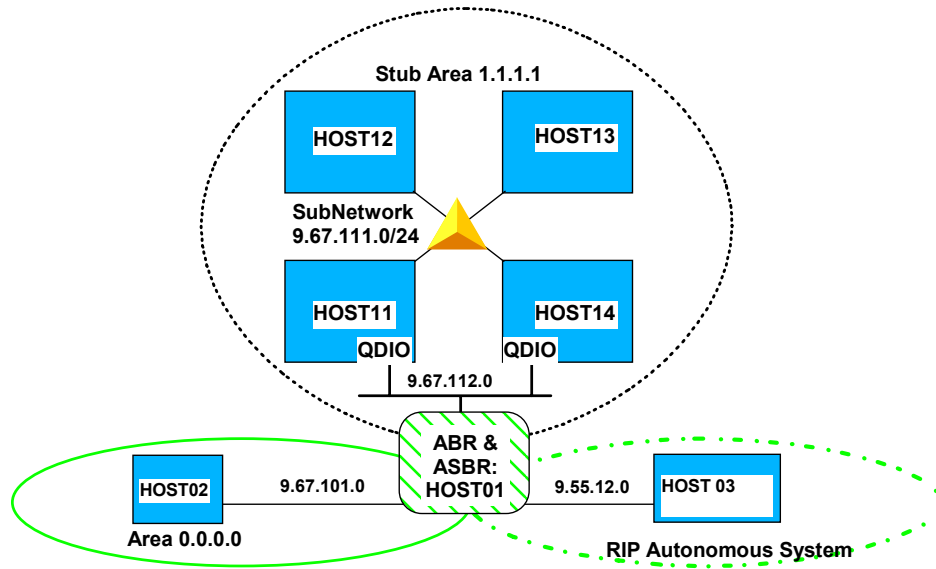
1. One ABR in a stub area does not accept the summary default route from another ABR in the same stub area. In other words, ABR #2 will not accept the stub area summary default route from ABR #1, so packets sent to the backbone via ABR #2 cannot be delivered by going to ABR #1 through the totally stubby area
2. Dead gateway processing may kick in when packets fail to be delivered via ABR #2, but only in some cases, and its recovery is not robust enough to give the redundancy that should be expected when two ABRs attach a totally stubby area to the backbone.

How to Interconnect ABRs for Totally Stubby Area? Ex. 2



- ABRs interconnected via Link in Backbone Area 0 AND Link in Area 1; two Default Gateways for each HOST1n.
 - As before, LSA about loss of connectivity to subnet 9.67.103.0 via ABR #2 is **NOT** sent into Totally Stubby Area 1.1.1.1 by either ABR #1 or ABR #2 since only default routes are sent into Totally Stubby Areas
 - Packets sent to ABR#2 for 9.67.103.0 will still reach the destination, via ABR#1 and the link in Area 0
 - HOST11 and HOST14 know of two default gateways; regardless of which gateway they choose (they have coded IPCONFIG MULTIPATH), the packets can arrive at 9.67.103.0

Sysplex Attached to ASBR/ABR #4



ABR is now also an ASBR attached to the backbone and to the RIP autonomous system.

OMPROUTE Configuration: Host 01

```
RouterID=9.67.101.1;
```

```
AS_Boundary_Routing
  Import_RIP_Routes=Yes
  Originate_Default_Route=Yes;
```

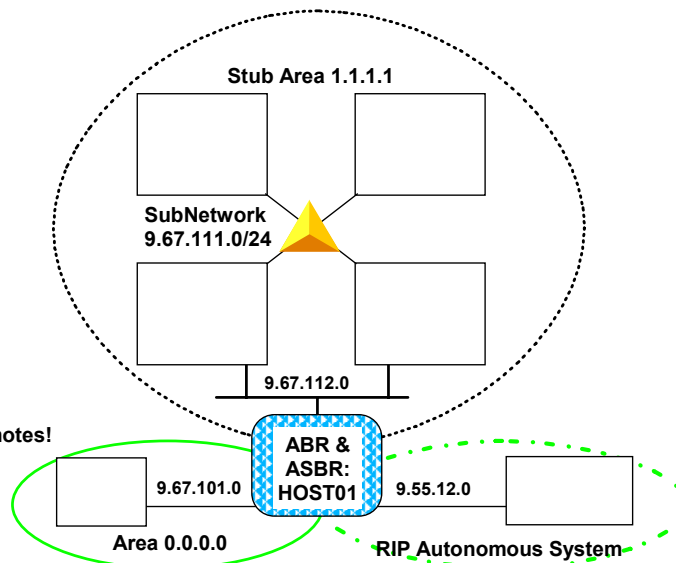
```
Area
  Area_number=1.1.1.1
  Stub_area=yes
  Stub_default_cost=10;
```

```
Area
  Area_number=0.0.0.0;
```

[OSPF Interfaces elided]

```
RIP_interface
  IP_address=9.55.12.1
  Name=CTC1TO3
  Subnet_mask=255.255.255.0
  RIPV2=YES
  [send_only=default] <== see notes!
  MTU=10000;
```

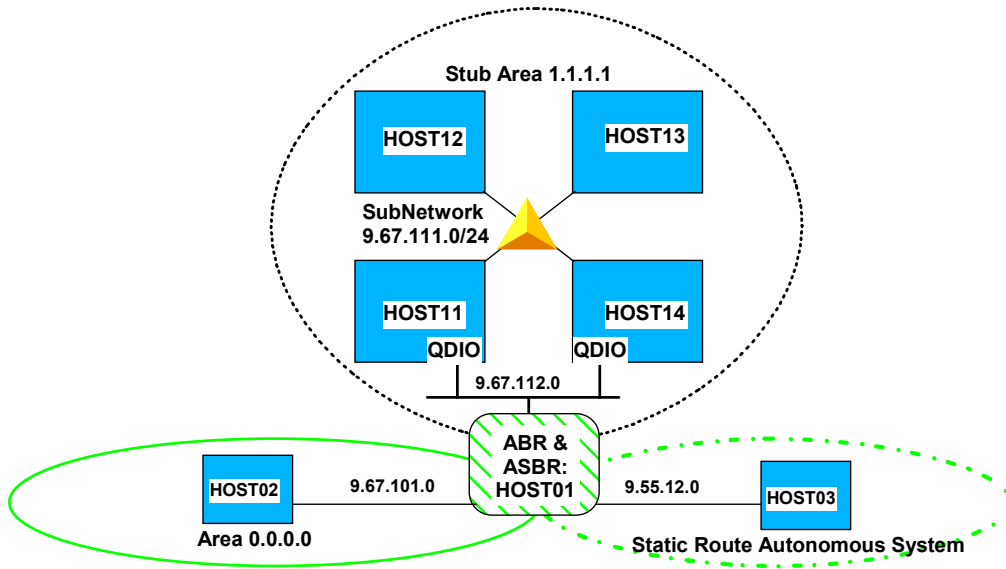
```
Originate_RIP_default
  condition=always;
```



© IBM Corporation 2001, 2002

1. This host is the area-border router between areas 0.0.0.0 (the backbone) and 1.1.1.1 (the sysplex's stub area). It is now also an Autonomous System Boundary router (ASBR) between the RIP and OSPF AS's
2. The AS_Boundary_Routing statement is an OSPF statement. It tells this router to import RIP routes into the OSPF AS, and to originate an OSPF default route into the OSPF AS. This OSPF statement does NOT affect what OMPROUTE advertises into the RIP AS, that is defined by RIP definition statements.
3. A RIP_interface is defined for the interface that connects to the RIP AS
4. the originate_RIP_default statement tells OMPROUTE to always advertise a default route into the RIP AS.
5. So this router advertises default routes into all networks to which it is attached. This is a pretty important router!
6. If this host is the only ASBR for the RIP subnet and you don't want to flood your RIP network with the contents of the OSPF network (which could get quite large), you can add the send_only=default parameter to your RIP_interface statement. This will prevent this host from importing any routes into the RIP AS except for the default route. As a result, all hosts in the RIP AS will route to this host for destinations not known within the RIP AS. This is NOT advised if this host is a needed router **within** the RIP AS (i.e., if any RIP hosts depend on this host to route to other hosts within the same RIP AS).

Sysplex Attached to ASBR/ABR #5



ABR is now also an ASBR attached to backbone and to autonomous system of Static Routes.

