LAN Connection and IP Networks
NetPerformer® System Reference

MEMOTEC
redefining network efficiency

COPYRIGHTS AND DISCLAIMERS

Published Date: April 2014

Document # 1605

This publication contains information proprietary and confidential to Memotec Inc. Any reproduction, disclosure or unauthorized use of this publication is expressly prohibited except as Memotec Inc. may otherwise authorize in writing.

Memotec Inc. reserves the right to make changes without notice in product or component design as warranted by evolution in user needs or progress in engineering or manufacturing technology. Changes which affect the operation of the unit will be documented in the next revision of the manual.

We have made every effort to ensure the accuracy of the information presented in our documentation. However, Memotec assumes no responsibility for the accuracy of the information published. Product documentation is subject to change without notice. Changes, if any, will be incorporated in new editions of these documents. Memotec may make improvements or changes in the products or programs described within the documents at any time without notice. Mention of products or services not manufactured or sold by Memotec is for informational purposes only and constitutes neither an endorsement nor a recommendation for such products or services.

Memotec Inc. is a wholly owned subsidiary of Comtech EF Data Corp., and its parent company Comtech Telecommunications Corp (NASDAQ: CMTL).

Memotec Inc.
7755 Henri Bourassa Blvd. West
Montreal, Quebec
Canada H4S 1P7
Tel.: (514) 738-4781
FAX: (514) 738-4436
www.memotec.com

Contents

1

Ethernet LAN Connection

1.1 Configuring the Ethernet LAN Port

This section describes specific parameters in the Setup menu that define the Ethernet port connection. Use the Setup Port menu to configure this port.

Console	SNMP	Text-based Config
SE/PORT	*iflan* (category)	*[iflan#]* heading

The Setup Slot or Setup Port menu lets you configure all parameters that affect the operation of the Ethernet LAN ports.

If you are configuring the NetPerformer using SNMP, select the *iflan* category, which includes all variables affecting the LAN ports. For text-based configuration the *[iflan#]* heading is used, where # represents the LAN number.

SE/PORT/ETH example

```
AF.9210-1>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/ELOG/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/SS7/USER/VLAN,
def:PORT) ? PORT
Port number (ETH1/ETH2/CSL/1,def:ETH1) ? ETH1
PORT ETH 1> Protocol (def:ETH AUTO) ?
PORT ETH 1> Link integrity (def:YES) ?
PORT ETH 1> LAN speed (mbps) (def:AUTO) ?
PORT ETH 1> MAC address (def:000000000000) ?
PORT ETH 1> Redundancy MAC address active (def:NO) ?
PORT ETH 1> DHCP (def:DISABLE) ?
PORT ETH 1> IP address 1 (def:000.000.000.000) ? 192.168.0.1
PORT ETH 1> Subnet mask 1 (number of bits) (0-32,def:8) ? 24
{255.255.255.000}
PORT ETH 1> IP address 2 (def:000.000.000.000) ? 64.10.4.1
PORT ETH 1> Subnet mask 2 (number of bits) (0-32,def:8) ? 24
{255.255.255.000}
PORT ETH 1> Redundancy IP address 1 (def:000.000.000.000) ?
PORT ETH 1> Redundancy subnet mask 1 (number of bits) (0-32,def:8) ?

{255.000.000.000}
PORT ETH 1> Redundancy IP address 2 (def:000.000.000.000) ?
PORT ETH 1> Redundancy subnet mask 2 (number of bits) (0-32,def:8) ?

{255.000.000.000}
PORT ETH 1> Allow routing between IP networks (def:YES) ? NO
PORT ETH 1> Frame size (128-8192,def:1500) ?
PORT ETH 1> IP RIP (def:V1) ?
PORT ETH 1> IP RIP TX/RX (def:DUPLEX) ?
PORT ETH 1> OSPF (def:DISABLE) ?
PORT ETH 1> IGMP enable (def:NO) ?
PORT ETH 1> IP multicast active (def:NO) ?
PORT ETH 1> IP multicast 1 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 2 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 3 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 4 (def:000.000.000.000) ?
PORT ETH 1> NAT enable (def:NO) ?
PORT ETH 1> VLAN enable (def:NO) ?
PORT ETH 1> IPX RIP (def:DISABLE) ?
PORT ETH 1> IPX SAP (def:DISABLE) ?
PORT ETH 1> IPX network number (def:00000000) ?
PORT ETH 1> IPX encapsulation (def:ETH 802.2) ?
PORT ETH 1> Physical connectivity detection (def:DISABLE) ?
AF.9210-1>
```

For details on these parameters, turn to the appendix “SE/PORT/ETH Configuration Parameters” on page 8-1.

1.1.1 Supported Connections

A NetPerformer Ethernet connection supports:

- 10 Mbps and/or 100 Mbps traffic
- Ethernet IEEE 802.3 and Ethernet V2.0 formats
- Link Integrity function
- Use of a burned-in address, a locally established address (manually supplied) or a DHCP client address (automatically supplied).
- IP (RIP/OSPF) and IPX (RIP/SAP) routing on all platforms.

NOTE: OSPF routing is covered in the chapter “OSPF Network Support” on page 7-1. IPX routing is described in the *Digital Data* section.

1.1.2 MAC Addresses

The Ethernet port drivers support two MAC addresses to improve the reliability of a redundant system at startup.

- The first is the current BIA or MAC address (if configured).
- The second is the redundant MAC address, which is used when the unit sends an IP frame with the redundant IP address as the source IP address.

To activate the redundant MAC address on an Ethernet port:

1. Set the Ethernet port parameter *Redundancy MAC address active* to **YES.**
2. Configure the unit for Redundancy.
3. The unit must also be in operating state. If it is in **STANDBY** or **REPLACED** mode, the redundant MAC address state will be displayed as **inactive**.

NOTE: **In addition to the above requirements, the redundant MAC address will be used on transmission only if the source IP address is a redundant IP address configured on the Ethernet port**. The redundant MAC address is used on reception to validate a received frame regardless of the destination IP address in the IP frame.

The LAN interface is included in the ARP table to distinguish cases where more than one ARP entry has the same destination IP address. For more information, see “Displaying the ARP Cache” on page 2-34.

1.1.3 MAC Address Statistics

You can display statistics for the current status of the redundant MAC address in both the Display States (**DS**) command and *statSystem* for SNMP. These statistics include the redundant MAC address and its current status: **inactive** or **active**.

DS Command on an SDM-8400:

```
UNITAF1>DS
DISPLAY STATES
Item (GLOBAL/PORT/PVC/REDUNDANCY,def:GLOBAL) ? PORT
PORT ETH> Protocol.................................ETHERNET
PORT ETH> Interface................................10BASET
PORT ETH> Speed....................................100M
PORT ETH> Duplex mode..............................HALF
PORT ETH> Operating mode...........................L-
PORT ETH> State....................................OPEN
PORT ETH> Network address..........................00200AB0AFEF
PORT ETH> Redundancy MAC address...................00005E000000
(inactive)
PORT ETH> Burned-in address........................00200AB0AFEF
PORT ETH> Number of deferred transmissions.........0
PORT ETH> Number of collision frames...............0
```

DS command on an SDM-9620:

```
AF.9620-2>DS
DISPLAY STATES
Item (GLOBAL/PORT/REDUNDANCY/SPAN,def:GLOBAL) ? PORT
PORT ETH 1> Protocol...............................ETHERNET
PORT ETH 1> Interface..............................10BASET
PORT ETH 1> Speed..................................100M
PORT ETH 1> Duplex mode............................HALF
PORT ETH 1> Operating mode.........................L-
PORT ETH 1> State..................................OPEN
PORT ETH 1> Network address........................00200CE014F1
PORT ETH 1> Redundancy MAC address.................00005E000000
(inactive)
PORT ETH 1> Burned-in address......................00200CE014F1
PORT ETH 1> Number of deferred transmissions.......0
PORT ETH 1> Number of collision frames.............0

PORT ETH 2> Protocol...............................ETHERNET
PORT ETH 2> Interface..............................10BASET
PORT ETH 2> Speed..................................UNKNOWN
PORT ETH 2> Duplex mode............................HALF
PORT ETH 2> Operating mode.........................--
PORT ETH 2> State..................................OPEN
PORT ETH 2> Network address.........................00200CCE014F1
PORT ETH 2> Redundancy MAC address..................00005E000001
(active)
PORT ETH 2> Burned-in address.......................00200CE014F2
PORT ETH 2> Number of deferred transmissions........0
PORT ETH 2> Number of collision frames..............0
```

The equivalent information can be retrieved in SNMP using new entries in the *statSystem* table:

- *statSystemRedunMacAddrState1*
- *statSystemRedunMacAddr1*
- *statSystemRedunMacAddrState2*
- *statSystemRedunMacAddr2.*

1.2 LAN Connection Status

The following commands are available from the console to view the status of LAN connections on the NetPerformer:

- **Display Counters (DC):** shows all counters stored in memory, including the mean or peak value of the transmitter and receiver rates on the LAN port (see next section)
- **Display States (DS):** provides current status information (see "Display States (DS) Command" on page 1-7)
- **Display Errors (DE):** shows the values of the error counters (see "Display Errors (DE) Command" on page 1-7).

In addition, the Ethernet port is equipped with LED indicators that provide visual status information:

- **LI or LNK**: (red) This LED goes on when Link Integrity is established
- **COL**: (yellow) This LED flashes each time a collision occurs
- **RX**: (green) This LED goes on when LAN traffic is received at the Ethernet port
- **TX**: (red) This LED goes on when LAN traffic is transmitted from the Ethernet port.

For further information on these LED indicators, refer to the *Hardware Installation Guide* for your NetPerformer product.

1.2.1 Display Counters (DC) Command

Console	SNMP
DC/PORT	*statIflan* (category)

The Display Counters command allows you to view all counters stored in memory. These counters include the mean and peak values of the transmitter and receiver rates on the LAN port. The NetPerformer takes a snapshot of these counters every 5 seconds, and keeps this information for a maximum of 2 minutes. Mean rates are calculated over the entire 2-minute period, whereas peak rates are obtained by comparing all the snapshots taken.

To view the LAN counters, enter **DC** on the console command line and select **PORT**. You can choose to display either mean (**M**) or peak (**P**) rates.

DC/PORT example

```
SDM-9380>DC
DISPLAY COUNTERS
Item (BOOTP/CONFIG/DNS/IP/NAT/PORT/PVC/Q922/Q933/QOS/SLOT/SVC/TIMEP,
def:Q933) ? PORT
Counters (MEAN/PEAK,def:MEAN) ?
Compression rate................................6.04 (M)
Decompression rate..............................6.18 (M)
PORT ETH> Transmitter rate......................0    kbps (M)
PORT ETH> Receiver rate.........................0    kbps (M)
```

Details on these counters are provided in the “Ethernet Port Statistics” on page 11-2.

1.2.2 Display States (DS) Command

Console	SNMP
DS/PORT	*statIflan* (category)

The Display States command lets you view status information for the LAN interface. Enter **DS** followed by **PORT**. A display like the following will appear on the screen:

DS/PORT example

```
SDM-9360>DS
DISPLAY STATES
Item (GLOBAL/PORT/PU/PVC/SLOT/SVC/VLAN,def:GLOBAL) ? PORT
PORT ETH> Protocol.............................ETHERNET
PORT ETH> Interface............................10BASET
PORT ETH> Speed................................10M
PORT ETH> Duplex mode..........................HALF
PORT ETH> Operating mode.......................L-
PORT ETH> State................................OPEN
PORT ETH> Network address......................AAAAAAAAA001
PORT ETH> Burned-in address....................00200AB05825
PORT ETH> Number of deferred transmissions......430
PORT ETH> Number of collision frames...........2261
```

Details on these statistics are provided on “DS/PORT/ETH” on page 11-2.

1.2.3 Display Errors (DE) Command

Console	SNMP
DE/PORT	*statIflan* (category)

The Display Errors command displays the number of errors that have occurred for each error type stored in memory. Enter **DE** followed by **PORT**. Use the Reset Counters (**RC**) command to return the error counters to zero.

DE/PORT example

```
SDM-9230>DE
DISPLAY ERRORS
Item (BOOTP/CHANNEL/DICT/GROUP/NAT/PORT/PU/PVC/Q922/SLOT/SVC/TIMEP,
def:SLOT) ? PORT
PORT ETH> Number of excessive collisions........0
PORT ETH> Number of late TX collision errors....0
PORT ETH> Number of underruns...................0
PORT ETH> Number of late RX collision errors....0
PORT ETH> Number of overruns....................0
PORT ETH> Number of busy conditions.............0
PORT ETH> Number of FCS errors..................0
PORT ETH> Number of alignment errors............0
PORT ETH> Number of carrier sense errors........0
PORT ETH> Number of bad frames..................0       ------
PORT ETH> Number of retries.....................0
PORT ETH> Number of restarts....................0
```

Details on these errors are provided on “DE/PORT/ETH” on page 11-4.

2

IP Connections

NOTE: This chapter describes basic IP applications. Other NetPerformer IP applications are discussed in “Advanced IP Applications” on page 8-1.

2.1 Multihomed IP Addressing

A multihomed IP address permits using several host IP addresses at a single point in the network in order to access various remote sites. This system reduces the overall number of network IP addresses required. It also permits IP routing **without** using the RIP protocol (*IP RIP* parameter disabled).

NOTE: The *Router* parameter of the Setup IP Global menu must be enabled to permit routing the required TCP/IP frames for any IP routing application.

Multihomed addressing is optional when IP RIP is enabled (the NetPerformer default setting). It may be used if IP RIP is disabled on a NetPerformer that sends LAN data to another site. To configure a multihomed IP address, use the global *Default IP Address* and Default *IP Mask* parameters in the Setup Global menu. For a general description of these parameters, refer to the chapter *Global Functions* in the *Quick Configuration* fascicle of this document series.