OMPROUTE Configuration: Host 01

```
RouterID=9.67.101.1;
```

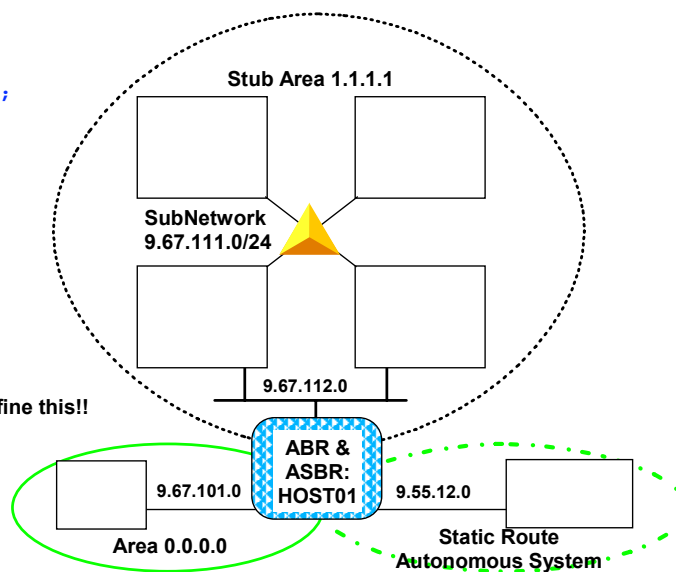
```
AS_Boundary_Routing
  Import_Static_Routes=yes
  Import_Direct_Routes=Yes
  Originate_Default_Route=Yes;
```

```
Area
  Area_number=1.1.1.1
  Stub_area=yes
  Stub_default_cost=10;
```

```
Area
  Area_number=0.0.0.0;
```

[OSPF Interfaces elided]

```
Interface <=== don't forget to define this!!
  (see notes)
  IP_address=9.55.12.1
  Name=CTC1TO3
  Subnet_mask=255.255.255.0
  MTU=10000;
```



© IBM Corporation 2001, 2002

1. This host is the area-border router between areas 0.0.0.0 (the backbone) and 1.1.1.1 (the sysplex's stub area). It is now also an Autonomous System Boundary router (ASBR) between the OSPF AS and an AS that is using static routing, meaning that any routes needed to reach destinations in the static AS are defined in the TCP/IP profile in a GATEWAY or BEGINROUTES statement. OMPROUTE is using NO routing protocol in the static AS.
2. The AS_Boundary_Routing statement is an OSPF statement. It tells this router to import static routes into the OSPF AS, and to originate an OSPF default route into the OSPF AS. It also tells OMPROUTE to import direct routes into the OSPF AS. In this case, that means that OMPROUTE will advertise a direct route to the subnet that the non-routing interface is attached to, into the OSPF network (in this example, that subnet is 9.55.12.0/24). A direct route is one that is not learned from a routing protocol, but rather from OSPF's knowledge of an Interface.
3. An "Interface" statement is defined for the interface that connects to the static AS. The Interface statement indicates that OMPROUTE is not communicating **any** routing protocol over that interface.
4. **Please note** that even though OMPROUTE is not communicating a routing protocol over the CTC1TO3 interface, it should still be defined to OMPROUTE, using the Interface statement. If it is not, then OMPROUTE will use a default value for its MTU size (576) AND update the stack's MTU value with that value. Also, if the interface is not defined to OMPROUTE, OMPROUTE will use the class mask for the interface's subnet mask (in this case 255.0.0.0), with possible undesirable results (for example, a direct route will be to 9.0.0.0 over that interface will be advertised and added to the TCP/IP routing table!)



3.3. Summary of Coding Hints



© IBM Corporation 2001, 2002

OSPF/OMPRoute Hints and Tips (1)

- > Specify all interfaces in the OMPROUTE configuration file
- > Use link metrics (weights) that make sense
- > Always specify MTU size and Subnet Mask
- > Always specify the RouterID to avoid the accidental assignment to a Dynamic VIPA (Recommended: Static VIPA)
- > If possible on multiaccess networks, let nodes other than CS be the Designated Router (Set OMPROUTE Router_Priority=0); EXCEPTION on HiperSockets LAN
- > OSPF timer defaults are normally OK, but at the very least match the HELLO and Dead Router Interval with adjacent routers
- > Run OMPRoute in a good performance group/WLM policy; make non-swappable only if necessary



1. Specify all interfaces in the OMPROUTE configuration file
 1. Otherwise OMPROUTE builds "Interface" definitions - not "OSPF_Interface" or "RIP_Interface."
 2. Also, OMPROUTE will use default values for MTU and subnet mask, which may not be desirable.
2. Use link metrics (weights) that make sense
 1. Plan ahead. With a metric = 1, difficult to lower metric as link speeds get higher
3. Always specify MTU size; otherwise a default of 576 is taken. Note that OMPROUTE MTU values override BSDRoutingparms MTU values, so if you are running OMPROUTE make sure OMPROUTE has the MTU defined with the values that you want.
4. Always specify subnet value; otherwise the native class network mask is used
5. Do not allow the RouterID to default; you need to ensure that the RouterID is not assigned to a Dynamic VIPA. Although we recommend a Static VIPA as the RouterID, in fact any physical interface value is also acceptable.
6. To eliminate extraneous CPU processing/cycles, allow nodes other than the CS node to be eligible to become the Designated Router (DR). However, remember that a HiperSockets network requires a Designated Router and at least one of the LPARs participating in a HiperSockets LAN should be eligible to be a DR (Router_Priority=1).
7. OSPF timer defaults are normally OK
 1. Default time intervals:
 2. Hello Interval - 10 seconds (MUST match what is coded in all routers on same medium)
 3. Dead router interval - 40 seconds (Recommended to be at least 4x Hello interval value)
 4. Re-transmit interval - 10 seconds
8. Run OMPRoute in a good performance group/WLM policy
 1. If OMPRoute is starved for CPU
 1. It can't respond quickly to network changes
 2. Neighbors may be lost or Neighbors may not become fully adjacent
 3. At times, if you are unsuccessful at managing performance with WLM you may consider making OMPROUTE non-swappable, although this should be unnecessary if your WLM performance groups have been properly established.
 1. To make OMPROUTE non-swappable, start it as a started task rather than via BPXBATCH; this allows you to add OMPROUTE to the PPT table as non-swappable.
9. OMPROUTE JCL may need to be revised to use an LE Environment variable to free storage in certain cases. The symptoms are loss of neighbors for DYNAMICXCF connections of all flavors: Message BPXF024I (TCPIP)X Feb 25 22:42:51 omproute 67108971 : EZZ7921I OSPF 194 Adjacency failure, neighbor 192.168.5.171, old state 32, new state 8, event 7
 1. Change the EXEC statement in the OMPROUTE JCL to use the "HEAP(,,FREE)" setting:
 1. //OMPRROUTE EXEC PGM=OMPRROUTE,REGION=0K,TIME=NOLIMIT,
 2. // PARM=(' POSIX (ON) ',
 3. // ' ENVAR (" _CEE _ENVFILE=DD:STDENV" , ',
 4. // ' " _CEE RUNOPTS=HEAP(,,FREE) ") ',
 5. // ' "TZ=EST"/ -t1')

OSPF/OMPRRoute Hints and Tips (2)

- Be careful if mixing OSPF AS with AS of different protocol - metrics may not be consistent
 Recommended reading: *Step 9: Defining Route Precedence in a MultiProtocol Environment (If OSPF Protocol is Used)* in OMPROUTE section of IP Configuration Guide (V2R10 and later)
- Some routers on a Token-Ring network do not understand the MAC multicast address used by OSPF and RIP V2. Use Translate Statement in the PROFILE:
 TRANSLATE 224.0.0.0 IBMTR FFFFFFFFFFFFFFFF linkname
- Consult the Informational APAR on OMPROUTE Problems: II12026.
- Consult APAR PQ25823 if you are on V2R7 or earlier (now unsupported releases) and are planning to use the same IP Subnet for the VIPA on several stacks AT V2R6 AND V2R7. This APAR allows OMPROUTE to send host routes for VIPA.



1. Be careful if mixing OSPF AS with AS of different protocol - metrics may not be consistent. In V2R10 we significantly enhanced the routing information in our pubs, including a discussion of this consideration.
2. Some routers on a Token-Ring network do not understand the MAC multicast address used by OSPF and RIP V2. Please consult the TCP/IP Hints and Tips FLASH (N3196) at www.ibm.com/support/techdocs or the IP Configuration Guide for a special use of the TRANSLATE statement in the PROFILE if token-ring-attached routers are not listening for the TR multicast mac address of 0xC000.0004.000.
 1. TRANSLATE 224.0.0.0 IBMTR FFFFFFFFFFFFFFFF linkname
3. If you are counting on VIPA Host Routes on any release prior to V2R8, ensure that you have consulted APARs PQ25823 and PQ29859. Without these APARS only subnet destinations will be sent.
4. Periodically consult the Informational APAR II12026 for information on avoiding common OMPROUTE problems.

Neighbor Coding for OSPF

- Link State Advertisements originated by both sides of a connection must agree on the media type and connection type.
- Some idiosyncrasies ...
 - Cisco codes CLAW as "point-to-multipoint" and "broadcast" in older releases of Cisco software. Higher releases allow coding of "point-to-point."
 - Ensure that IBM side of CLAW connection is compatible with what is coded in the Cisco. You may want to use new "P2MP" keyword on the LINK statement in the TCP/IP stack.
 - Consult RFC 2328, APAR PQ35265 and APAR PQ48766, and Flash N3196 for V2R8 IP Migration Hints and Tips.
 - MPC+ can be perceived as either P2P or P2MP by the TCP/IP stack in Communications Server; the coding on the Cisco side of an MPC+ connection is the same in either case.



1. See the CISCO Examples in the APPENDIX.
2. CS/390 OSPF perceives a CLAW interface as point-to-point and a CISCO router codes it as point-to-multipoint. Note that Cisco does not have a configuration parameter for P2P for CLAW; it must be configured using the P2MP *AND* BROADCAST parameters to treat CLAW like P2P for compatibility with the IBM CLAW.
 1. For example, a CISCO router considers a broadcast network to be point-to-multipoint; a non-broadcast network is point-to-point. Accordingly the CISCO router will send in Link State Advertisements according to RFC 2328, section 12.4. If you have problems connecting properly or receiving appropriate route updates, you may have a mismatch in the configurations between CS/390 and the router.
 2. APARs PQ35265 and PQ48766 describe how to correct these scenarios. Please note new CLAW syntax for LINK in the IBM stack: P2MP. See examples in the Appendix of the presentation.



4. Diagnosing OSPF in CS



© IBM Corporation 2001, 2002



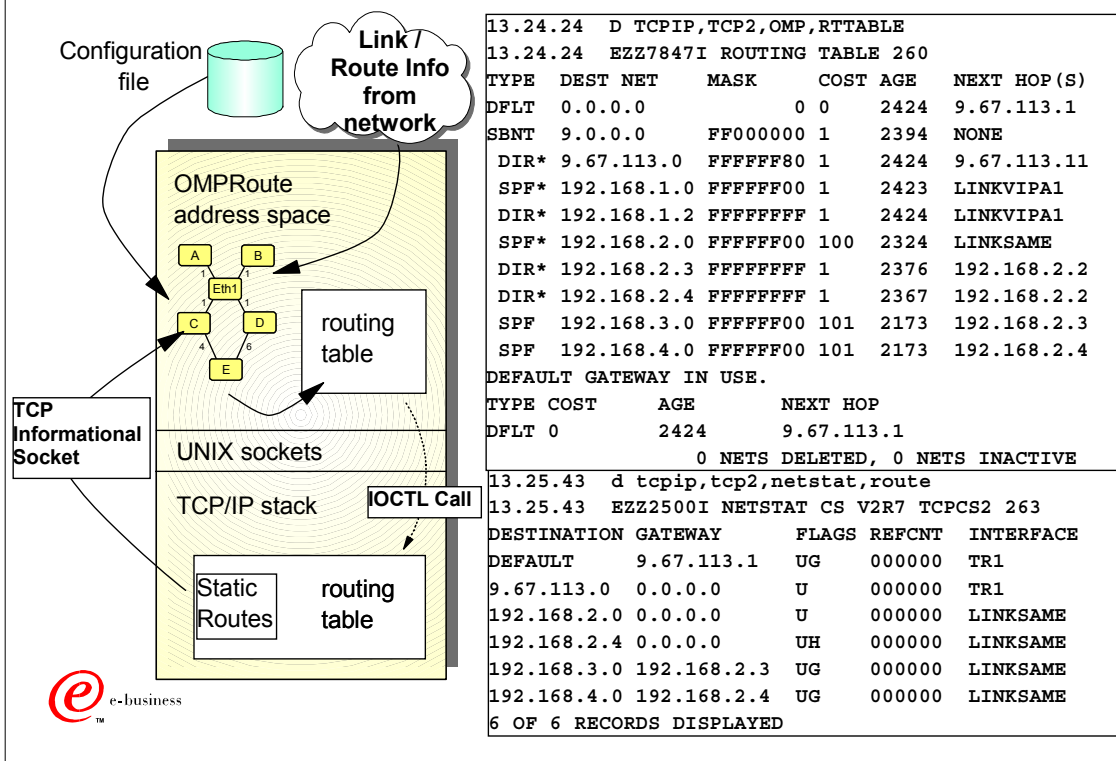
4.1. Diagnosis: Understanding Basic OMPROUTE Structure in CS



© IBM Corporation 2001, 2002

1. OSPF is a routing protocol that requires multicast links.
2. In OS/390, OSPF has been made available as a protocol that can be configured in a server named OMPROUTE.
 1. OMPROUTE supports OSPF interfaces and RIP interfaces.
3. You may combine static routes with the PROFILE GATEWAY or BEGINROUTES statement and OMPROUTE.
4. You may also code for VIPA with OMPROUTE.
5. OMPROUTE implements a multipath concept, whereby there is traffic distribution over multiple routes to the network.
 1. Multipath was introduced in IP of V2R6 of OS/390.
 2. Multipath was enhanced in V2R8 with a more robust distribution algorithm.

OMPROUTE Structure



1. OMPROUTE maintains a routing table in its address space (RTTABLE); it receives information for this table from OSPF LSAs and from the TCP/IP stack via a TCP Informational Socket. Via an IOCTL call, Omproute sends its routing information to the TCP/IP stack, which makes the actual routing decisions.
2. OSPF support on CS for OS/390 has same code base as IBM 2216/2210
3. VM/TCPIP also has ported this code, their daemon is named MPROUTE. It was ported from V2R6 OMPROUTE.
 1. Routing Daemon: OMPRoute, standing for OpenMVS MultiProtocol Router
 1. Can run RIP on some interfaces, OSPF on others
 2. Receives information from the IP stack via a TCP Informational Socket. (This socket is visible with D TCPIP,,N,CONN.)
 3. Maintains its own routing table and feeds information to stack table via an IOCTL program call. Stack makes routing decision.
 4. Presents stack with multiple equal-cost routes so that stack can distribute Traffic Among Equal-Cost Routes
 1. IPCONFIG MULTIPATH
 2. Both RFC 2328 and RFC 1583 define OSPF. However, CS for OS/390 OMPROUTE supports 1583+ and not RFC2328.
 3. OSPF uses raw sockets; unlike UDP and TCP, the concept of Port number is unknown to raw sockets. Therefore, no port number is coded for OSPF in the IP Profile.
4. Has Link State Database of network; computes from it a routing table. Updates the IP stack routing table with this one.
5. Can discover/React to Failures More Quickly
6. Cannot run ORouteD and OMPRoute on same stack
7. Extends Addressing
8. Varying-Sized Subnets
9. Implements SNMP Subagent
10. The TYPE field of RTTABLE may have many values, some of which we will see later in a more complete discussion of this table. :
 1. SBNT = placeholder to indicate net has been subnetted
 2. DIR = direct
 3. RIP = Learned through RIP
 4. SPF = Intra-Area destination
 5. SPIA = Inter-Area destination
 6. SPE1 = External Type 1 Route
 7. SPE2 = External Type 2 Route
 8. RNGE = if ranges have been set
 9. * = router has directly connected backup
 10. % = RIP updates unconditionally accepted for this net/subnet
 11. Next Hop = IP Address or link name

OSPF & Multipath

OSPF Multipath support : OSPF enables the stack to perform ...

Load balancing over multiple equal weight routes

Multipath routes chosen in a round-robin scheme

Inactive multipath routes are skipped

Outbound bind-specific connections and directed IP traffic over specified interfaces not affected

Including new versions of OPING and OTRACERT

Per Destination Multipath support in R6 only

Load balancing to destination hosts

One route per source and destination address pair

Activated by MULTIPATH option in config file

Per Connection Multipath support in R8+

Load balancing for multiple routes between same source and destination address pair

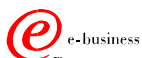
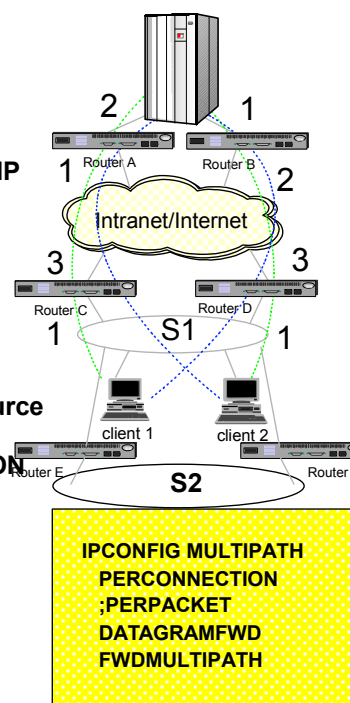
One route per TCP/IP connection: **PERCONNECTION**