In rare cases, multihomed IP addressing can interfere with RIP routing. This can arise, for example, when two NetPerformer units are on the same LAN and are also connected via a WAN link. To turn multihomed IP addressing off, use the **MULTIHOMEDTYPE** extended parameter. for details, refer to the *Extended Parameters* fascicle of this document series.

NOTE: If you are using RIP Version 2 and different subnet masks have different lengths, the MULTIHOMEDTYPE extended parameter must be set to **IGNORENET**. This would be required, for example, when routing an IP frame with a destination address that uses a subnet with more bits than the local IP address mask.

2.1.1 Routing with the Default Configuration (RIP Version 1)

To be able to route IP data across the network the IP RIP parameter and either the global Default IP Address or the PVCR port IP Address must be properly configured. By default, all IP RIP parameters on the NetPerformer are enabled, which permits end-to-end IP routing. If you use the default IP RIP values you must:

- Configure each LAN with a unique network IP address using the IP Address parameter on the Ethernet port.
- Configure each link used to pass data from one NetPerformer to the next with its own network IP address. Use the PVCR port IP Address parameter. Each time the data passes over a router, the network IP address in the header information changes to the address of the current location.

In a simple routing application neither the global Default IP Address nor the PVCR port IP

Address parameters are required. Only the LAN IP addresses are used to pass information between the local and remote sites. The NetPerformer generates a routing table, and the local PC can reach the remote NetPerformer.

For example, you could configure and manage the remote NetPerformer using the SNMP agent available at the local PC.

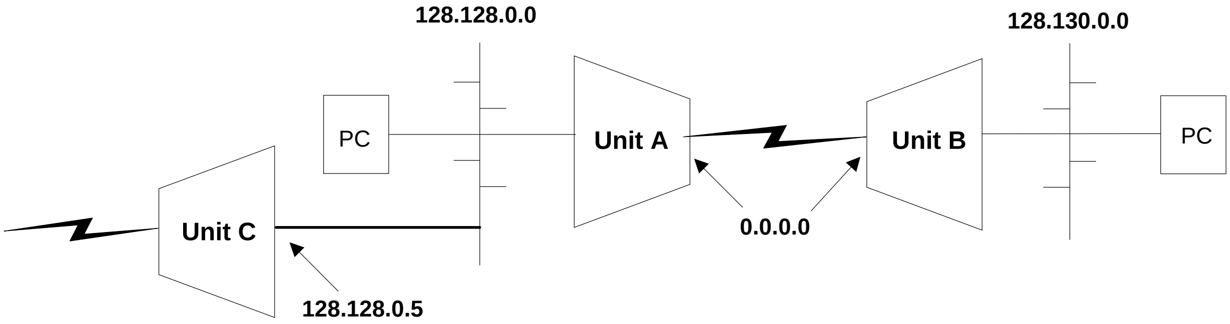


Figure 2-1: A remote NetPerformer using the SNMP agent

This addressing system can be used as long as:

- Each NetPerformer is connected to an active LAN,
- All PVCR links between NetPerformers have IP RIP enabled, and
- No IP addresses are configured for the PVCR links or for the NetPerformer as a whole. The LAN network addresses are sufficient to permit routing between NetPerformers.

In the example above, a default gateway for unit A is optionally required, depending on the role of the unit when routing information (hub versus simple router). If a remote LAN was attached to unit C, a default gateway could be defined on unit A to route information to that LAN. In this case the default gateway of unit A would be equivalent to the IP address of unit C: 128.128.0.5. To define a default gateway, use the global *Default Gateway* parameter.

NOTE: In the table of example addresses (Table 2-1), the *Default Gateway* parameter is not defined (**0.0.0.0**). However, in each example a default gateway could be defined in order to reach a remote LAN, if required.

Parameter	Local Unit “A”	Remote Unit “B”
IP Addr. of attached PC:	128.128.0.1	128.130.0.1
IP Mask of attached PC:	255.255.0.0	255.255.0.0

Table 2-1: Example addresses

Parameter	Local Unit “A”	Remote Unit “B”
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	0.0.0.0	0.0.0.0
Default IP Mask:	0.0.0.0	0.0.0.0
LAN IP Address:	128.128.0.2	128.130.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0
PVCR IP Address:	0.0.0.0	0.0.0.0
PVCR IP Mask:	0.0.0.0	0.0.0.0

Table 2-1: Example addresses

2.1.2 Configuring the PVCR Port IP Addresses

As an alternative to the addressing system above, you can configure the PVCR port IP addresses when IP RIP is enabled on the PVCR links.

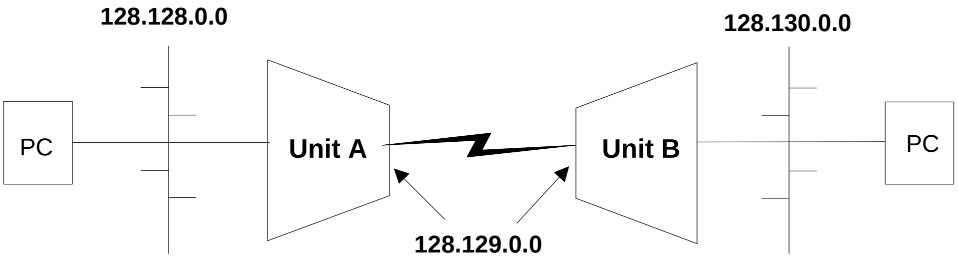


Figure 2-2: IP RIP enabled on the PVCR links

Three network addresses are used: one for the local LAN, one for the WAN link and one for the remote LAN. For this setup:

- Enable IP RIP on all PVCR ports (leave the PVCR port *IP RIP* parameter at its default value).
- IP RIP on the LAN ports can be enabled or disabled.

Parameter	Local Unit	Remote Unit
IP Addr. of attached PC:	128.128.0.1	128.130.0.1
IP Mask of attached PC:	255.255.0.0	255.255.0.0
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	0.0.0.0	0.0.0.0

Table 2-2: Examples of PVCR Port IP Addresses

Parameter	Local Unit	Remote Unit
Default IP Mask:	0.0.0.0	0.0.0.0
LAN IP Address:	128.128.0.2	128.130.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0
PVCR IP Address:	128.129.0.1	128.129.0.2
PVCR IP Mask:	255.255.0.0	255.255.0.0

Table 2-2: Examples of PVCR Port IP Addresses

2.1.3 Global IP Address (not multihomed)

Instead of configuring the PVCR port IP addresses, you can assign a global Default IP Address which will be automatically applied to the PVCR links.

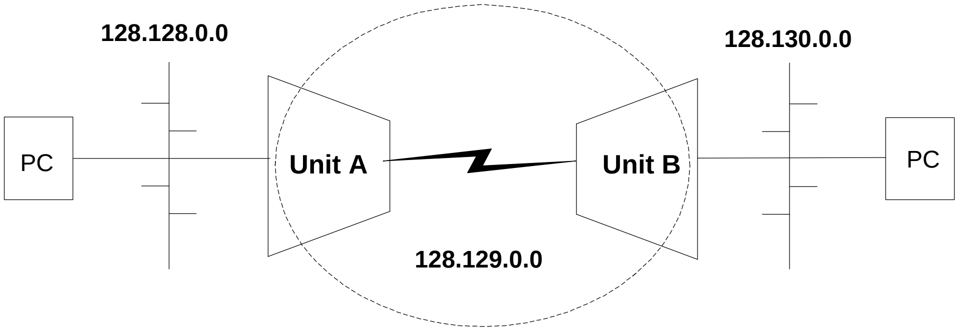


Figure 2-3:

Three network addresses are used: one for the local LAN, one for the NetPerformers and one for the remote LAN. For this setup:

- Enable IP RIP on all PVCR ports (leave the PVCR port IP RIP parameter at its default value).
- IP RIP on the LAN ports can be enabled or disabled.

Example addresses:

Parameter	Local Unit	Remote Unit
IP Addr. of attached PC:	128.128.0.1	128.130.0.1
IP Mask of attached PC:	255.255.0.0	255.255.0.0
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	128.129.0.1	128.129.0.2
Default IP Mask:	255.255.0.0	255.255.0.0

Table 2-3:

Parameter	Local Unit	Remote Unit
LAN IP Address:	128.128.0.2	128.130.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0
PVCR IP Address:	0.0.0.0	0.0.0.0
PVCR IP Mask:	0.0.0.0	0.0.0.0

Table 2-3:

2.1.4 Multihomed Global IP Address

The Global IP address can also be used for all devices connected through the NetPerformer port interfaces.

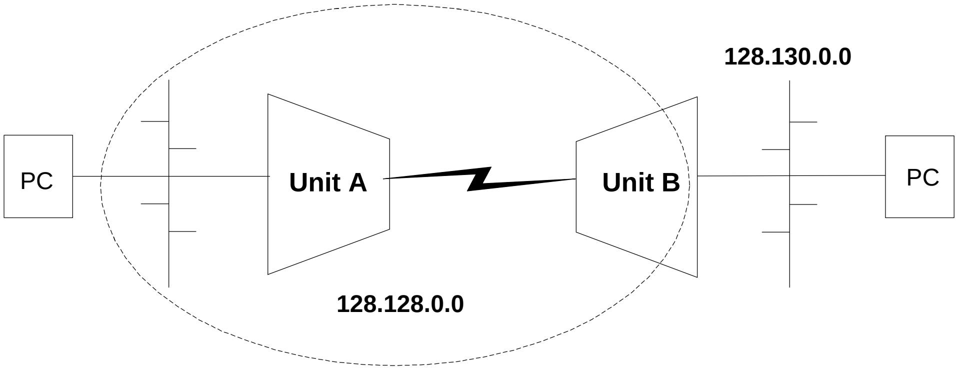


Figure 2-4:

Only two network addresses are used: one for the local LAN and one for the remote LAN. For multihomed addressing, the local LAN network area includes all devices attached to the LAN via the PVCR links of the NetPerformer. This setup requires fewer configuration changes than the other setups mentioned earlier:

- IP RIP can be enabled or disabled on all ports (both PVCR and LAN).

Another advantage is that you save one network IP address, since the local and remote NetPerformers use the same network address as the local LAN.

Example addresses:

Parameter	Local Unit	Remote Unit
IP Addr. of attached PC:	128.128.0.1	128.130.0.1
IP Mask of attached PC:	255.255.0.0	255.255.0.0
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	128.128.0.10	128.128.0.11

Table 2-4:

Parameter	Local Unit	Remote Unit
Default IP Mask:	255.255.0.0	255.255.0.0
LAN IP Address:	128.128.0.2	128.130.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0
PVCR IP Address:	0.0.0.0	0.0.0.0
PVCR IP Mask:	0.0.0.0	0.0.0.0

Table 2-4:

2.1.5 Multihomed Addressing in a Distributed Network

The advantage of multihomed IP addressing is particularly evident for complex networks involving several hops. All remote devices that can be accessed via the PVCR links of the local NetPerformer have the same network IP address. Far fewer network IP addresses are required.

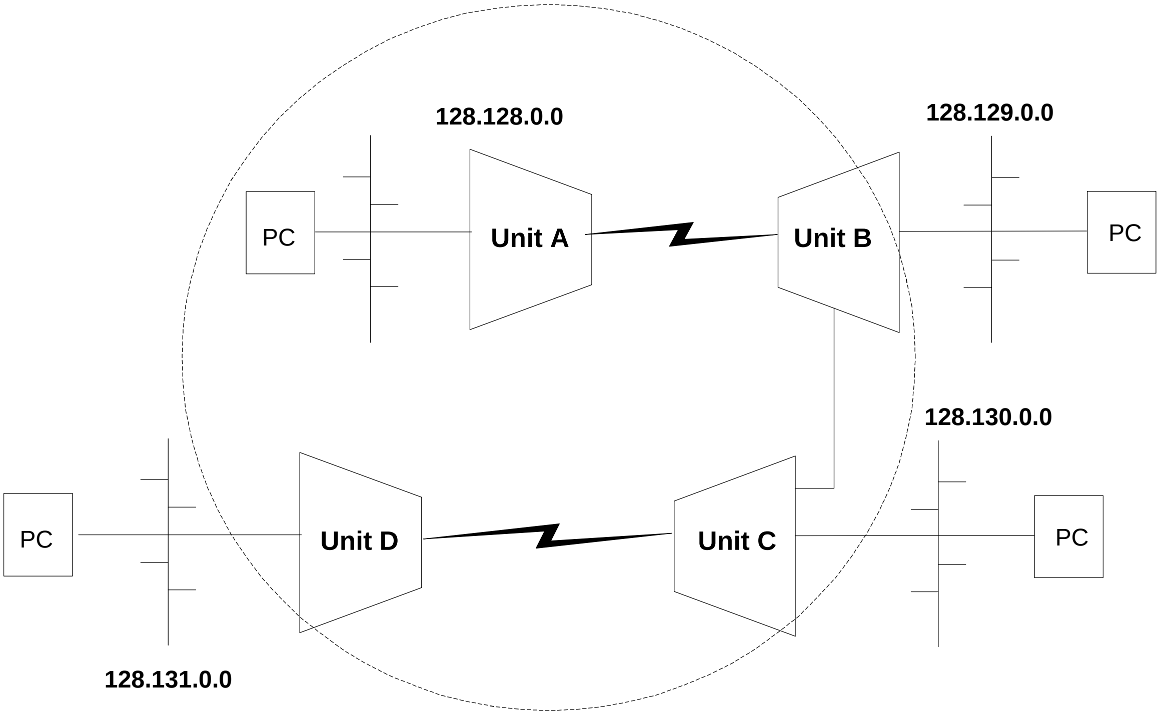


Figure 2-5:

Example addresses:

Parameter	Enterprise Unit “A” (Local)	Enterprise Unit “B”	Enterprise Unit “C”	Enterprise Unit “D”
IP Addr. of attached PC:	128.128.0.1	128.129.0.1	128.130.0.1	128.131.0.1
IP Mask of attached PC:	255.255.0.0	255.255.0.0	255.255.0.0	255.255.0.0
Default Gateway:	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0
Default IP Address:	128.128.0.10	128.128.0.11	128.128.0.12	128.128.0.13
Default IP Mask:	255.255.0.0	255.255.0.0	255.255.0.0	255.255.0.0
LAN IP Address:	128.128.0.2	128.129.0.2	128.130.0.2	128.131.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0	255.255.0.0	255.255.0.0
PVCR IP Address:	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0
PVCR IP Mask:	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0

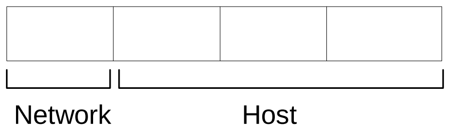
Table 2-5:

2.2 IP Subnetting and Supernetting

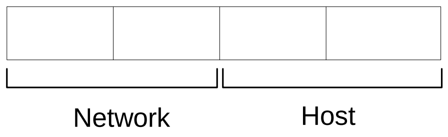
2.2.1 IP Subnetting

Each IP address has 4 bytes. If you are subdividing your network into several subnetworks, you can define which bytes define the network address and which define the host address:

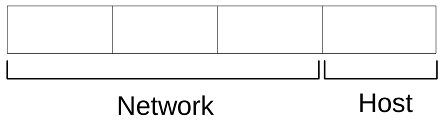
Class A (First Byte Values: 1 - 126):



Class B (First Byte Values: 128 - 191):



Class C (First Byte Values: 192 - 223):



For example, the Default IP Mask, defined using the Global Setup menu, identifies which bits of the Default IP Address correspond to the physical network, and which bits correspond to host identifiers. IP address 198.168.43.2 indicates a Class C address. The mask 255.255.255.0 has all network (or sub-network) bits set to 1 and all host bits set to 0. When this mask is applied to the IP address, the resulting network (or sub-network) address is 198.168.43.0.