Per Packet Multipath support in R8+

Load balancing of IP traffic at the packet level

New route chosen per IP packet: **PERPACKET**

OMPROUTE supports up to 4 multipath routes to a destination



IBM Corporation 2001, 2002

- OSPF enables the stack to perform Multipath forwarding by presenting the stack with multiple routes to the same destination. However, the stack must have been coded to support MULTIPATH to be able to take advantage of the multiple routes with which it has been presented.
- Multipath with V2R6:
 - Perdestination
 - Assume destination is Client 1 in S1 network. IP will choose a path(Router A?) for that Client 1 and all connections to that client flow over Router A. Client 2's traffic may flow over Router B.
- Multipath with V2R8:
 - Perconnection:
 - The TELNET connection to Client1 passes over RouterA. The FTP connection to Client1 passes over RouterB.
 - The TELNET connection to Client2 passes over RouterB. The FTP connection to Client2 passes over RouterA.
 - PERCONNECTION is the DEFAULT.
 - Perpacket:
 - The packets using the same destination and source IP addresses will use different outbound paths. IP will select a route on a round-robin basis from a multipath routing list to that destination host.
- DATAGRAMFWD allows IP to be an intermediate router and transfer data between networks
 - If you add FWDMULTIPATH, then multipath logic will be applied to the transferred data.
 - NOFWDMULTIPATH is the DEFAULT.
- If you have multiple OSPF interfaces on the same subnet, all are used for traffic, but only one is used for OSPF packets. (Per RFC 2178)
 - You may control which interface becomes primary with the use of the PARALLEL_OSPF parameter on the OSPF_Interface statement. Otherwise, OSPF chooses the primary, which sends all OSPF packets.
 - One interface selected as "primary"
 - If primary fails, backup interface becomes new primary
- OMPROUTE supports up to 4 multipaths to the same destination. If you require more, you'll have to use static routing, which supports unlimited multipath. See slide 79 for more information on this limitation.



4.2. Diagnosis: Commands to Display OMPROUTE Environment



© IBM Corporation 2001, 2002

Interpreting Routing Table Displays

- **The next two charts illustrate how to interpret routing table displays.**
 - **First the output from a NETSTAT ROUTE command is shown.**
 - **Then the output from a display of OMPROUTE's internal routing table on the same configuration is shown.**
 - **These displays were performed on a test system that was configured for illustrative purposes and does not necessarily represent recommended configurations or practices!**



1. Recall that OMPROUTE and the stack each keep their own routing table. OMPROUTE keeps the stack's routing table updated with dynamically learned routes, and the stack keeps OMPROUTE's routing table updated with static routes.
2. The stack's routing table is the only one used to z/OS to make routing decisions on the data path. So OMPROUTE's responsibility stops with getting the correct routes into the stack's routing table.

d tcpip,tcps7,n,route

```

EZZ2500I NETSTAT CS V1R2 TCPCS7
DESTINATION GATEWAY FLAGS REFCNT INTERFACE
DEFAULT 9.67.103.6 UGR 000000 CTC7TO6 <=== Default learned via RIP
5.5.5.4 9.67.107.5 UGO 000000 CTC7TO5 <=== subnet learned via OSPF
5.5.5.5 9.67.107.5 UGHO 000000 CTC7TO5 <=== host learned via OSPF
6.6.6.4 9.67.103.6 UGR 000000 CTC7TO6 <=== subnet learned via RIP
7.7.7.7 0.0.0.0 UH 000000 VIPA1A <=== a local VIPA interface
8.8.8.8 9.67.100.8 UGHO 000000 CTC7TO8 <=== host learned via OSPF
8.8.8.8 9.67.100.8 UGO 000000 CTC7TO8 <=== subnet learned via OSPF
9.67.100.0 0.0.0.0 UC 000000 CTC7TO8 <=== subnet of an active local interface
9.67.100.7 0.0.0.0 UH 000000 CTC7TO8 <=== home addr of active local interface
9.67.100.8 0.0.0.0 UHO 000000 CTC7TO8 <=== dest addr of active local interface
[some routes elided to save space]
9.67.102.7 0.0.0.0 H 000000 CTC7TO3 <=== home addr. of inactive local interface
[some routes elided to save space]
9.67.111.0 0.0.0.0 UC 000000 EZASAMEMVS <= subnet of active local interface
9.67.111.7 0.0.0.0 UH 000000 EZASAMEMVS <= home addr of active local interface
9.67.111.8 0.0.0.0 UHS 000000 EZASAMEMVS <= static route
30.1.1.0 9.67.103.6 UGR 000000 CTC7TO6 <=== subnet learned via RIP
30.1.1.4 9.67.103.6 UGHR 000000 CTC7TO6 <=== host learned via RIP
30.1.1.8 9.67.103.6 UGHR 000000 CTC7TO6 <=== host learned via RIP
127.0.0.1 0.0.0.0 UH 000004 LOOPBACK <=== home addr of local interface
130.200.0.0 9.67.100.8 UGO 000000 CTC7TO8 <=== subnet learned via OSPF
130.200.1.18 9.67.100.8 UGHO 000000 CTC7TO8 <=== host learned via OSPF
130.201.0.0 9.67.100.8 UGO 000000 CTC7TO8 <=== subnet learned via OSPF
130.202.0.0 9.67.100.8 UGO 000000 CTC7TO8 <=== host learned via OSPF
30 OF 30 RECORDS DISPLAYED

```

Note: Protocol flags (R,O,C,S, etc.) are new for z/OS v1r2!



1. This is the netstat display of the IP routing table on a host running OMPROUTE as an ASBR talking both OSPF (over interfaces CTC7TO8 and CTC7TO5) and RIP (over interface CTC7TO6) and also attached to a static network (over interface EZASAMEMVS).
2. Beginning with z/OS v1r2, the netstat display is enhanced with additional flags that indicate route protocol. These flags can help with debugging routing tables as they indicate where routes came from. The values are:
 1. O -- OSPF
 2. R -- RIP
 3. C -- other
 4. S -- static
 5. Z -- replaceable static (z/OS v1r2 enhancement)
3. Routes flagged O,R,or C are learned from OMPROUTE. Others are not. Routes with no protocol flag are generally implicit routes to local interfaces.
4. This display will tell you if a route is a host route or not (by the presence or absence of the H flag) but will not give you the subnet mask for non-host routes (but see the next slide for that information).
5. This display is the format that system administrators are used to looking at
6. If a route is in this display correctly, then OMPROUTE has done its job. If routing to the destination still fails, there may be an IP or a network problem, but there is NOT an OMPROUTE problem. If an expected route is missing from this table, there may be an OMPROUTE problem.

d tcpip,tcps7,omp,rttable

```

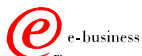
EZZ7847I ROUTING TABLE
TYPE  DEST NET          MASK      COST    AGE    NEXT HOP(S)

RIP   0.0.0.0            0 2      20     9.67.103.6 <== default route
SPF   5.5.5.4            FFFFFFFC 2 39     9.67.107.5 <== OSPF subnet route
SPF   5.5.5.5            FFFFFFFF 2 39     9.67.107.5 <== OSPF host route
RIP   6.6.6.4            FFFFFFFC 2 20     9.67.103.6 <== not a host route!
DIR*  7.7.7.4            FFFFFFFC 1 48     7.7.7.7    <== subnet of local vipa
DIR*  7.7.7.7            FFFFFFFF 1 48     VIPALA     <== local vipa
SPF   8.8.8.8            FFFFFFFC 2 32     9.67.100.8
SPF   8.8.8.8            FFFFFFFF 2 32     9.67.100.8
DIR*  9.67.100.0         FFFFFFF0 1 52     9.67.100.7 <== direct route
SPF   9.67.100.7         FFFFFFFF 2 32     CTC7TO8
SPF*  9.67.100.8         FFFFFFFF 1 42     9.67.100.8 <== OSPF route w/direct backup

[same routes as previous slide elided to save space]
DIR*  9.67.111.0         FFFFFFF0 1 51     9.67.111.7
SPF   9.67.111.7         FFFFFFFF 0 48     EZASAMEMVS
STAT* 9.67.111.8         FFFFFFFF 0 54     9.67.111.7 <== static route
RIP   30.1.1.0         FFFFFFF0 2 20     9.67.103.6
RIP % 30.1.1.4         FFFFFFFF 2 20     9.67.103.6
RIP % 30.1.1.8         FFFFFFFF 2 20     9.67.103.6
SPE2  130.200.0.0       FFFF0000 0 23     9.67.100.8 <== imported by another ASBR
SPE2  130.200.1.18     FFFFFFFF 0 23     9.67.100.8 <== imported by another ASBR
SPE2  130.201.0.0       FFFF0000 0 23     9.67.100.8 <== imported by another ASBR
SPE2  130.202.0.0       FFFF0000 0 23     9.67.100.8 <== imported by another ASBR

DEFAULT GATEWAY IN USE.
TYPE COST    AGE    NEXT HOP
RIP 2      20     9.67.103.6
0 NETS DELETED, 7 NETS INACTIVE