When a NetPerformer is connected to a subnet, you define the multihomed IP address in the same way as for a regular network, using the correct subnet addresses.

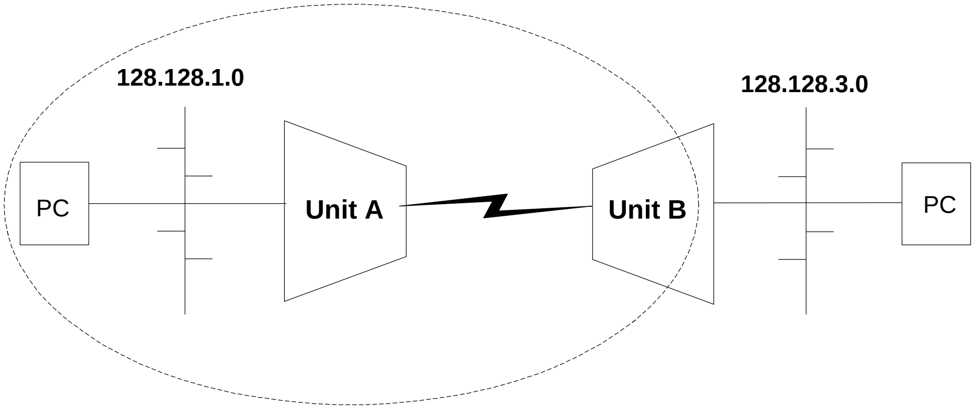


Figure 2-6:

Example addresses:

Parameter	Local Unit	Remote Unit
IP Addr. of attached PC:	128.128.1.1	128.128.3.1
IP Mask of attached PC:	255.255.255.0	255.255.255.0
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	128.128.1.10	128.128.1.11
Default IP Mask:	255.255.255.0	255.255.255.0
LAN IP Address:	128.128.1.2	128.128.3.2
LAN IP Mask:	255.255.255.0	255.255.255.0
PVCR IP Address:	0.0.0.0	0.0.0.0
PVCR IP Mask:	0.0.0.0	0.0.0.0

Table 2-6:

2.2.2 IP Supernetting

IP Supernetting is supported on all NetPerformer product types. With this addressing method, the network portion of the IP address can be split into two or more supernets, which reduces the size of the routing table. In terms of its functionality, supernetting is the opposite of subnetting.

The IP *Subnet mask* parameter determines how the supernet is defined. For example, if the *IP address* is set to **198.148.62.X**, you can use a *Subnet mask* of **255.255.254.0** to define the supernets **198.148.62.X** and **198.148.63.X**.

NOTE: To support IP Supernetting, the format of the *Subnet mask* parameter has been changed in NetPerformer V10.2:

- Enter the number of bits that are set to **1**, starting from the most significant bit of the first byte. For example, the value **23** is interpreted as **255.255.254.0**.
- The range of values for the *Subnet mask* parameter is now **0** to **32**, with a default value of **8**. At the console, the NetPerformer echoes back the value you enter in 4-byte dotted decimal format.

2.3 IP Multicast Addressing

An IP Multicast application typically has a server at the central site, and several client stations at multiple remote sites.

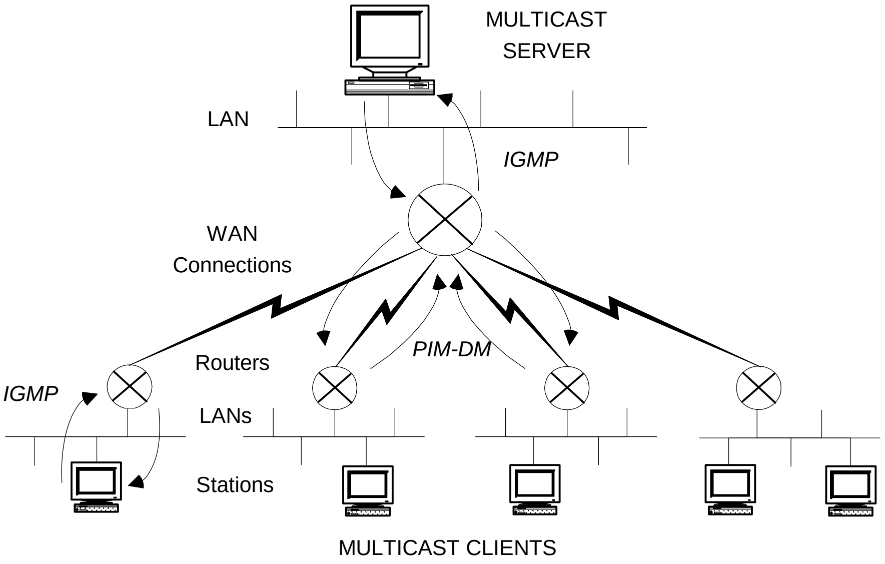


Figure 2-7: IP Multicast Network

2.3.1 IGMP and PIM2 Protocols

Two protocols are required to permit routing in an IP Multicast application:

- Internet Group Management Protocol (IGMP), which operates between a router and the LAN.
    - The NetPerformer can be configured to use either V1 or V2 of the IGMP protocol. V2 allows changes in group membership to be reported more quickly to the routing protocol. V1 provides compatibility with older networks.
    - Periodically (approximately every 2 minutes), the router sends a query for multicast members in the network. Membership queries may be used to determine which groups have members on an attached network, or whether a particular group has any members.
    - The LAN-attached stations respond to this query. In this way, the router knows which multicast groups have members.
    - IGMP does not route frames to a particular destination.
    - If a station comes up, it goes on the LAN and sends an IGMP Report frame that identifies its multicast group.
    - When IGMP V2 is used, the station also sends a leave message when the client quits the application.

- Protocol Independent Multicast Version 2 (PIM2), which operates between the routers.
  - The NetPerformer uses the PIM-DM protocol (Protocol Independent Multicast - Dense Mode). This is a routing algorithm designed for multicast groups that are densely distributed across the network.
  - Routers send PIM-DM HELLO frames to discover each other.
  - If the router receives a frame that carries a particular multicast destination for the first time, it will send the frame on all ports that have either:
    - ❑ a PIM-DM router attached on the remote side, *or*
    - ❑ at least one member on the LAN for the group that is specified in the multicast destination.

2.3.2 Queries and Messages

IGMP is an integral part of IP, and IGMP messages are encapsulated in IP datagrams. The NetPerformer uses IGMP to learn which groups have active members on the LAN. It keeps a list of multicast group memberships, that is, groups that have at least one member. It also keeps a delay timer for each membership. All IGMP messages are sent with the IP TTL (Time To Live) equal to 1.

If it is the highest router in the multicast hierarchy, the NetPerformer sends queries over the LAN port. Only one router can send queries at any one time. Queries may be:

- General, for soliciting membership information. For these queries the group address field is set to zero. The destination group is ALL-SYSTEMS. A General Query addressed to the all-systems multicast group is 224.0.0.1, with a group address field of 0.
- Group-specific, for determining whether a particular group has any members. The destination group is the group being queried. The IP multicast group address for this destination is held in the group address field of the IGMP message.

Two other message types may be sent using IGMP V2:

- Leave Group message, sent by the client when it quits the multicast group. The group address field contains the IP multicast group address of the group the client is leaving. The message is sent to all routers in the multicast network, 224.0.0.2.
- Membership Report, where the client responds to a membership query on an unsolicited transmission, informing the router that a new member is present in the group. The group address field contains the IP multicast group address of the group being reported.

2.3.3 Broadcast and Prune Mechanism

PIM-DM works on the principle that all downstream routers running PIM-DM want to receive multicast datagrams. When the first source and group address duo (S,G) is received, multicast datagrams are broadcast to all ports in the network. Responses from the downstream routers determine how subsequent transmissions are disseminated.

- **PRUNE state***:* If the response from a router indicates that its area of the network does not have any outgoing ports for this (S,G) duo, PIM-DM prunes off the forwarding branch.
  - A PRUNE message is data link multicast and IP addressed to the ALL-PIM-ROUTERS group address, 224.0.0.13.
  - PRUNE state is automatically disabled after a timer expires, allowing data to go down the branch once again.
- **FORWARD state***:* When a new member appears for an (S,G) duo for which there is no outgoing port, the router can send a GRAFT message to the upstream router for this source and group address.
  - A GRAFT message is unicast to the upstream Reverse Path Forwarding (RPF) neighbor.
  - A GRAFT turns a pruned branch back into FORWARD state. Together, all forwarding branches create a *source rooted tree*, which leads from the source to all members of the group.
  - PIM-DM initiates the FORWARD state in routers when a source begins to send. If the receiving router does not already have a forwarding entry, it creates it for the particular (S,G) duo.

2.3.4 Example Operation

In a multicast network:

- All designated routers on the LAN generate a General Query.
  - The query is addressed to the ALL-SYSTEMS multicast group, 224.0.0.1, and transported using IGMP.
  - Queries are sent approximately every 2 minutes.
  - Stations on a LAN send a Membership Report in response to a query, also using IGMP.
- Routers communicate with each other using PIM-DM.
  - PIM-DM Hello frames are sent every 30 seconds in both directions.

An example multicast network is shown in Figure 2-8.

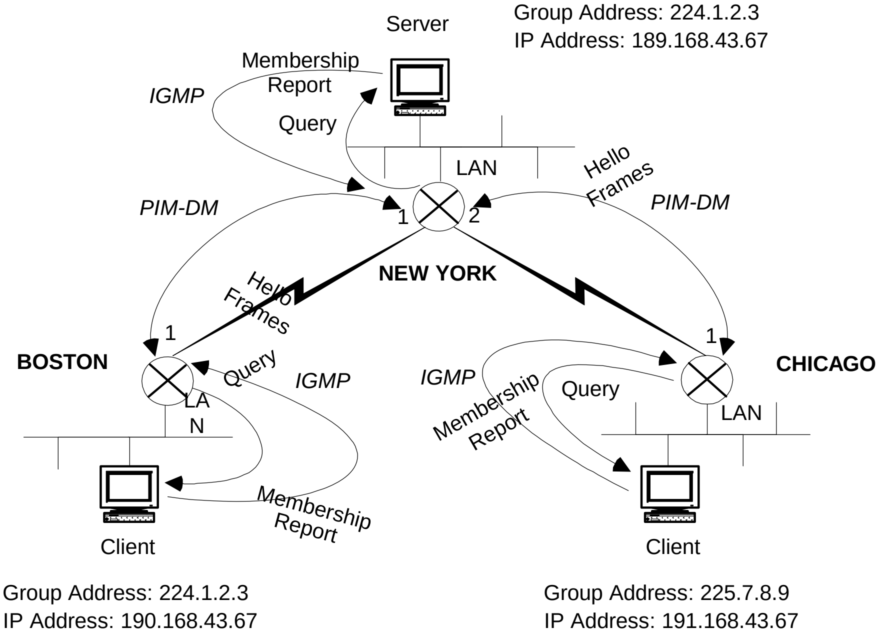


Figure 2-8: IP Multicast Application Example

- In this example, the server attached to the LAN at the NEW YORK site first sends a frame to group address 224.1.2.3. (Source: 189.168.43.67, Destination: 224.1.2.3)
- The router NEW YORK receives this frame, and ascertains that it does not have an entry for this address in its routing table.