```



1. This is the display of the OMPROUTE routing table in the same configuration. It gives different information about routes that OMPROUTE knows about.
2. Note that implicit routes to interfaces that are not active, are not known to OMPROUTE (hence no route to 9.67.102.7)
3. Note that the TYPE column is roughly equivalent to the protocol flags on the netstat display. Since the netstat protocol flags are new for z/OS V1R2, this display is your only way of learning a route's protocol in earlier releases. Also, note that in some cases, more detail is given than on netstat. For example, OSPF routes will tell what type of OSPF (all of these types result in the O flag on netstat displays):
 1. SPF for regular intra-area OSPF
 2. SPE1 for OSPF type1 external (i.e., for example, a route that was imported by a different ASBR into OSPF, the original type could have been RIP, Static, Direct, Default, etc -- only the importing ASBR knows the original type of the imported route).
 3. SPE2 for OSPF type 2 external (same info as above, the only difference between type 1 and type 2 external is whether or not the metrics are comparable with OSPF metrics)
 4. SPIA for OSPF inter-area.
4. The following OMPROUTE route types result in the C flag on netstat displays:
 1. DIR is for "direct" routes. These are routes that are known solely because of a direct connection, for example the subnet of a directly attached interface, or the destination address of a directly attached CTC over which no routing protocol is being used
 2. DFLT is for routes defined with the DEFAULT_ROUTE statement in the OMPROUTE configuration file. Note that default routes learned dynamically are typed with the protocol that was used to learn them (RIP or OSPF) and that default routes defined using GATEWAY or BEGINROUTES are type STAT or RSTAT (RSTA is replaceable static, it applies to z/OS v1R2 or later releases)
5. Note the column after the type may contain symbols. Those symbols can be:
 1. * indicates that a static or direct route is known to the destination, possibly in addition to a dynamic route that is being used now (dynamic routes are always preferred over direct and replaceable static routes). If you have coded a DESTIP value in the OMPROUTE configuration statement, you might see this asterisk.
 2. % indicates that this is a RIP route whose advertisement was accepted because of an ACCEPT_RIP_ROUTE definition (i.e., it's a route whose advertisement may be blocked by RIP filters like accept_dynamic_hosts=no, but the user has specifically coded that it is to be accepted)
6. Note that in this display the subnet mask is given in hex. Look at it carefully. For example, the first route to 8.8.8.8 is NOT a host route!
7. The 0 NETS DELETED, 7 NETS INACTIVE is mainly used by development for internal debugging. It probably should not be shown but since we've shipped with it since R6, we can't take it away now!

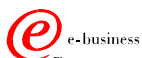
Displaying Interfaces and Neighbors

```

D TCPIP,TCPIP,OMPR,OSPF,IF
EZZ7849I INTERFACES 153
IFC ADDRESS      PHYS          ASSOC. AREA    TYPE   STATE  #NBRS  #ADJS
1.1.2.201        LNKVIPAL       0.0.0.0       VIPA   N/A    N/A    N/A
192.168.20.2    MPC49LNK       0.0.0.0       P-2-MP 16     1      1
192.168.10.2    MPC28LNK       0.0.0.0       P-2-MP 16     1      1
192.168.110.2   LNKOSAF1       0.0.0.0       BRDCST 32     3      2
192.168.100.2   GIGFE          0.0.0.0       BRDCST 32     3      2

D TCPIP,TCPIP,OMPR,OSPF,NBR
EZZ7851I NEIGHBOR SUMMARY 842
NEIGHBOR ADDR   NEIGHBOR ID    STATE  LSRXL DBSUM LSREQ HSUP  IFC
192.168.10.3    206.10.20.1   128    0     0     0    OFF MPC28LNK
192.168.110.4   206.10.20.2   128    0     0     0    OFF LNKOSAF1
192.168.110.3   206.10.20.1   128    0     0     0    OFF LNKOSAF1
192.168.100.3   206.10.20.1   128    0     0     0    OFF GIGFE
192.168.100.4   206.10.20.2   128    0     0     0    OFF GIGFE

```



1. When you first bring up OSPF with OMPROUTE, issue the Interface and the Neighbor displays that you see here. A state of 128 on the Neighbor display indicates that the connection is healthy and that you are fully adjacent.
2. Message EZZ7849I State Values
 1. 1 = down
 2. 2 = Backup
 3. 4 = looped back
 4. 8 = waiting for DR determination
 5. 16 = pt-pt
 6. 32 = DR Other
 7. 64 = BU DR
 8. 128 = DR
3. # NBRs = # of routers whose hellos have been received plus those configured
4. # ADJS = # of neighbors in state Exchange or greater; they have synchronized the DB or are in the process of synchronizing
5. Message EZZ7851I : This reflects the neighbor states that you heard about in part 1 of this 2-part presentation on OSPF (An OSPF Tutorial - The Architecture).
 1. Neighbor ID = Router ID
 2. 1 = down
 3. 2 = attempt
 4. 4 = initializing
 5. 8 = two-way
 6. 16 = Exchange Start
 7. 32 = Exchange
 8. 64 = Loading
 9. 128 = Full Adjacency
6. LSRXL = Link-State ReXMIT List
7. DBSUM = Size waiting to be sent
8. LSREQ = Size requested by neighbors
9. HSUP = Hello Suppression
- 10.

Displaying the OSPF Configuration

```

D TCPIP,TCPIP,OMPROUTE,OSPF,LIST,ALL
EZZ7831I GLOBAL CONFIGURATION 750
TRACE: 0, DEBUG: 0, SADEBUG LEVEL: 0
STACK AFFINITY:          TCPIP
OSPF PROTOCOL:          ENABLED
EXTERNAL COMPARISON:    TYPE 2
AS BOUNDARY CAPABILITY: ENABLED
IMPORT EXTERNAL ROUTES: STA DIR SUB
ORIG. DEFAULT ROUTE:   NO
DEFAULT ROUTE COST:    (1, TYPE 2)
DEFAULT FORWARD. ADDR.: 0.0.0.0
DEMAND CIRCUITS:       ENABLED

EZZ7832I AREA CONFIGURATION
AREA ID      AUTYPE      STUB?  DEFAULT-COST  IMPORT-SUMMARIES?
0.0.0.0      0=NONE          NO      N/A           N/A
0.0.0.0      0=NONE          NO      N/A           N/A

EZZ7833I INTERFACE CONFIGURATION
IP ADDRESS   AREA      COST  RTRNS  TRNSDLY  PRI  HELLO  DEAD
1.1.2.201    0.0.0.0   3     5     1        1    2      8
192.168.20.2 0.0.0.0   3     5     1        1    2      8
192.168.10.2 0.0.0.0   3     5     1        1    2      8
192.168.110.2 0.0.0.0   3     5     1        1    2      8
192.168.100.2 0.0.0.0   3     5     1        1    2      8

```



© IBM Corporation 2001, 2002

1. External Comparison: type used when importing external routes into OSPF and when comparing OSPF External routes to RIP routes
2. RTRNS = retransmission of unacknowledged routing info
3. TRNSDLY = estimated # of seconds required to transmit routing info over the interface
4. PRIORITY = Router Priority



4.3 Diagnosis: Tracing in OMPROUTE



INFO APAR on Solving Common OMPROUTE Problems: I112026

© IBM Corporation 2001, 2002

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

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. Logging is sent to the MVS console and the UNIX SYSLOGD log.
8. 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.
9. 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).

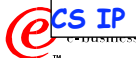
OMPROUTE Trace Details (1)

```

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

```

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



© IBM Corporation 2001, 2002

- 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"
- 10 OSPF neighbor transitions to state "ExStart"

OMPROUTE Trace Details (2)

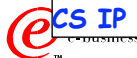
```

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)

```

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



© IBM Corporation 2001, 2002

- 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: 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.



5. Enhancements for OSPF in z/OS V1R2



© IBM Corporation 2001, 2002

Enhancements for OSPF in z/OS V1R2

1. Replaceable Static Routes
 - (see slide 16)
2. Enhanced NETSTAT display
 - (see slide 56)
3. MD5 authentication for Router Exchanges
 - improves interoperability with Cisco routers
4. Enhanced RIP filter support
 - improves migration from OROUTED



1. 1. Replaceable static routes has already been discussed on slide 16
2. 2. NETSTAT display enhancements were described on slide 54
3. 3. MD5 is an improved method of authenticating (NOT encrypting) router exchanges. It is supported by Cisco routers, and has been added to OMPROUTE to improve Cisco interoperability.
4. 4. Several RIP filters that OROUTED supports were added to OMPROUTE. Since OROUTED is due to be sunsetted, this should help smooth the migration to OMPROUTE



A. APPENDICES



e-business™

© IBM Corporation 2001, 2002



A.1. Simple CISCO Coding Examples for OSPF



© IBM Corporation 2001, 2002

Information: IBM & CISCO Interoperability

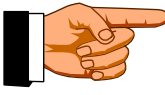
- Redbook on OSPF, EIGRP, MNLB, Sysplex Distributor:
 - Networking with z/OS and Cisco Routers: An Interoperability Guide (SG24-6297)
- Whitepaper on OSPF with IBM & CISCO:
 - "OSPF Design and Interoperability Recommendations for Catalyst 6500 and OSA-Express Environments"
http://www-1.ibm.com/servers/eserver/zseries/networking/pdf/ospf_design.pdf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