  It creates an entry in the GROUP multicast routing table with the following information: Group: 224.1.2.3, Source: 189.168.43.67, Incoming interface: LAN.
- Since this is the first outgoing frame to this particular destination, the router NEW YORK examines all available ports to select the outgoing interface.
  - If the router determines that there is a PIM-DM router at the other end of a port, it puts that port number in its list of outgoing interfaces.
  - In this example, NEW YORK port 1 leads to router BOSTON, and port 2 leads to CHICAGO. Thus NEW YORK puts both 1 and 2 on the list of outgoing interfaces. The GROUP multicast routing table entry for the new destination is:

```
Group:  224.001.002.003
Source: 189.168.043.067    upstream neighbor: 000.000.000.000
Incoming interface: LAN    # frames received: 1
Outgoing interface: 1      # frames transmitted: 1
Outgoing interface: 2      # frames transmitted: 1
```

- NEW YORK sends the frame via port 1 to the router BOSTON, and via port 2 to the router CHICAGO.
- BOSTON has only a LAN connection, with no PIM-DM router at the other end. However, a client for group 224.1.2.3 is attached to the LAN.
  - BOSTON creates a routing table entry with the following information:

```
Group:  224.001.002.003
Source: 189.168.043.067    upstream neighbor: 005.010.002.300
Incoming interface: 1      # frames received: 1
Outgoing interface: LAN    # frames transmitted: 1
```

  - BOSTON sends the frame over the LAN connection to the client.
- The router CHICAGO at the remote end of NEW YORK port 2 has no client for 224.1.2.3, and no PIM-DM router at the other end of the LAN.
  - CHICAGO sends a PRUNE message back to NEW YORK.
  - When NEW YORK receives the PRUNE message, it removes “2” from its list of outgoing interfaces, and adds “2” to its list of pruned interfaces.

```
Group:  224.001.002.003
Source: 189.168.043.067    upstream neighbor: 000.000.000.000
Incoming interface: LAN    # frames received: 1
Outgoing interface: 1      # frames transmitted: 1
Pruned interface:   2      Holdtime (sec): 1
```

2.3.5 Configuring an IP Multicast Application

NetPerformer configuration for IP Multicast transmissions involves both LAN and WAN (PVCR) port parameters. To ensure correct routing, you must not only configure the multicast parameters for all LAN ports, but also for any WAN PVCR ports, PVCR PVCs and RFC1490 PVCs that may participate in IP multicast transmissions, as follows:

- For each LAN connection that participates in the multicast network:

```
PORT ETH 1> IGMP enable (def:NO) ? YES
PORT ETH 1> IGMP version (1-2,def:2) ?
PORT ETH 1> IGMP send report (def:NO) ?
PORT ETH 1> IP multicast active (NO/YES,def:NO) ? YES
PORT ETH 1> IP multicast protocol (NONE/PIMDM,def:NONE) ? PIMDM
PORT ETH 1> IP multicast 1 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 2 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 3 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 4 (def:000.000.000.000) ?
```

  For detailed descriptions of these parameters, refer to the appendix “SE/PORT/ETH Configuration Parameters” on page 8-1.

- For each WAN (PVCR) port that participates in the multicast network:

```
PORT 1> IP multicast active (def:NO) ? YES
PORT 1> IP multicast protocol (def:NONE) ? PIMDM
```

For detailed descriptions of these parameters, refer to the appendix *SE/PORT/#/PVCR Configuration Parameters* in the *WAN/Leased Lines* fascicle of this document series.

- For each PVCR and RFC1490 PVC that participates in the multicast network:

```
PVC 1> IP multicast active (def:NO) ? YES
PVC 1> IP multicast protocol (def:NONE) ? PIMDM
```

For detailed descriptions of these parameters, refer to the appendix *SE/PVC Configuration Parameters* in the *WAN/Frame Relay* fascicle of this document series.

2.3.6 Verifying the IP Multicast Connections

The Display Routing Table (**DR**) command includes a display option for viewing the IP Multicast routing table. Refer to “IP Multicast Routing Table” on page 2-32.

2.4 Configuring the IP Connections

As mentioned in “Multihomed IP Addressing” on page 2-2 and “IP Multicast Addressing” on page 2-12, several areas of the NetPerformer console are involved when configuring an IP connection:

- Global parameters (refer to the chapter *Global Functions* in the *Quick Configuration* fascicle of this document series):

```
GLOBAL> Default IP address (def:000.000.000.000) ?
GLOBAL> Default IP mask (number of bits) (0-32,def:0) ?
{000.000.000.000}
GLOBAL> Default gateway (def:000.000.000.000) ?
```

- PVCR port parameters (refer to the *WAN/Leased Lines* fascicle of this document series):

```
PORT 1> IP address (def:000.000.000.000) ?
PORT 1> Subnet mask (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT 1> IP RIP (def:V1) ? V2 MULTICAST
PORT 1> IP RIP TX/RX (def:DUPLEX) ?
PORT 1> IP RIP Authentication (def:NONE) ?
PORT 1> IP RIP Password (def:) ?
PORT 1> OSPF (def:DISABLE) ? ENABLE
PORT 1> OSPF Area ID (def:000.000.000.000) ?
PORT 1> OSPF Transit delay (1-360,def:1) ?
PORT 1> OSPF Retransmit interval (1-360,def:5) ?
PORT 1> OSPF Hello interval (1-360,def:10) ?
PORT 1> OSPF Dead interval (1-2000,def:40) ?
PORT 1> OSPF Password (def:) ?
PORT 1> OSPF Metric cost (1-65534,def:10) ?
PORT 1> IP multicast active (def:NO) ? YES
PORT 1> IP multicast protocol (def:NONE) ? PIMDM
PORT 1> NAT enable (def:NO) ? YES
PORT 1> NAT rule (1-10) (def:) ?
PORT 1> NAT side (def:INTERNAL) ?
```

- PPP port parameters (refer to the *WAN/Point-to-Point (PPP)* fascicle of this document series):

```
PORT 1> IP address (def:000.000.000.000) ?
PORT 1> Subnet mask (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT 1> IP RIP (def:V1) ? V2 MULTICAST
PORT 1> IP RIP TX/RX (def:DUPLEX) ?
PORT 1> IP RIP Authentication (def:NONE) ?
PORT 1> IP RIP Password (def:) ?
PORT 1> NAT enable (def:NO) ? YES
PORT 1> NAT rule (1-10) (def:) ?
PORT 1> NAT side (def:INTERNAL) ?
```

- PVCR PVC parameters (refer to the *WAN/Frame Relay* fascicle of this document series):

```
PVC 1> IP address (def:000.000.000.000) ?
PVC 1> Subnet mask (number of bits) (0-32,def:8) ?    {255.000.000.000}
PVC 1> NAT enable (def:NO) ? YES
PVC 1> NAT rule (1-10) (def:) ?
PVC 1> NAT side (def:INTERNAL) ?
PVC 1> IP RIP (def:V1) ? V2 BROADCAST
PVC 1> IP RIP TX/RX (def:DUPLEX) ?
PVC 1> IP RIP Authentication (def:NONE) ?
PVC 1> IP RIP Password (def:) ?
PVC 1> OSPF (def:DISABLE) ? ENABLE
PVC 1> OSPF Area ID (def:000.000.000.000) ?
PVC 1> OSPF Transit delay (1-360,def:1) ?
PVC 1> OSPF Retransmit interval (1-360,def:5) ?
PVC 1> OSPF Hello interval (1-360,def:10) ?
PVC 1> OSPF Dead interval (1-2000,def:40) ?
PVC 1> OSPF Password (def:) ?
PVC 1> OSPF Metric cost (1-65534,def:10) ?
PVC 1> IP multicast active (def:NO) ?
PVC 1> IP multicast protocol (def:NONE) ?
```

- RFC1490 PVC parameters (refer to the *WAN/Frame Relay* fascicle of this document series):

```
PVC 2> IP address (def:000.000.000.000) ?
PVC 2> Subnet mask (number of bits) (0-32,def:8) ?    {255.000.000.000}
PVC 2> NAT enable (def:NO) ? YES
PVC 2> NAT rule (1-10) (def:) ?
PVC 2> NAT side (def:INTERNAL) ?
PVC 2> Frame size (128-8192,def:1500) ?
PVC 2> IP RIP (def:V1) ? V2 MULTICAST
PVC 2> IP RIP TX/RX (def:DUPLEX) ?
PVC 2> IP RIP Authentication (def:NONE) ?
PVC 2> IP RIP Password (def:) ?
PVC 2> OSPF (def:DISABLE) ? ENABLE
PVC 2> OSPF Area ID (def:000.000.000.000) ?
PVC 2> OSPF Transit delay (1-360,def:1) ?
PVC 2> OSPF Retransmit interval (1-360,def:5) ?
PVC 2> OSPF Hello interval (1-360,def:10) ?
PVC 2> OSPF Dead interval (1-2000,def:40) ?
PVC 2> OSPF Password (def:) ?
PVC 2> OSPF Metric cost (1-65534,def:10) ?
PVC 2> IP multicast active (def:NO) ?
PVC 2> IP multicast protocol (def:NONE) ?
...
PVC 2> BRG connection (def:NO) ?
PVC 2> IP  connection (def:YES) ?
```

- The LAN port parameters, which are configured using the **ETH**, **ETH1** or **ETH2** option of the **SETUP/PORT** submenu. Refer to the chapter “Ethernet LAN Connection” on page 1-1.

- The IP parameters, which are configured using the **IP** option of the **SETUP** command:

Console	SNMP	Text-based Config
SE/IP	*ip* (category)	[ip] (heading)

The Setup IP menu lets you control IP routing on the NetPerformer. For SNMP, the IP configuration variables are found under the *ip* category. For text-based configuration the [ip] heading is used. At the console, enter the menu sequence: **SE ↵ IP** to reach the Setup IP menu. A second *Item* prompt is displayed to select one of the IP submenus, as shown below.

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ?
...
```

The IP submenus control the following:

- **GLOBAL**: To configure the unit for general IP routing characteristics (see next section)
- **STATIC**: To configure the parameters required to enable IP routing to a particular destination when RIP is disabled on the NetPerformer (see "Configuring the Static IP Parameters" on page 2-21)
- **SOURCE-STATIC**: To configure the parameters required to enable IP routing to a particular destination using the source IP address rather than the destination IP address (see "Configuring the SOURCE-STATIC Parameters" on page 2-23)
- **BOOTP**: To configure the parameters required to set up the BOOTP relay agent function (see "Configuring the BOOTP Parameters" on page 2-23)
- **OSPF**: To configure the NetPerformer to support a network that uses the OSPF routing protocol. Refer to the chapter "OSPF Network Support" on page 7-1 for details on this submenu.
- **TIMEP**: To configure the parameters required to resynchronize the real time clock (see "Configuring the TIMEP Parameters" on page 2-25)
- **SNMP**: To configure the SNMP parameters (see "Configuring the SNMP Parameters" on page 2-28)
- **NAT:** To configure the NAT parameters. Refer to the chapter "Network Address Translation (NAT)" on page 6-1 for details on this submenu.
- **TELNET:** To restrict or disable TELNET access to the NetPerformer console. For details on this submenu, refer to the chapter *Controlling Access to the NetPerformer* in the *Quick Configuration* fascicle of this document series.

- **FTP:** To restrict or disable FTP access to the NetPerformer console. This is detailed in the chapter *Controlling Access to the NetPerformer* in the *Quick Configuration* fascicle.
- **DNS:** To configure the NetPerformer as a DNS client for resolving the IP address of a server based on its domain name (see the chapter "DNS Address Resolution" on page 5-1).
- **RADIUS:** To configure the NetPerformer for Remote Authentication Dial-In User Service (RADIUS) authentication of console and Telnet access to the network. Refer to the chapter *Controlling Access to the NetPerformer* in the *Quick Configuration* fascicle.

2.4.1 Configuring the Global IP Parameters

To configure the Global IP parameters using the NetPerformer console, enter the menu sequence: **SE** ↵ **IP** ↵ **GLOBAL**. For SNMP, the *npip* category includes all variables affecting configuration of global IP parameters. For text-based configuration the [npip] heading is used.

Console	SNMP	Text-based Config
SE/IP/GLOBAL	*npip* (category)	[npip] (heading)

Configurable IP parameters in this submenu include the following:

SE/IP/GLOBAL example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? GLOBAL
IP> Router (def:ENABLE) ?
IP> Route broadcast to end station (def:NO) ?
IP> OSPF AS boundary router (def:NO) ? YES
IP> RIP to OSPF metric conversion cost (1-65534,def:2000) ?
IP> OSPF AS forwards RIP entries (def:YES) ?
IP> OSPF AS forwards STATIC entries (def:YES) ?
IP> RIP AS boundary router (def:NO) ?
IP> IP Precedence for FR over IP (0-7,def:0) ?
IP> Allow LAN-to-LAN routing (def:YES) ?
```

These parameters are detailed in the appendix "SE/IP Configuration Parameters" on page 9-1.

2.4.2 Configuring the Static IP Parameters

If you disable IP RIP on the NetPerformer but require IP routing to a particular destination, you can configure a static IP address entry using the IP Static menu. Each entry is based on the IP address of the remote NetPerformer and the network address of the destination device.

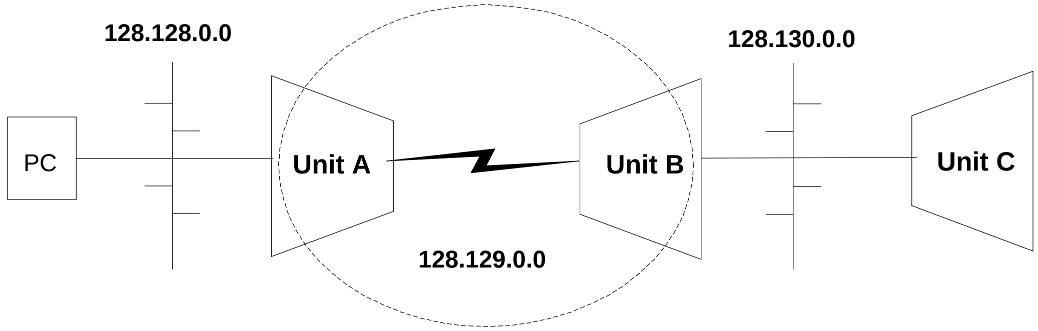


Figure 2-9:

Parameter	Local Unit	Remote Unit
IP Addr. of attached device:	128.128.0.1	128.130.0.1
IP Mask of attached device:	255.255.0.0	255.255.0.0
Default Gateway:	0.0.0.0	0.0.0.0
Default IP Address:	0.0.0.0	0.0.0.0
Default IP Mask:	0.0.0.0	0.0.0.0
LAN IP Address:	128.128.0.2	128.130.0.2
LAN IP Mask:	255.255.0.0	255.255.0.0
PVCR IP Address:	128.129.0.1	128.129.0.2
PVCR IP Mask:	255.255.0.0	255.255.0.0

Table 2-7: Examples of Static IP addresses

The Setup IP Static menu lets you configure all parameters required to enable IP routing to a particular destination when IP RIP is disabled on the NetPerformer. For example, a PC attached to the local NetPerformer may be used for SNMP management of all devices attached to a remote LAN. To do this, the local unit must be able to route information across the remote LAN even if IP RIP is disabled.