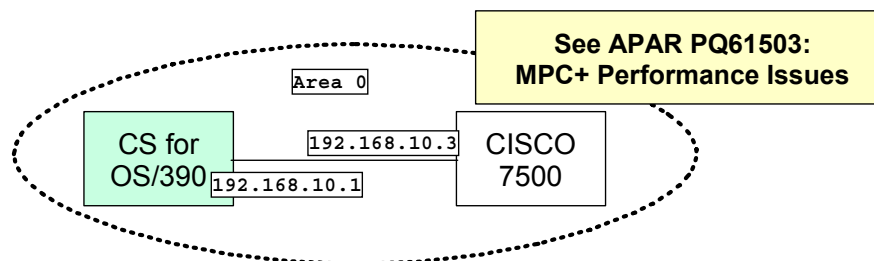
Consult APAR PQ48766 for information about CLAW implications for OSPF and RIP!



```
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
```

- Traditionally OS/390 or z/OS view 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.
- 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 that regardless of the number of hosts learned by the protocol. If you have coded nothing on the LINK statement, the output will reflect P2P.
- Summary: Prior to APAR PQ48766, CS/390 OSPF perceives a CLAW interface as point-to-point, whereas CISCO codes it to be point-to-multipoint. Note that Cisco does 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.
 - 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.
 - P2MP
 - Treat this CLAW link as a point-to-multipoint link.
 - The default is point-to-point.
 - 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.
 - 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.
 - 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.")
-
- ; CISCO CLAW DEFINITIONS in OS/390 TCP/IP
-
- ;
- DEVICE CIP2A CLAW D20 MVS001B C7507A PACKED 15 15 4096 4096
- LINK CISCO2 IP 0 CIP2A
- If you don't want to use the PACKING feature, you must change PACKED PACKED to TCPIP TCPIP in the CLAW statements.
- In the TCP/IP profiles, you must change PACKED to NONE.
- 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.
- 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.
- Consult INFO APARS I12353, I12361, and I12494 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.
- At z/OS V1R2, 60K CLAW packing is available.

MPC+ Coding with CISCO



OMPRROUTE.CONF for MPC+ to CISCO

```
; CIP 3C
OSPF_INTERFACE
  IP_address=192.168.10.1
  Name=LNKCIP3C
  Cost0=3
  Hello_Interval=2
  Dead_Router_Interval=8
  Subnet_mask=255.255.255.0
  MTU=4000
  Router_Priority=0
;
```

CISCO MPC+ MVS

```
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
```



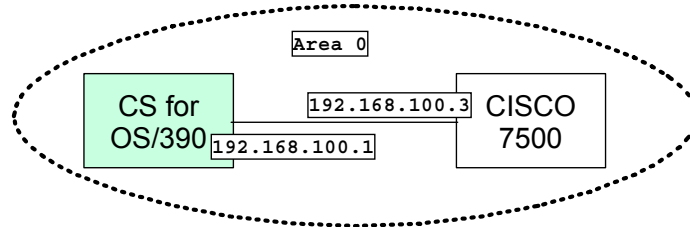
1. MPC+ can be perceived by OS/390 or z/OS as either point-to-point or point-to-multipoint. If the RIP or OSPF protocol determines that there are more than two hosts in the (sub)network represented by the MPC+ connection, the interface will be perceived as point-to-multipoint. If, on the other hand, 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 CPMC 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



OMPROUTE.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

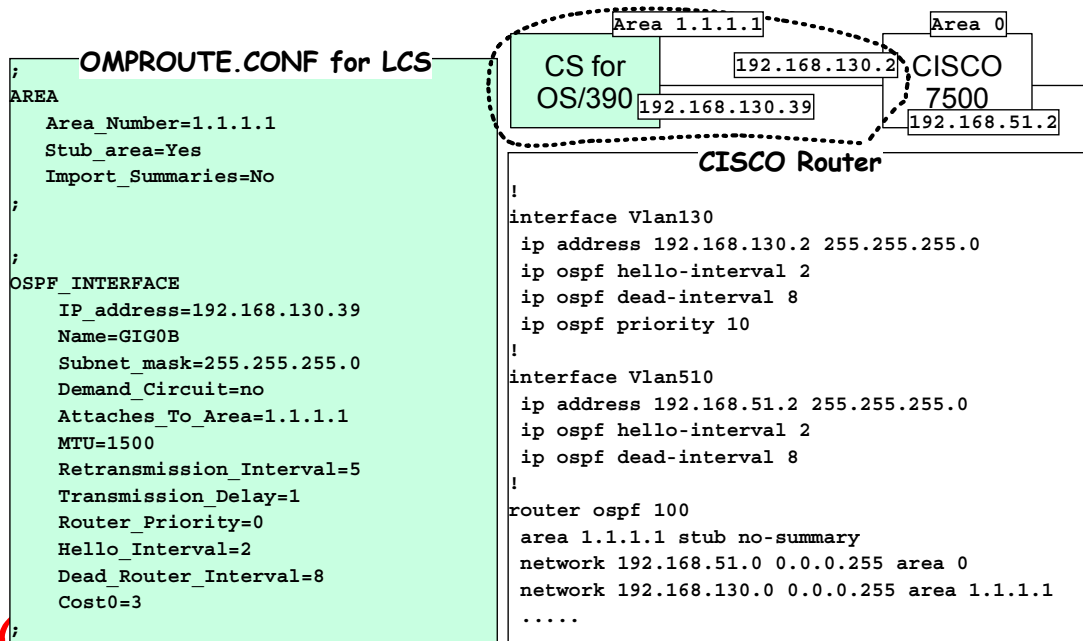
```

!
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 QDIO in Totally Stubby Area



IBM Corporation 2001, 2002

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.

OSPF/EIGRP Interoperability

EIGRP is a proprietary routing protocol supported by Cisco routers.

- **CS for z/OS does not support EIGRP.**
- **However, OSPF and EIGRP interoperate effectively using Cisco routers as the ASBRs.**

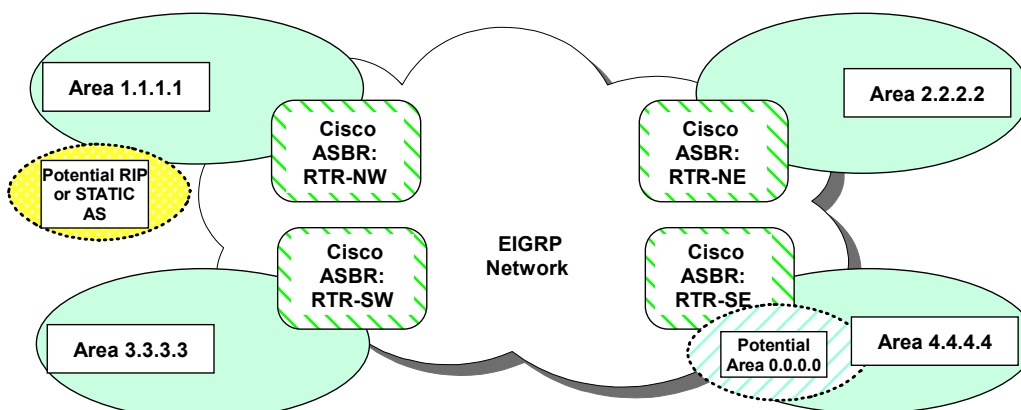
Though EIGRP is used in the following examples, in fact the backbone autonomous system could be any non-OSPF protocol.

- **How to configure the Cisco routers to interoperate EIGRP and OSPF is beyond the scope of this presentation.**

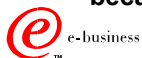


1. This information is useful for customers who have an existing Cisco EIGRP backbone and want to attach z/OS sysplexes or mainframe complexes to them, and use dynamic routing.

EIGRP + OSPF as Non-Backbone



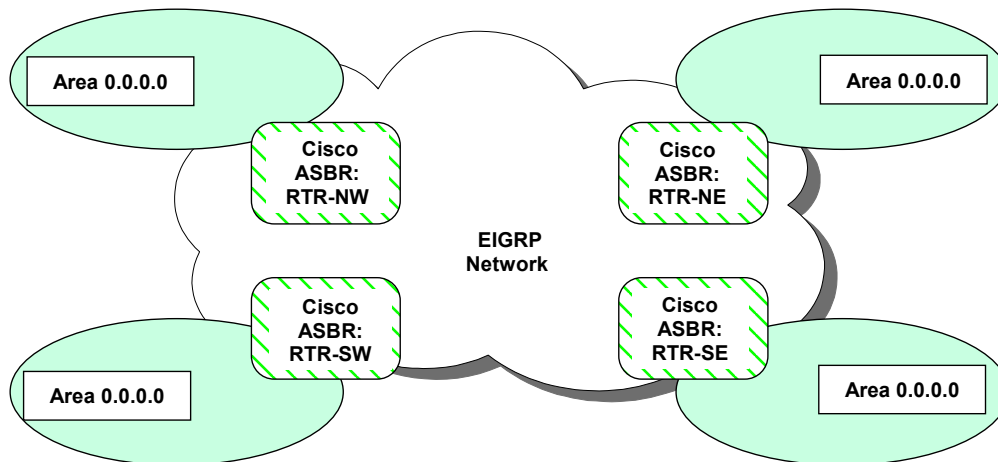
- OSPF areas may be attached as non-backbone to an EIGRP Autonomous System.
 - All routing between OSPF areas occurs through ASBRs and the EIGRP AS.
 - Cannot connect a second area to any of the OSPF areas unless second area is a backbone area (Area 0).
 - Permits strategy to insert non z/OS routers 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

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 be part of a Stub or Totally Stubby Area.

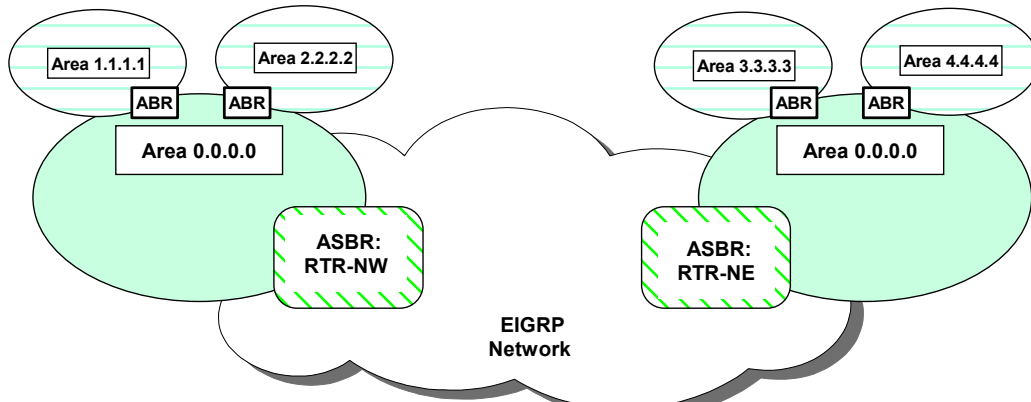
EIGRP + OSPF as Backbone



- **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.)**
 - **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).**



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.

MD5 Authentication: IBM & Cisco

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

IBM: Standard MD5 Key

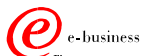
```
interface Ethernet0
 ip address 48.51.32.224 255.255.240.0
 ip ospf message-digest-key 4 md5 ABCDEFGHJKLMNPO
 ip ospf hello-interval 30
 ip ospf retransmit-interval 60
end
```

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

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.



A.2. Diagnosis: More Commands to Display OMPROUTE Environment



© IBM Corporation 2001, 2002

Multicast for Device Using OSPF