NOTE: The *Router* parameter of the Setup IP Global menu must be enabled to permit routing the required TCP/IP frames for this application.

Console	SNMP	Text-based Config
SE/IP/STATIC	*ipstatic* (category)	[ipstatic] (heading)

To configure an IP Static address entry from the console, enter the menu sequence: **SE ↵ IP ↵ STATIC**. For SNMP, the *ipstatic* category includes all variables affecting configuration of an IP Static entry. For text-based configuration the [ipstatic] heading is used. The following IP Static parameters are listed on the console*:*

SE/IP/STATIC example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? STATIC
IP static entry number (1-200,def:1) ?
IP STATIC 1> Valid (def:NO) ? YES
IP STATIC 1> Destination address (def:000.000.000.000) ? 128.130.0.0
IP STATIC 1> Subnet mask (number of bits) (0-32,def:8) ?  16 {255.255.000.000}
IP STATIC 1> Next hop (def:000.000.000.000) ? 128.129.0.1
```

For details on these parameters, turn to "SE/IP/STATIC Submenu" on page 9-6.

2.4.3 Configuring the SOURCE-STATIC Parameters

Source Static Routing allows frames to be routed from the source IP address rather than the destination IP address. A new parameter set has been added to the console to control this feature.

To configure a source-static address entry from the console, enter the menu sequence: **SE ↵ IP ↵ SOURCE-STATIC**.

2.4.4 Configuring the BOOTP Parameters

The Setup IP BOOTP menu lets you configure all IP parameters that are required to set up the BOOTP relay agent function. The BOOTP protocol allows a computer to determine its IP address, subnet mask, gateway address and other IP characteristics from information carried within the BOOTP/DHCP frame.

NOTE: The *Router* parameter of the Setup IP Global menu must be enabled to permit routing the required TCP/IP frames for this application.

Console	SNMP	Text-based Config
SE/IP/BOOTP	*bootp* (category)	[bootp] (heading)

To configure the BOOTP parameters from the console, enter the menu sequence: **SE ↵ IP ↵ BOOTP**. For SNMP, the *bootp* category includes all variables affecting configuration of the BOOTP parameters. For text-based configuration the *[bootp]* heading is used. The following BOOTP parameters are listed on the console:

SE/IP/BOOTP example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ?
BOOTP> BOOTP (def:DISABLE) ? ENABLE
BOOTP> Max hops (0-16,def:4) ?
BOOTP> Destination IP address 1 (def:000.000.000.000) ? 192.68.20.1
BOOTP> Destination IP address 2 (def:000.000.000.000) ?
BOOTP> Destination IP address 3 (def:000.000.000.000) ?
BOOTP> Destination IP address 4 (def:000.000.000.000) ?
```

For details on these parameters, consult the section "SE/IP/BOOTP Submenu" on page 9-8.

BOOTP Counters

The NetPerformer keeps statistics on BOOTP frame forwarding when it is configured as a relay agent. The Display Counters command allows you to view these statistics.

Console	SNMP
DC/BOOTP	*statBootp* (category)

To access the BOOTP counters from the console, enter **DC** on the command line, and then select **BOOTP**. For SNMP, use the *statBootp* category.

DC/BOOTP example

```
SDM-9380>DC
DISPLAY COUNTERS
Item (BOOTP/CONFIG/DNS/IP/NAT/PORT/PVC/Q922/Q933/QOS/SLOT/SVC/TIMEP,
def:IP) ? BOOTP
Number of BOOTREQUEST frames received...........0
Number of BOOTREQUEST frames sent...............0
Number of BOOTREPLY frames received.............0
Number of BOOTREPLY frames sent.................0
```

Details on the BOOTP counters can be found in "BOOTP Statistics" on page 11-22.

BOOTP Errors

The NetPerformer keeps statistics on errors that may occur during BOOTP operations. The Display Errors command allows you to view these statistics. Use the Reset Counters (**RC**) command to return the error counters to zero.

Console	SNMP
DE/BOOTP	*statBootp* (category)

To display the errors from the console, enter **DE** on the command line and select **BOOTP**. A display like the following will appear on the screen:

DE/BOOTP example

```
SDM-9380>DE
DISPLAY ERRORS
Item (BOOTP/CHANNEL/DICT/GROUP/NAT/PORT/PU/PVC/Q922/SLOT/SVC/TIMEP,
def:DICT) ? BOOTP
Reply with invalid giaddr......................0
Hops limit exceeded............................0
Request received on port bootpc................0
Invalid op code field..........................0
Cannot route frame.............................0
Frame too small to be a BOOTP frame............0
Reply received on port bootpc..................0
Cannot receive and forward on the same port.....0
```

These statistics can be used to diagnose problems that take place when the NetPerformer forwards or receives BOOTP frames. For details, turn to “DE/BOOTP” on page 11-22.

2.4.5 Configuring the TIMEP Parameters

The Setup IP TIMEP menu lets you configure all IP parameters that are required to resynchronize the real-time clock. The Time Protocol implemented in the NetPerformer adheres to RFC 868. It includes a time CLIENT that operates under UDP or TCP, and a time SERVER that operates under UDP, TCP or both. The NetPerformer Time Protocol function is often referred to as TIMEP.

On start-up, the NetPerformer will try to interrogate the time server for up to 20 times, at brief intervals. If after 20 attempts, the time client has not synchronized, the NetPerformer reverts to its normal interrogation pattern. This avoid wasting bandwidth if the server is not online.

You can manually cause the NetPerformer to update its clock by using the Update Time (**UT**) command. Refer to “Update Time (UT) Command” on page 2-26.

NOTE: If the NetPerformer LAN port is turned off, TIMEP in TCP mode will work properly once the routing tables are updated. Allow about 5 minutes for this update. The UDP client starts working immediately. When the LAN port is off, the Global IP Address is used, as long as it is not null (**0.0.0.0**).

Console	SNMP	Text-based Config
SE/IP/TIMEP	*timep* (category)	[timep] (heading)

To configure the TIMEP parameters from the console, enter the menu sequence: **SE** ↵ **IP** ↵ **TIMEP**. For SNMP, the *timep* category includes all variables affecting configuration of the TIMEP parameters. For text-based configuration the *[timep]* heading is used. The following TIMEP parameters are listed on the console:

SE/IP/TIMEP example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? TIMEP
TIMEP> Negative time zone (def:YES) ?
TIMEP> Time zone offset from GMT (min) (0-720,def:300) ?
TIMEP> Time server protocol (NONE/UDP/TCP/BOTH,def:NONE) ? BOTH
TIMEP> Time client protocol (NONE/UDP/TCP,def:NONE) ? TCP
TIMEP> Time client server IP address (def:000.000.000.000) ?
TIMEP> Time client update interval (min) (1-65534,def:1440) ?
TIMEP> Time client UDP timeout (s) (1-255,def:20) ?
TIMEP> Time client UDP retransmissions (0-255,def:3) ?
```

Details on these parameters can be found in the section “SE/IP/TIMEP Submenu” on page 9-23.

2.4.6 Update Time (UT) Command

The Update Time (**UT**) command causes the NetPerformer to update its clock using the TIMEP parameters. All TIMEP parameters must be properly defined for the Update Time command to work.

To execute the Update Time command, enter **UT** at the console prompt. The console responds with a message that indicates the status of the time update.

- When the time update request is successfully executed:

UT example: successful execution

```
SDM-9230>UT
UPDATE TIME FROM SERVER
Time update requested. Check time in 1 minute
```

- When the time update request cannot be executed because the TIMEP client is inactive:

UT example: inactive TIMEP client

```
SDM-9230>UT
UPDATE TIME FROM SERVER
No effect. TIMEP client is inactive
```

- When the time update request cannot be executed because no IP address has been defined for the TIMEP server:

UT example: undefined IP address

```
SDM-9230>UT
UPDATE TIME FROM SERVER
No effect. No IP address for TIMEP server
```

Use the Display Time (**DT**) command to view the effect of the time update.

DT example: before UT

```
SDM-9230>DT
DISPLAY TIME
Time> TUE   2002/10/22 15:45:45
```

DT example: after UT

```
SDM-9230>DT
DISPLAY TIME
Time> WED   2003/10/22 15:46:25
```

2.4.7 TIMEP Counters

Console	SNMP
DC/TIMEP	*statTime* (category)

The NetPerformer keeps statistics on TIMEP frame forwarding when it is configured as a relay agent. The Display Counters command allows you to view these statistics.

To access the TIMEP counters from the console, enter **DC** on the command line, and then select **TIMEP**. For SNMP, use the *statTime* category.

DC/TIMEP example

```
SDM-9380>DC
DISPLAY COUNTERS
Item (BOOTP/CONFIG/DNS/IP/NAT/PORT/PVC/Q922/Q933/QOS/SLOT/SVC/TIMEP,
def:QOS) ? TIMEP
Number of frames received......................1608
Number of frames sent..........................97
Server’s number of requests received...........0
Server’s number of replies sent................0
Client’s number of requests sent...............6
Client’s number of replies received............6
```

Details on the TIMEP counters are provided in the section “DC/TIMEP” on page 11-25.

2.4.8 TIMEP Errors

The NetPerformer keeps statistics on errors that may occur during TIMEP operations. The Display Errors command allows you to view these statistics. Use the Reset Counters (**RC**) command to return the error counters to zero.

Console	SNMP
DE/TIMEP	*statTime* (category)

To display the errors from the console, enter **DE** on the command line and select **TIMEP**.

DE/TIMEP example

```
SDM-9380>DE
DISPLAY ERRORS
Item (BOOTP/CHANNEL/DICT/GROUP/NAT/PORT/PU/PVC/Q922/SLOT/SVC/TIMEP,
def:Q922) ? TIMEP
Client’s number of retransmissions.............0
Client’s number of sync failures...............0
Invalid local IP address.......................0
Frames received with invalid port number........0
```

These statistics can be used to diagnose problems that take place when the NetPerformer sends or receives TIMEP frames. For details, turn to “DE/TIMEP” on page 11-26.

2.4.9 Configuring the SNMP Parameters

The Setup SNMP menu lets you configure all parameters that govern the requests and traps that the SNMP agent can process on the NetPerformer. It is available from the console only. To configure the SNMP parameters, enter the menu sequence: **SE** ↵ **IP** ↵ **SNMP**.

Console	SNMP
SE/IP/SNMP	(not available)

SE/IP/SNMP example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? SNMP
SNMP> Get community (def:PUBLIC) ?
SNMP> Set community (def:PUBLIC) ?
SNMP> Trap community (def:PUBLIC) ?
```

NOTE: **In order to access the SNMP agent, these parameters must be previously set using the console interface.** Refer to the *Getting Started* fascicle of this document series.

2.5 IP Connection Status

Several commands are available from the console to view the status of IP connections on the NetPerformer.

- The Display Routing Table (**DR**) command displays RIP/OSPF, Multihomed and Source-static IP routing tables (see next section)
- Display ARP Cache (**ARP**) provides information on the NetPerformer ARP table (see "Displaying the ARP Cache" on page 2-34)
- Display Counters (**DC**) provides statistical information concerning IP connections (see "IP Counters" on page 2-35), BOOTP (see "BOOTP Counters" on page 2-24) and TIMEP frame forwarding (see "TIMEP Counters" on page 2-27)
- The Display Errors (**DE**) command provides BOOTP errors (see "BOOTP Errors" on page 2-24) and TIMEP errors (see "TIMEP Errors" on page 2-27)
- The Ping Remote Unit (**PING**) command can be used to test the current status of gateway connections (see "PING Remote Unit Command" on page 2-36).

To access IP statistics and routing tables using SNMP, open the Standard MIB (MIB II) and bring up the *ip* category under the *mgmt* section.

2.5.1 IP Routing Tables

Console	SNMP
DR/IP	MIB II - *mgmt* - ipRouteTable

The IP option of the Destination Routing Table command provides the current status of connections via the NetPerformer and other routers using RIP (Routing Information Protocol) IP. To view these tables, the *RIP* parameter must be enabled on one or more ports or PVCs.

NOTE: To view the status of remote IP connection, the *Router* parameter of the Setup IP Global menu must also be enabled.

To access the IP tables using SNMP, open the Standard MIB (MIB II) and bring up the *ipRouteTable* table under the *mgmt* section. To access these tables from the console, enter **DR** on the command line, and then select **IP**. Two types of routing tables are available:

- **UNICAST:** Includes the IP RIP/OSPF, multihomed and source-static routing tables
- **MULTICAST**: Includes the Group and IGMP routing tables.

2.5.2 IP RIP Routing Table

To view the IP RIP table from the console:

- Enter **DR** at the NetPerformer command line prompt
- Enter **IP** at the first *Item* prompt of the Display Routing Table command
- Enter **UNICAST** at the second *Item* prompt
- Enter **RIP** at the third *Item* prompt.