```

D TCPIP, ,N,DEV

DEVNAME: DEVOSAF1          DEVTYPE: LCS          DEVNUM: 0580
DEVSTATUS: READY
LNKNAME: LNKOSAF1          LNKTYPE: ETH          LNKSTATUS: READY
NETNUM: 0    QUESIZE: 0    BYTEIN: 0000410028    BYTEOUT:
0000092858
BROADCASTCAPABILITY: YES
BSD ROUTING PARAMETERS:
MTU SIZE: 01500            METRIC: 03
DESTADDR: 0.0.0.0          SUBNETMASK: 255.255.255.0
MULTICAST SPECIFIC:
MULTICAST CAPABILITY: YES
GROUP          REFCNT
-----
224.0.0.5      0000000001
224.0.0.1      0000000001
    
```



1. 224.0.0.5 is the multicast group to which OSPF belongs. (AllSPFRouters)

Displaying OMPRoute Environment

```

D TCPIP,,OMPRoute+--,OSPF -+-,LIST+--,ALL-----
|                                     |
|                                     | ,AREAS
|                                     | ,InterFaceS
|                                     | ,NeighBoRS
|                                     | ,NBMA
|                                     | +--,VLINKS
|                                     |
| +--,EXTERNAL
| +--,LSA--,LSTYPE=-ls_type-, ...
| |-,AREASUM
| |-,DATABASE--,AREAID=-area-id
| |-,DBSIZE
| |-,InterFace+-----
| |                                     |
| |                                     | +--NAME=-if_name
| |-,NeighBoR-----
| |-,ROUTERS
| | +--,STATiStics
| |-,RIP-----
| |-,RTTABLE

```



© IBM Corporation 2001, 2002

1. This is not meant to represent all the possibilities of executing the commands to display the OSPF and RIP environments under OMPROUTE.
2. Useful commands to start with for OSPF are ...
 1. D TCPIP,,OMPR,OSPF,NBRS
 2. D TCPIP,,OMPR,OSPF,IFS
 3. D TCPIP,,OMPR,RTTABLE
3. The commands preceded by the word "LIST" display output from what you have coded in the OMPRoute Configuration file. They do not necessarily display the status of the running system, with its discovered neighbors and so on.

Displaying the Link State Database

```

D TCPIP,TCPIP,OMPR,OSPF,DATABASE,AREAID=0.0.0.0
EZZ7853I AREA LINK STATE DATABASE 274
TYPE LS DESTINATION      LS ORIGINATOR      SEQNO      AGE      XSUM
1 @1.1.2.201             1.1.2.201          0X80000010 1454    0X76A3
1 @192.168.11.1          192.168.11.1       0X80000015 343     0X3509
1 @192.168.100.5         192.168.100.5      0X80000094 1505    0X8AF4
1 @206.10.20.1           206.10.20.1        0X800000DA 722     0X4304
1 @206.10.20.2           206.10.20.2        0X80000158 1982    0XB9FC
2 @192.168.0.4           206.10.20.2        0X80000008 967     0XB841
2 @192.168.70.5          192.168.100.5      0X8000000A 1505    0XAE47
2 @192.168.100.4         206.10.20.2        0X8000000C 1462    0X1F27
2 @192.168.110.4         206.10.20.2        0X8000000A 1462    0XB489
# ADVERTISEMENTS:      9
CHECKSUM TOTAL:        0X46DD8

D TCPIP,TCPIP,OMPR,OSPF,DBSIZE
EZZ7854I LINK STATE DATABASE SIZE 276
# ROUTER-LSAS:          5
# NETWORK-LSAS:         4
# SUMMARY-LSAS:         0
# SUMMARY ROUTER-LSAS:  0
# AS EXTERNAL-LSAS:    16
# INTRA-AREA ROUTES:   13
# INTER-AREA ROUTES:   0
# TYPE 1 EXTERNAL ROUTES: 0
# TYPE 2 EXTERNAL ROUTES: 4

```



1. All non ASBR or ABR routers in the same area should have identical LSA databases.
2. Type = TYPE OF LS
 1. 1 = ROUTER LINKS
 2. 2 = NETWORKLINKS
 3. 3 = NETWORK SUMMARIES
 4. 4 = ASBR SUMMARIES
 5. 5 = EXTERNAL LSAs
 6. (7 = NSSA summary LSAs -- sorry, no support for this in OMPROUTE!!!)
3. DESTINATION =IP Subnet/Network #
4. LSAGE = in seconds (max = 3600)
 1. Seqno., age, and xsum compared to see which instance is most recent
5. CHECKSUM TOTAL = can be used to compare if 2 routers have synched DBs.



A.3. Miscellaneous Information



e-business™

© IBM Corporation 2001, 2002

OMPROUTE Proc: Trace to HFS

```
//OMPROUTE EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
//          PARM='PGM /usr/lpp/tcpip/sbin/omproute'
//*        PARM='PGM /usr/lpp/tcpip/sbin/omproute -t1'
//* STDENV  DD PATH='/u/usernnn/envcs2',
//STDENV   DD DSN=SYS1.TCPPARMS(OMPRENV),DISP=SHR
//          PATHOPTS=(ORDONLY)
//STDOUT   DD PATH='/tmp/omproute.stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//STDERR   DD PATH='/tmp/omproute.stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

```
export RESOLVER_CONFIG=// 'SYS1.TCPPARMS(TCPDATA)'
export OMPROUTE_FILE=/etc/omproute.conf
export OMPROUTE_DEBUG_FILE=/tmp/omproute.debug
export OMPROUTE_DEBUG_FILE_CONTROL=1000,10
export TZ=EST4EDT
```

```
Autolog
OMPROUTE
EndAutolog
Port
520 UDP OMPROUTE NoAutolog
```

Warning: Tracing!
Warning: UDP 520!