DR/IP/ UNICAST/RIP example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IP) ?
Item (UNICAST/MULTICAST,def:UNICAST) ?
Item (RIP/MULTIHOMED/SOURCE-STATIC,def:RIP) ?

The routing table has 14 entry(ies)

DESTINATION   VAL COST INTRF   NEXT HOP        AGE  MASK            TYPE PROT

000.000.000.000 Y 4    LAN-1   007.248.000.001 24 s 000.000.000.000 DGTW RIP
007.000.000.000 Y 0    LAN-1   007.248.000.001 0  s 255.000.000.000 NET  local
007.000.001.000 Y 2    LAN-1   007.248.000.001 24 s 255.255.255.000 SUB  RIP
007.020.020.000 Y 15   PVC   1 000.000.000.000 23 s 255.255.255.000 SUB  RIP
007.248.000.000 Y 0    LAN-1   007.248.000.001 0  s 255.255.255.000 SUB  LOCAL
007.248.000.001 Y 0    LAN-1   007.248.000.001 0  s 255.255.255.255 HOST LOCAL
007.248.001.000 Y 2    PVC   1 000.000.000.000 26 s 255.255.255.000 SUB  RIP
007.248.002.000 Y 1    PVC   1 000.000.000.000 26 s 255.255.255.000 SUB  RIP
007.251.003.000 Y 2    PVC   1 000.000.000.000 26 s 255.255.255.000 SUB  RIP
004.000.000.000 Y 7    LAN-1   007.248.000.001 27 s 255.000.000.000 NET  RIP
010.000.000.000 Y 4    LAN-1   007.248.000.001 27 s 255.000.000.000 NET  RIP
194.236.000.000 Y 3    LAN-1   007.248.000.001 27 s 255.255.255.000 NET  RIP
194.236.007.000 Y 4    LAN-1   007.248.000.001 27 s 255.255.255.000 NET  RIP
196.236.057.000 Y 4    LAN-1   007.248.000.001 27 s 255.255.255.000 NET  RIP
```

The RIP IP routing table entries are detailed in the section "DR/IP/UNICAST/RIP" on page 11-13.

2.5.3 Multihomed IP Routing Table

Multihomed IP addresses can be displayed separately from RIP IP addresses, using the Display Routing Table (**DR**) command at the NetPerformer console.

NOTE: The Multihomed routing table is not available on a VoIP Gateway product with SIP, since it is never used for a routing decision on these products.

To view the multihomed addresses:

- Enter **DR** at the NetPerformer command line prompt
- Enter **IP** at the *Item* prompt
- Enter **UNICAST** at the second *Item* prompt
- Enter **MULTIHOMED** (or an abbreviation of this term) at the third *Item* prompt.

DR/IP/UNICAST/MULTIHOMED example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IPX) ? IP
Item (UNICAST/MULTICAST,def:UNICAST) ?
Item (RIP/MULTIHOMED/SOURCE-STATIC,def:RIP) ? MULTIHOMED

DESTINATION      MASK            VALID METRIC   INTRF   TTL

007.248.000.000  255.255.255.000   Y      0      LAN-1   10 m
007.248.000.002  255.255.255.000   Y      0      LAN-1   10 m
007.248.219.002  255.255.255.000   Y      0      PVC   6 10 m
007.248.220.001  255.255.255.000   Y      1      PVC   5 10 m
007.248.220.002  255.255.255.000   Y      0      PVC   5 10 m
```

The **DESTINATION**, **MASK**, **VALID**, **METRIC** and **INTRF** columns contain the same type of information as is found in the RIP IP routing table, described earlier. The **Age** statistic is replaced by the **TTL** statistic (see “DR/IP/UNICAST/MULTIHOMED” on page 11-15).

2.5.4 Source-static IP Routing Table

NOTE: A source-static IP routing application is described on “Source-static Routing” on page 8-19.

To view the source-static routing table from the console:

- Enter **DR** at the NetPerformer command line prompt
- Enter **IP** at the *Item* prompt
- Enter **UNICAST** at the second *Item* prompt
- Enter **SOURCE-STATIC** (or an abbreviation of this term) at the third *Item* prompt.

DR/IP/UNICAST/SOURCE-STATIC example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IPX) ? IP
Item (UNICAST/MULTICAST,def:UNICAST) ?
Item (RIP/MULTIHOMED/SOURCE-STATIC,def:RIP) ? SOURCE-STATIC
```

2.5.5 IP Multicast Routing Table

Multicast IP addresses are displayed separately from RIP IP addresses. For Multicast IP, different information is presented in GROUP and IGMP routing table displays.

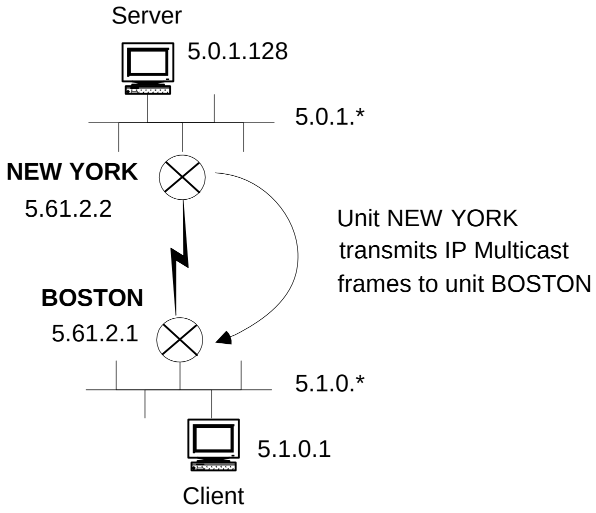


Figure 2-10: Scenario for Multicast Routing Table

To view the multicast addresses:

- Enter **DR** at the NetPerformer command line prompt
- Enter **IP** at the *Item* prompt
- Enter **MULTICAST** at the second *Item* prompt
- Choose the IP Multicast routing table type:
  - **GROUP:** To display the group addresses, source of the IP multicast traffic, the nearest upstream neighbor, physical interfaces used and the number of frames transmitted and received for each group.
  - **IGMP:** To display the IP Multicast information that is passed between this router and the LAN.

DR/IP/MULTICAST/GROUP example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IP) ?
Item (UNICAST/MULTICAST,def:UNICAST) ? MULTICAST
Item (GROUP/IGMP,def:GROUP) ? GROUP

Group:  225.000.001.001
Source: 005.000.001.128    upstream neighbor: 005.061.002.002
Incoming interface: 2      # frames received: 10958
Outgoing interface: LAN    # frames transmitted: 10958


Group:  225.000.002.002
Source: 005.000.001.128    upstream neighbor: 005.061.002.002
```

```
Incoming interface: 2      # frames received: 10422
Outgoing interface: LAN    # frames transmitted: 10422


Group:  225.000.003.003
Source: 005.000.001.128    upstream neighbor: 005.061.002.002
Incoming interface: 2      # frames received: 10329
Outgoing interface: LAN    # frames transmitted: 10329


Group:  225.000.004.004
Source: 005.000.001.128    upstream neighbor: 005.061.002.002
Incoming interface: 2      # frames received: 10206
Outgoing interface: LAN    # frames transmitted: 10206
```

DR/IP/MULTICAST/IGMP example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IP) ?
Item (UNICAST/MULTICAST,def:MULTICAST) ?
Item (GROUP/IGMP,def:GROUP) ? IGMP

ADDRESS         INTRF  AGE   TTL   LAST REPORTER

225.000.001.001 LAN    399 s 234 s 005.001.000.001
225.000.002.002 LAN    407 s 238 s 005.001.000.001
225.000.003.003 LAN    403 s 231 s 005.001.000.001
225.000.004.004 LAN    402 s 232 s 005.001.000.001
```

For details on these statistics, turn to “DR/IP/MULTICAST” on page 11-16.

2.5.6 Displaying the ARP Cache

The Display ARP Cache command provides a means of viewing and clearing the NetPerformer ARP table. This command is available from the console only.

Console	SNMP
ARP	(not available)

An ARP request is sent to the ARP cache one minute before the Time To Live (TTL) ends if the NetPerformer needs to send a packet to a specific MAC address. This ensures that the entry for this MAC address will not be erased from the ARP table. It also prevents traffic bursts when ARP entries are renewed.

The **DISPLAY** operation of the **ARP** command includes the port index of the LAN interface. This distinguishes entries in the ARP cache with the same destination address, and resolves a problem with SNMP.

ARP/DISPLAY example

```
SDM-9620-BOTTOM>ARP
ARP CACHE
Operation (DISPLAY/CLEAR,def:DISPLAY) ?

The ARP cache has 4 entry(ies) used on a maximum of 300 entries.

PORT        DESTINATION        MAC ADDRESS    STATE        TTL

LAN1-R1     002.222.000.002    00200CE014F1   RESOLVED     3 h
LAN1-R1     002.222.000.006    00200CE014F2   RESOLVED     3 h
LAN1-R2     002.222.100.002    00200CE014F1   RESOLVED     3 h
LAN1-R2     002.222.100.006    00200CE014F2   RESOLVED     3 h
```

NOTE: The notation for a redundant IP address on the SDM-9620 is, e.g. **LAN1-R1** for *Redundancy IP address 1* on port **ETH1**.

To execute the Display ARP Cache command, enter **ARP** on the command line, and then select **DISPLAY**. The current ARP cache of the NetPerformer is dumped on the screen, as in the following example:

The ARP cache may contain up to 300 entries. The state of an ARP entry may be:

- **RESOLVED**: Both the IP address and the MAC address are known, and the TTL (Time To Live) of the entry is greater than 0. The TTL is decremented every 30 seconds. When the TTL of a RESOLVED entry reaches 0, the entry is destroyed if it is no longer used. If it is still being used, the entry is removed from the list, the TTL reset to 630 seconds and the new entry added to the end of the list.

  No matter how rapidly you execute the ARP command in succession, the ARP cache is updated only once every 30 seconds. Its entries appear from the smallest TTL at the top, to the largest TTL at the bottom of the list.

- **WAIT**: The IP address is known, but the MAC address has not yet been confirmed. The NetPerformer has sent a request through the network for the MAC address, but has not yet received a response. A newly allocated entry is in WAIT state, and is initialized with a TTL of 630 seconds.
- **FREE**: The entry is not used. FREE entries are not displayed with the ARP command.

To clear the NetPerformer ARP table, enter **ARP** on the command line, and then select **CLEAR**. All entries in the RESOLVED state will be cleared from the table. Entries in the WAIT state are cleared automatically after they time out (2 seconds).

ARP/CLEAR example

```
SDM-9230>ARP
ARP CACHE
Operation (DISPLAY/CLEAR,def:DISPLAY) ? CLEAR
```

2.5.7 IP Counters

Console	SNMP
DC/IP	MIB II - *mgmt* - *ip* (category)

The NetPerformer keeps statistics on all IP connections to the unit. The Display Counters command allows you to view these statistics.

To access the IP counters from the console, enter **DC** on the command line, and then select **IP**. For SNMP, open the Standard MIB (MIB II) and bring up the *ip* category under the *mgmt* section.

DC/IP example

```
SDM-9380>DC
DISPLAY COUNTERS
Item (BOOTP/CONFIG/DNS/IP/NAT/PORT/PVC/Q922/Q933/QOS/SLOT/SVC/TIMEP,
def:BOOTP) ? IP
In received.....................................259086
In header errors................................0
In address errors...............................0
In unknown protocols............................0
In discarded....................................37
In delivered....................................32298
Reasm timeout...................................0
Reasm requested.................................0
Reasm ok........................................0
Reasm failed....................................0
Forwarded datagrams.............................226751
Out requested...................................312907
Out discarded...................................13
Out no routes...................................0
Fragmentation ok................................0
Fragmentation failed............................0
Fragments created...............................0
```

```
Out DF discarded................................0
RIP frames discarded............................0
```

The IP counters are detailed in the section "DC/IP" on page 11-17.

2.5.8 PING Remote Unit Command

The Ping Remote Unit command provides a means of testing the current status of gateway connections to the NetPerformer. When you execute the Ping Remote Unit command, a ping is performed on the gateway to determine if it is still alive, that is, recognized by the LAN. Any device in the IP RIP routing table can also be pinged. This command is available from the console only.

Console	SNMP
PING	(not available)

To execute the Ping Remote Unit command:

- Enter **PING** at the console command prompt.
- At the *Display old PING counters* prompt, enter **YES** to view the latest ping results, or **NO** to pass to the next question.
- Select the PING type. Enter **CR** (Cell Relay) to execute a ping along PVCR routes. Enter **IP** to send a ping via other routers.
- Cell Relay Ping: You will be requested to enter the Ping Destination. Enter the unit name of the destination device. Check the Display Destinations Table for a list of destination units (refer to the *WAN/Leased Lines* fascicle of this document series).
- IP Ping: You will be requested to enter the IP address of the destination device. Check the Destination Routing Table, described on page "IP Routing Tables" on page 2-29, for a list of destination addresses.
- If you have chosen the IP Ping type, the next prompt requests the ping length. Enter the number of bytes you want in the ping.
- Enter the desired test duration at the *PING test duration (s)* prompt. This is the duration of the entire test, including retries.
- Enter a timeout value at the *PING timeout (ms)* prompt. If no response is detected during this time, another ping is sent to the gateway. This cycle will continue until either a response is received or the configured test duration runs out.
- Enter the ping delay at the *Delay between PINGs (ms)* prompt. This determines the minimum delay between each ping performed on a gateway.
- The *Background* prompt refers to how the test should be carried out and the results displayed. Enter **NO** to have the test results appear in the foreground. A new row of data will be displayed every 5 seconds for the duration of the test. Enter **YES** to have the test performed in the background. You can continue immediately with another console command.

- If you choose background mode for the test, the *Number of test repetitions* prompt appears. Enter the number of times you would like the complete test to be repeated.

2.5.9 PING in Foreground Mode

Here is an example of how the Ping Remote Unit command is executed (IP type) and what results are displayed in foreground mode:

Ping example: in foreground mode

```
SDM-9380>PING
PING REMOTE UNIT
Display old PING counters (NO/YES,def:NO) ? NO
PING Type (IP/CR,def:IP) ? IP
PING IP address (def:000.000.000.000) ? 205.201.43.237
PING length (0-4096,def:64) ? 256
PING test duration (s) (0-1000000,def:5) ? 10
PING timeout (ms) (0-1000000,def:1000) ? 1000
Delay between PINGs (ms) (0-1000000,def:100) ? 150
Background (NO/YES,def:NO) ? NO
PING to: 205.201.043.237
  Time    Transmit  Receive  Timeout    Error    MinResp    MaxResp   MeanResp
    5.007        5        5        0        0      0.006      0.010      0.007
   10.031       10       10        0        0      0.006      0.010      0.006
   15.050       15       15        0        0      0.006      0.011      0.007
   20.067       20       20        0        0      0.005      0.010      0.007
```

Each row of the results display has been added after a five-second interval. The statistics in this display are detailed in the section “PING” on page 11-20.

2.5.10 PING in Background Mode

If the ping has been performed in background mode, you can view the text results as follows:

- Execute the Ping Remote Unit command again.
- Enter **YES** at the *Display old PING counters* prompt. Only the final test results are displayed.
- Press **[Esc]** to return to the main console prompt.

Ping example: in background mode

```
SDM-9380>PING
PING REMOTE UNIT
Display old PING counters (NO/YES,def:NO) ? YES
PING to: 205.201.043.237
  Time    Transmit  Receive  Timeout    Error    MinResp    MaxResp   MeanResp
   20.067       20       20        0        0      0.005      0.010      0.007
PING Type (IP/CR,def:IP) ? [Esc]
SDM-9380>
```

2.5.11 PING with an Argument

The Ping Remote Unit command can also be executed using an IP address or a domain name as an argument. No other parameters are required.

PING example: using IP address

```
SDM-9230>PING 168.98.10.1
PING REMOTE UNIT
Ping in progress...