1. This cataloged procedure runs the OMPROUTE daemon as a BPXBATCH program. You may wish to consult the Informational APAR on OMPROUTE Problems: II12026 for more information. When OMPROUTE runs as a BPXBATCH program, it cannot be made non-swappable; i.e., it cannot be placed in the PPT.
2. Beware of leaving tracing on permanently, as this example shows. You should specify tracing in the procedure only when you are starting out and becoming familiar with OMPROUTE. Thereafter, remove tracing from the procedure and start tracing with a MODIFY command when necessary. Tracing has significant negative impact on OMPROUTE performance.
3. BPXBATCH programs write STDOUT and STDERR to HFS files, designated by the "DD PATH" statement in the JCL.
4. The PATHOPTS "OTRUNC" erases the STDOUT and STDERR file at every initialization of OMPROUTE. If you wish to preserve the files, replace "OTRUNC" with "OAPPEND."
 1. Although the STDOUT HFS files cannot grow beyond whatever you have specified in the "CONTROL" statement, you should ensure that you have enough space in the HFS you designate. If you are using debug or trace, clean out the debug and trace files before every trace.
5. The STDENV DD card points to a file containing the environment definitions ("OMPRENV"). This file may be an HFS or an MVS dataset, as in this example.
 1. The STDENV file contains the location and name of the OMPROUTE configuration file, of the resolver configuration file (TCPDATA), possibly an OMPROUTE debug file, or even other environment variables, such as the time zone that will provide correct timestamp data.
 1. Standard debug output flows to STDOUT. You may override this location with the environment variable "OMPROUTE_DEBUG_FILE=...."
 2. Output from the trace or debug options is written to stdout with one exception. The exception is when the routing application was started with no tracing, and then a MODIFY command was issued to enable tracing. In this case, the output destination defaults to the file omproute_debug in the current temporary directory (the default is /tmp). The debug output destination will have an override capability via the use of an environment variable (OMPROUTE_DEBUG_FILE). This environment variable was added via PTF to V2R6, V2R7, and V2R8.
 3. To control the size of the debug file that is allocated, a new environment variable was introduced via PTF to V2R8 (APAR PQ37046): OMPROUTE_DEBUG_FILE_CONTROL.
 4. OMPROUTE_DEBUG_FILE_CONTROL=<size of file>,<num of files> . The default values for <size of file> and <num of files> are 200,5 respectively which were the hard coded values used before this APAR. Size is in kilobytes; the files wrap. RECOMMENDED to erase trace/debug files before issuing "MODIFY OMPROUTE,TRACE=n." (To disable tracing, issue "MODIFY OMPROUTE,TRACE=0".)
6. The OMPROUTE configuration file may be placed in an MVS dataset or in an HFS file. If it is placed in an MVS dataset, ensure that there are no leading blanks or trailing numbers. Otherwise the Configuration file (and any other file coded in an MVS dataset) may not be found.
7. Note an important point if you have a STDENV definition that resides in a Fixed Block (FB) dataset. The _CEE_ENVFILE variable has a restriction with FB datasets. It does not strip trailing blanks from the end of the variable. Thus, you have the problem that ishell can't view files in the HFS; therefore point STDENV to an HFS file or to a Variable Blocked MVS file; in this case the file will be created without blanks.

OMPRoute Proc: Trace to STDOUT

```
//OMPROUTE PROC
//*
//OMPROUTE EXEC PGM=OMPROUTE,REGION=0M,TIME=NOLIMIT,
//          PARM='POSIX(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/ -t2'
//STDENV   DD DSN=SYS1.TCPPARMS(OMPENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

SYS1.TCPPARMS(OMPENV)

```
RESOLVER_CONFIG=// 'SYS1.TCPPARMS(TCPDATA) '
OMPROUTE_FILE=/etc/omproute.conf
TZ=EST4EDT
```

```
Autolog
OMPROUTE
EndAutolog
;
Port
520 UDP OMPROUTE NoAutolog
```



Warning: Tracing!
Warning: UDP 520!



1. You may wish to consult the Informational APAR on OMPROUTE Problems: II12026 for more information.
2. You can start OMPROUTE as a normal POSIX(ON) program and have run-time messages sent to standard sysout by adding a SYSPRINT and a SYSOUT DD statement. The sample JCL that is shipped in hlq.SEZAINST at V2R6 uses BPXBATCH and HFS files. Later Samples at V2R7 and higher use JCL that includes "POSIX(ON)" coding, as you see here. If you need to run OMPROUTE as non-swappable, you must run it as shown here: as a started task and not as a BPXBATCH program.
3. Beware of leaving tracing on permanently, as this example shows. You should specify tracing in the procedure only when you are starting out and becoming familiar with OMPROUTE. Thereafter, remove tracing from the procedure and start tracing with a MODIFY command when necessary.
4. If _CEE_ENVFILE is used with an MVS data set, the data set must be allocated with RECFM=V. (RECFM=F enables padding with blanks for the environment variables, and is therefore not recommended. For example, the dynamically allocated tracing files would not be legible from the ishell as they would contain trailing blanks.)
5. Tracing consumes storage and drives CPU utilization. When possible enable tracing through MODIFY command and disable as soon as trace data is captured.
6. Common error with OMPROUTE when RIP interfaces are not defined in the configuration file:
 1. Ensure that UDP port 520 (used by RIP) is either omitted from the PORT statement or that it is coded with NoAutolog. Otherwise, OMPROUTE will be repeatedly stopped and restarted by the AUTOLOG process.

Common Errors at V2R6 or Higher

```

S OMPROUTE
$HASP100 OMPROUTE ON STCINRDR
IEF695I START OMPROUTE WITH JOBNAME OMPROUTE IS ASSIGNED TO
USER TCPIP1A, GROUP OMVSGRP
$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(TCPIP1A ) GROUP(OMVSGRP ) NAME(#####)
MVS.ROUTE MGR.OMPROUTE CL(OPERC MDS)
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

```



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.ROUTE MGR.OMPROUTE) UACC(NONE)
 2. PERMIT MVS.ROUTE MGR.OMPROUTE ACCESS(CONTROL) - CLASS(OPERC MDS) ID(TCPIP1A)
 3. SETROPTS RACLIST(OPERC MDS) 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.

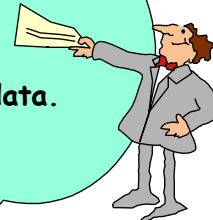


1. OMPROUTE requires association with a Superuser just as the TCPIP cataloged procedure does.
2. In addition, even this superuser must be permitted to the operator command "MVS.ROUTE MGR.OMPROUTE."
 1. OROUTED at V2R6 also requires special command authorization for "MVS.ROUTE MGR. OROUTED."
 1. At V2R5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

Using SNMP with OMPROUTE

```
ROUTESA_CONFIG  
COMMUNITY="FRED"  
AGENT=161  
ENABLED=YES;
```

1. MIBs supported per RFC 1850; Use SNMP as an alternative to the MVS Display Commands
2. Add ROUTESA_CONFIG to the Omproute Configuration File.
3. Start OSNMPD (specify community name)
4. When OMPROUTE Subagent connects with agent, the agent may now collect OSPF SNMP data.



1. The syntax for ROUTESA_CONFIG is incorrectly documented in several manuals. An INFO APAR has been taken. Note the quotation marks around the community name. Also note the use of the "=" sign.
2. MIB Groups supported per RFC 1850:
 1. ospfBasicGroup
 2. ospfAreaGroup
 3. ospfStubAreaGroup
 4. ospfLsdbGroup
 5. ospfIfGroup
 6. ospfIfMetricGroup
 7. ospfVirtIfGroup
 8. ospfNbrGroup
 9. ospfVirtNbrGroup
 10. ospfExtLsdbGroup
 11. ospfAreaAggregateGroup

Multipath Detail for CS for OS/390 IP

Multipath Routes	OMPROUTE (OSPF)	OMPROUTE (RIP)	OROUTED	GATEWAYS / BEGINROUTES / DEFAULT (Static)
Direct Network	Yes (up to 4)	Yes (up to 4 for redundant interfaces)	No	Yes (no Limit)
Direct Host	Yes (up to 4)	Yes (up to 4 redundant interfaces)	No	Yes (no Limit)
Indirect Network or Host	Yes (up to 4)	No	No	Yes (no Limit)
Default (indirect)	Yes (up to 4)	No	No	Yes (no Limit)

**** OMPROUTE presents the stack with multiple routes; the stack makes the decision about which interface to use.**

If you want multipath routes for non-direct networks with OROUTED, use the GATEWAY/DEFAULT multipath static routes.



© IBM Corporation 2001, 2002

1. In the past we have advised implementers of (O)ROUTED on OS/390 not to mix GATEWAY definitions (used for static routes) with BSDROUTINGPARMS statements (used for OROUTED).
2. Beginning with V2R6 of CS for OS/390 it is possible to implement something called multipath, whereby you may have multiple routes to the same destination and traffic is split among the routes according to the release of OS/390 and the selected implementation of multipath on OS/390.
3. As of V2R8 it is possible to implement multipath on a per-packet or a per-connection basis.
4. Prior to V2R8, multipath, if coded, is implemented on a per-destination basis.
5. If you would like to implement multipath and are using OROUTED, you must code the multipath routes in the GATEWAYS or BEGINROUTES section of your PROFILE.TCPIP. These routes are then static routes and not dynamic routes.



A.4. Bibliography & References




© IBM Corporation 2001, 2002

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)"**

© IBM Corporation 2001, 2002

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, Uyles	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





e-business™

© IBM Corporation 2001, 2002