PING to: 168.098.010.001
  Time    Transmit  Receive  Timeout    Error    MinResp    MaxResp   MeanResp
    5.005        5        0        4        0      0.000      0.000      0.000
    6.030        5        0        5        0      0.000      0.000      0.000
```

PING example: using domain name

```
UNIT2>PING MEMOTEC.COM
PING REMOTE UNIT
Ping in progress...

PING to: 198.068.000.001
  Time    Transmit  Receive  Timeout    Error    MinResp    MaxResp   MeanResp
    5.005       35       35        0        0      0.040      0.071      0.042
    5.029       35       35        0        0      0.040      0.071      0.042
```

NOTE: The NetPerformer must be configured to resolve the domain name as an IP address, using DNS address resolution. Refer to the chapter “DNS Address Resolution” on page 5-1 for details.

3

Bridge/Router Functions

3.1 NetPerformer Support of Bridge/Router Functions

The NetPerformer bridge/router function provides a physical connection between two LAN segments, and transfers information in the form of packets or datagrams from one LAN to the other. The connection is made using the LAN interface on the local and remote NetPerformer units.

- LAN types supported: Ethernet 802.3 and Ethernet V2. Token-Ring also supported on some legacy NetPerformer products.
- Routing protocols supported:
  - Routing Information Protocol (RIP) for IP
  - RIP for Internetwork Packet Exchange (IPX). Refer to the *Digital Data* fascicle of this document series.
  - BOOTP (bootstrap protocol) relay agent
  - Open Shortest Path First (OSPF) protocol. Refer to the chapter “OSPF Network Support” on page 7-1.

Packets arriving from the LAN connected to the local NetPerformer interface are selectively forwarded over analog or digital dedicated or switched WAN services to the LAN connected to the remote NetPerformer interface, and vice versa. Only completely valid frames or frame cells are reproduced.

NetPerformer bridge/router operations are entirely transparent to the network: any device connected to a LAN communicates across the NetPerformer using exactly the same hardware signals it uses to communicate on its own LAN segment.

- **As a bridge:** The NetPerformer performs data link layer relays between LANs. It participates as a device on the networks to which it is attached, exchanges information with other devices, and selectively forwards that information between the networks. The bridging method used is transparent routing.
- **As a router:** The NetPerformer performs network layer relays between networks. It can connect networks of different types, including point-to-point, multiaccess non-broadcast and multiaccess broadcast with Ethernet LANs.

  To keep track of the network layer address of each of its network connections, the NetPerformer maintains dynamic routing tables containing information about every reachable destination in the Internetwork.

The NetPerformer also supports IP Multicast routing. An IP Multicast application typically has a server at the central site, and several client stations at multiple remote sites.

- The NetPerformer can be configured to use the PIM-DM protocol (Protocol Independent Multicast - Dense Mode), which is a routing algorithm designed for multicast groups that are densely distributed across the network.
- Internet Group Management Protocol (IGMP) operates between a router and the stations on the LAN to provide group membership information.

For details, refer to “IP Connections” on page 2-1.

The NetPerformer can be configured to route all broadcast IP frames to the end station using the *Route Broadcast to End Station* parameter. This parameter provides support of IP Subnet Broadcasting and IP All Subnet Broadcasting.

3.1.1 Transparent Bridge

The NetPerformer acts as a transparent bridge for Ethernet 802.3 or Ethernet V2 LANs. Under transparent bridging, the NetPerformer allows only a single route in the Internetwork to forward frames at any one time. This single active route is maintained by the NetPerformers using the Spanning Tree algorithm, which ensures that frames do not loop. The end stations are completely unaware that their data is passing over a bridge, since the NetPerformer takes the entire responsibility for routing frames.

The NetPerformer bridge uses a series of indexed bridge ports representing physical port numbers on the NetPerformer. Usually, the first bridge port corresponds to the local physical LAN port, and the others correspond to serial WAN/user ports. The actual destination of each bridge port can be identified by examining the bridge port *Destination* statistic, described in “Display Bridge (DB) Command” on page 3-14.

The NetPerformer bridge uses source and destination MAC (Media Access Control) addresses to relay frames.

- When the NetPerformer receives a frame from any remote location or from the local LAN port, it examines the source address of the frame.
- It updates the filtering table with this address, plus the logical source port from which the frame was received.
- Then the NetPerformer examines the destination address, and searches this address in the filtering table.
- If the address is not found, the frame is forwarded to all other logical ports.
- If the address is found in the filtering table and the logical port associated with this address is not the port from which the frame was received, the NetPerformer sends the frame to the physical port associated with that port.

3.1.2 Spanning Tree Protocol

A network with no parallel active paths (that is, loop-free) is called a spanning tree. The NetPerformer forms a single spanning tree with all other transparent bridges for complete network interoperability.

The transparent bridge function of the NetPerformer depends on the IEEE 802.1D Spanning Tree Protocol (STP) to ensure that the correct bridge topology is established and maintained. Through the learning capabilities of the STP, the NetPerformers communicate with one another in order to avoid network loops, and reconfigure pathways when required.

The NetPerformer also uses the spanning tree algorithm to decide how to forward frames and how to propagate broadcast packets so that only one copy of a broadcast frame is delivered to each LAN. Without this algorithm catastrophic results would occur on an Ethernet network, since broadcast packets would be forwarded in both directions at the same time, and the bridge could forward traffic indefinitely. STP ensures that only one active route is used at one time during transparent bridging.

NOTE: To enable the Spanning Tree Protocol, set the bridge STP Enable parameter to **YES**. Refer to "Configuring the Bridge Parameters" on page 3-12.

Spanning Tree Topologies

In an active topology, frames are forwarded through the NetPerformer (or other bridge) ports that are in forwarding state. Other bridge ports that do not forward frames are held in blocking state. These may be put into forwarding state if the topology of the network changes as a result of adjustments made by the STP.

3.1.3 Path Cost

In an interconnected network some routes may be preferred over others. Whenever possible, a fast connection should be used over a slow one, and a lightly loaded LAN should be chosen over a heavily loaded LAN. This leads to the idea of a cost associated with each NetPerformer destination. The cost indicates to the NetPerformer the relative length of the path to a particular LAN or remote unit, and is taken into consideration by the spanning tree algorithm. The higher the cost, the less that route will be preferred.

3.2 IP Routing with RIP Version 2

The NetPerformer product family supports version 2 of the RIP algorithm. RIP Version 2 provides additional possibilities over RIP Version 1 for applications involving a subnetted network. In a subnetted network, several subnetworks are created from one network address by “stealing” some of the host bits and using them to define other networks.

RIP Version 1 does not support autonomous systems and IGP/EGP interactions, subnetting or authentication, since these concepts were not implemented at the time it was developed. In particular, the subnet mask is not transported in Version 1. It is thus very difficult for a router to route a frame properly in a subnetted network when Version 1 is implemented. When the router receives a routing table update with a bit different from 0 in the host part of the address, it cannot decide if the entry is a host address or a subnet address. To counteract this problem the NetPerformer uses the mask of the port on which the frame was received to determine the subnet mask for the address. With RIP Version 2 this is not required, since a subnet mask is transmitted for each address contained in the RIP frame. The RIP V2 algorithm accomplishes the following:

- If the network is subnetted, the subnet addresses are sent on the port that belongs to the network.
- If the port does not belong to the network corresponding to the subnet entry, only the network address is sent over the port.
- If the port is a WAN link and no address is configured on the port, all subnet addresses are sent over the port.

Table 1 shows how RIP frames are treated by the NetPerformer depending on the RIP configuration of its IP interface and the type of RIP frame that is received. Set the RIP configuration using the IP RIP parameter, which is available for WAN (PVCR) links, LAN interfaces, PPP connections and PVCs in PVCR or RFC1490 mode.

The *IP RIP* parameter as well as IP RIP directionality, authentication and password are configured in various areas of the configuration:

- **WAN Link**: Described in the *WAN/Leased Lines* fascicle of this document series
- **PPP Link**: Described in the *WAN/Point-to-Point (PPP)* fascicle
- **PVCs (PVCR or RFC1490)**: Described in the *WAN/Frame Relay* fascicle
- **Backplane Links**: See the *Global Functions* chapter of the *Quick Configuration* fascicle
- **LAN Interface**: See “Configuring the Ethernet LAN Port” on page 1-2.

Configuration of IP RIP Interface (IP RIP parameter)	RIP Version 1 Frame Received	RIP Version 2 Frame Received: no Authentication	RIP Version 2 Frame Received: with Authentication
DISABLE	Frame discarded	Frame discarded	Frame discarded

Table 1Processing of RIP Frames Received

Configuration of IP RIP Interface (IP RIP parameter)	RIP Version 1 Frame Received	RIP Version 2 Frame Received: no Authentication	RIP Version 2 Frame Received: with Authentication
V1	Frame processed as for RIP Version 1	This RIP frame was sent by broadcast. Frame processed as for RIP Version 1.	Frame discarded, since authentication cannot be set on the interface.
V2 BROADCAST	Frame processed as for RIP Version 1	Frame processed according to RIP Version 2 standard.	Frame processed according to RIP Version 2 standard.
V2 MULTICAST	Frame processed as for RIP Version 1	Frame processed according to RIP Version 2 standard.	Frame processed according to RIP Version 2 standard.

Table 1Processing of RIP Frames Received

NOTE: If you are using RIP Version 2 and different subnet masks have different lengths, the **MULTIHOMEDTYPE** extended parameter must be set to **IGNORENET**. This would be required, for example, when routing an IP frame with a destination address that uses a subnet with more bits than the local IP address mask.

NetPerformer V8.0.0 introduced support for integrated RIP and OSPF (Open Shortest Path First) routing protocols. For more information, see the chapter “OSPF Network Support” on page 7-1.

3.2.1 IP Routing with the BOOTP Protocol

The NetPerformer includes a BOOTP (bootstrap protocol) relay agent function that permits the forwarding of BOOTP/DHCP frames across the network. BOOTP is used by a computer that requires external information in order to determine its IP address, subnet mask, gateway address and other IP characteristics. Examples of computers that use BOOTP are a diskless machine and a laptop computer that migrates from one LAN to another within the same corporate network.

BOOTP is a layer of the TCP/IP architecture that runs on top of the UDP layer of IP. To function properly, IP routing must be enabled on the NetPerformer. Use the *Router* parameter of the Setup IP Global menu (see *“Configuring the Global IP Parameters” on page 2-21*).

The BOOTP/DHCP frame carries the following IP address information:

- **yiaddr**: (your IP address) The IP address of the client in the client/server relationship.
- **siaddr**: (server IP address) The IP address of the next server to use in the bootstrap.

- **giaddr**: (gateway IP address) The IP address of the relay agent, used in booting via a relay agent.
- **ciaddr**: (client IP address) This IP address is filled in only if the client can respond to ARP requests.
- **chaddr**: (client hardware address) The hardware address of the client in the client/server relationship.

NOTE: The BOOTP relay agent can forward both BOOTP and DHCP frames. From the point of view of the relay agent, BOOTP and DHCP frames are identical in structure.

A NetPerformer should be configured with the BOOTP relay agent capability when it will be used to sent BOOTREQUEST frames to another relay agent. Intervening routers do not require the relay agent capability, since the BOOTP/DHCP frames can be forwarded to the destination IP address in the same way as regular IP frames. The hop count is increased only at relay agents.

Further information regarding BOOTP configuration is provided in "Configuring the BOOTP Parameters" on page 2-23.

3.3 Virtual Connections

The concept of Virtual Connections was inspired by the ATM method of cell transfer, connection routing and reserved channel allocation and was adapted by Memotec for the NetPerformer in a medium-sized network. In tandem with PowerCell technology, Virtual Connections provide increased performance, guaranteed voice/fax prioritization and dynamic restructuring of connections according to optimal routing paths.

Through Virtual Connections the connection is routed, not the data frame. The virtual path uses virtual channels that are dynamically allocated between each pair of units participating in the Virtual Connection. Multiple virtual channels are available from each port in the network. Once a virtual path is opened, it remains available to the cell flow as long as the network is unchanged. The data cells flow through without further analysis of source and destination paths.

On power-up, all NetPerformers in the network exchange information and construct routing tables of all accessible destination units. These routing tables allow each unit to properly route the cells along the Virtual Connections. If any communication line fails, all tables are dynamically updated to reflect new routing paths. In this way, the network structure reacts instantaneously and reliably to adverse network conditions.

The network structure reacts instantaneously and reliably to adverse network conditions. If any communication line fails the network is automatically restructured, with no cabling changes or reconfiguration required. Since data transmission is cell-based, the frame does not have to be completely processed at any location before changing to a new virtual path. Alternate paths are established so quickly that no sessions are terminated, and there is no danger of frames being reordered due to latency problems.

3.3.1 Implications for Voice/Fax Transmissions

An advantage of transporting voice over virtual connections is that all of the bandwidth is made available to the data traffic when a telephone line goes from an off-hook to on-hook condition. The virtual channel is opened as soon as a call is active, and closed as soon as an on-hook condition is detected. Bandwidth usage is more dynamic and efficient than with any other transport method.

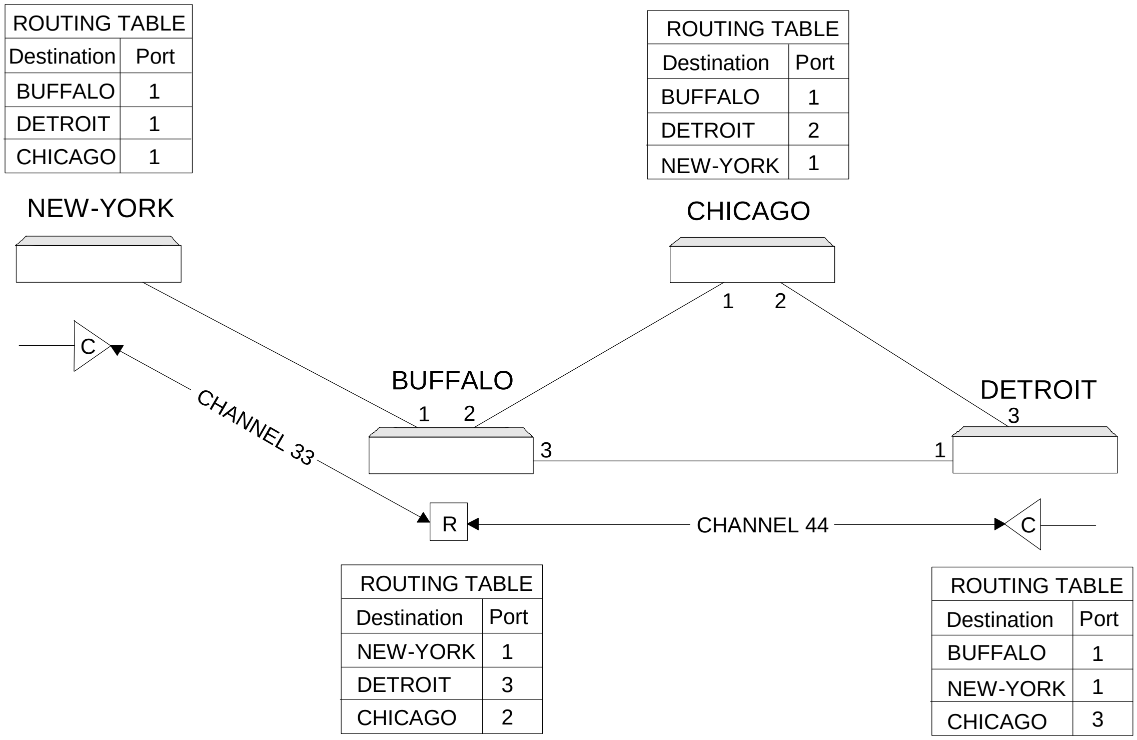
The technology routes the connection, not the data frame, along a virtual path of dynamically allocated virtual channels between two NetPerformers. Once a virtual path is opened, it remains available to the voice/fax cell flow as long as the network is unchanged.

3.4 Dynamic Routing Tables

Each NetPerformer is configured with a user-defined unit name. When the link between two NetPerformers comes up, each unit exchanges its name with all other units it can reach, and constructs a routing table containing the names of all destination units and the port numbers used to access them.

These routing tables allow each unit in the network to properly route the cell connections via Programmable Variable Cell Relay (PVCR) links, which use a cell-by-cell relay method to minimize network latency. Cell relay can begin as soon as two units are connected via the routing table.

For example, the unit named Buffalo in Figure 3-1 recognizes the unit Detroit when the link between them is activated, since the unit names are exchanged during the link establishment process. Buffalo deduces that it can reach Detroit via port 3, and Detroit deduces that it can reach Buffalo via port 1. Each unit enters this information in its routing table, and then broadcasts the destination and metric to all other units in the network.



ROUTING TABLE	
Destination	Port
BUFFALO	1
DETROIT	1
CHICAGO	1

ROUTING TABLE	
Destination	Port
BUFFALO	1
DETROIT	2
NEW-YORK	1

ROUTING TABLE	
Destination	Port
NEW-YORK	1
DETROIT	3
CHICAGO	2

ROUTING TABLE	
Destination	Port
BUFFALO	1
NEW-YORK	1
CHICAGO	3

Figure 3-1: Routing the Virtual Connections

If the network is stable, the routing tables remain unchanged. If it changes, however, the routing tables are refreshed dynamically. That is, if any line between two units fails the source unit sends a broadcast message announcing the failure, which forces all connections using the failed link to another path.

For example, if the direct line between Buffalo and Detroit fails, Buffalo will learn from Chicago that it can reach Detroit via Chicago. Detroit will learn the same thing from Chicago. Buffalo immediately updates its routing table entry for Detroit to port 2. At the same time, Detroit updates its routing table entry for Buffalo to port 3, and cells are now relayed between Buffalo and Detroit via the unit Chicago. This dynamic change is completed within a few seconds, with no harmful effect on data transmission and no active session loss.

When the failed line comes back up, the routing tables are changed once again, taking into account the reduced cost of the direct connection between Buffalo and Detroit.

3.4.1 Types of Routing Tables

The NetPerformer offers three types of destination routing tables:

- **PVCR** (Programmable Variable Cell Relay) routing table. This table provides the current status of all destinations reached by the NetPerformer via ports and PVCs configured with the PVCR protocol. The PVCR routing table is displayed using the Display Destinations (**DD**) command. It is discussed further in the *WAN/ Leased Lines* fascicle of this document series.
- **IP** routing table. This table provides the current status of connections via the NetPerformer and other routers using RIP (Routing Information Protocol) IP or multihomed IP addressing. For details on the IP routing tables, consult “IP Connection Status” on page 2-29.
- **IPX** (Internetwork Packet Exchange) routing table. This table provides the current status of IPX RIP and IPX SAP routing connections in a Novell network. For details on the IPX routing tables, refer to the discussion of IPX connections in the *Digital Data* fascicle of this document series.

To access the IP and IPX tables from the NetPerformer console, use the Destination Routing Table (DR) command. For SNMP access, use the *ipRouteTable* entry under the *mgmt* section of the standard MIB (MIB II) to view the IP tables, and the *private/ novell/ipx* group of the private Novell MIB to view the IPX routing tables.

3.4.2 Virtual Paths

Once the routing tables are fully constructed and one unit can reach another, the first message the source unit sends is a Connect Request containing the destination unit name and the desired priority class. For example, to communicate with Detroit the unit New York sends a Connect Request message carrying the destination unit name Detroit. As this message travels to its destination, a channel is reserved between each unit along its path. In the example in Figure 3-1, channel 33 is reserved from New York to Buffalo, then channel 44 is reserved from Buffalo to Detroit. Each unit chooses the first channel available in the appropriate priority class.

The Connect Request also identifies the source and destination ports, PUs or PVCs, and the type of connection required: transparent port, SNA PU, Frame Relay PVC, or router. When the destination unit receives the Connect Request message, it responds with a Connect Confirm message. In the example in Figure 3-1, Detroit sends a Connect Confirm to the source unit, New York. Channels are now reserved in the other direction as the Connect Confirm message travels back to New York. These channels may not be the same ones used to send the Connect Request. The best route between each pair of units is always chosen, based on the lowest number of units that must be accessed to reach the destination device.

Once the connect messages have been successfully exchanged and the appropriate channels reserved, the virtual path is established and ready for cell relay. The reserved channels act as a pipeline for all cells requiring this Virtual Connection between units. It is as though port 1 on source unit New York is now connected to channel 33, and port 1 on destination unit Detroit is connected to channel 44. All cells destined for Detroit port 1 will follow the virtual path via Buffalo. If a line failure occurs, the cells are immediately rerouted via Chicago according to the dynamic change in routing tables.

3.4.3 Overhead and Latency

When combined with Cell Relay technology, Virtual Connections result in low overhead and low latency, which means shorter transmission delays when compared to routers using the traditional encapsulation routing method.

The simple cell-by-cell relay of the NetPerformer avoids the high overhead associated with encapsulation techniques. Each cell carries a small header of 3 to 5 bytes, which is sufficient to handle any traffic type, be it native SNA, DDCMP, VIP, or TCP/IP. Much larger headers are required for techniques that route data on a frame-by-frame basis. This excess overhead must be absorbed and retransmitted by each unit along the route, resulting in slower response time than with the NetPerformer.

Encapsulation routing methods rely on a store and forward, frame-by-frame method of data transfer. Each router must receive the entire frame before sending it on to the next unit. Routing entire frames is much more complicated than simply sending a cell along an open virtual path. And the complexity increases with the size of the frame. As a result, encapsulation routing is plagued with high latency and transmission delays. The NetPerformer, on the other hand, can avoid store and forward, which reduces latency to very low levels.

NOTE: This applies to data routed between PUs or transparent ports only. TCP/IP data requires frame routing.

The high latency factor of encapsulation routing can result in frames being sent out of order when an alternate route is abandoned for a more direct route. That is, earlier frames can be received via the alternate route after frames are received along the direct route. With Virtual Connections and dynamic routing tables this danger is avoided, since network restructuring is accomplished in less time than it takes to segment a frame and relay its cells to the destination unit.

3.5 Configuring the Bridge Parameters

The Setup Bridge menu lets you configure all parameters for the NetPerformer bridging and routing functions. For the console, enter **SE** followed by **BRIDGE**. For SNMP, the *bridge* category includes all variables affecting bridge configuration. For text-based configuration the [bridge] heading is used.

Console	SNMP	Text-based Config
SE, BRIDGE	*bridge* (category)	[bridge] (heading)

SE/BRIDGE example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? BRIDGE
BRIDGE> Enabled (def:NO) ? YES
BRIDGE> Spanning Tree Protocol active (def:NO) ?
BRIDGE> Aging time (s) (10-1000000,def:300) ?
BRIDGE> Hello messages interval (s) (1-10,def:2) ?
BRIDGE> Hello frames maximum age (s) (6-40,def:20) ?
BRIDGE> Forward delay (s) (4-30,def:15) ?
BRIDGE> Bridge priority (0-65535,def:32767) ?
```

These parameters are detailed in the appendix “SE/BRIDGE Configuration Parameters” on page 10-1.

3.6 Viewing the Bridge Status

Two commands are available from the console to view the status of bridge connections on the NetPerformer. The Display Bridge Addresses (**DBA**) command lists all active bridge addresses, and the Display Bridge (**DB**) command displays the current status of the bridge and all ports participating in the bridge topology. The bridge statistics are also available from SNMP, using the *statBridgeBridge* group of variables and the *statBridgePortEntry* table.

3.6.1 Display Bridge Addresses (DBA) Command

Console	SNMP
DBA	(not available)

The Display Bridge Addresses (**DBA**) command can be used to display currently active bridge addresses. With this command you can view the MAC addresses that have been learned by the bridge in transparent mode.

The Display Bridge Addresses command is available from the console only. Enter **DBA** at the console command line.

DBA example

```
SDM-9230>DBA
DISPLAY BRIDGE ADDRESSES

ADDRESS           PORT     TIME(sec)

002083000435      LAN1    1180
0020830008B0      WAN1    276
002083000A44      LAN2    1570
```

The DBA command displays the following:

ADDRESS

Console	SNMP
ADDRESS	*not available*

The MAC address that has been learned by the bridge in transparent mode for this bridge connection.

PORT

Console	SNMP
PORT	*not available*

The port on which the MAC address was learned. On the console display, the values in this column correspond to physical connections on the NetPerformer:

- **WAN*x*:** PVCR port. The value *x* is equivalent to the serial port number
- **LAN*x*:** LAN port (Ethernet). The value *x* is equivalent to the LAN port number
- **PVC*x*:** Frame Relay or ATM PVC. The value *x* is equivalent to the PVC number
- **FW*xy*:** Firewire (backplane) connection on a SDM-9500 rackmount model. The value *x* is the Rack ID and *y* is the Slot ID.

TIME(sec)

Console	SNMP
TIME(sec)	*not available*

The duration of time, in seconds, that has elapsed since the last update for this address.

3.6.2 Display Bridge (DB) Command

Console	SNMP
DB	*statBridge* (category)

The Display Bridge command lets you view the current status of the bridge and all ports participating in the bridge topology. To access this function from the console, enter **DB** on the command line.

DB example

```
SDM-9230>DB
DISPLAY BRIDGE STATISTICS
BRIDGE     > Address discard....................0
BRIDGE     > Transparent frame discard..........0
BRIDGE     > Designated root....................7FFF00200AB08DF1
BRIDGE     > Root cost..........................0
BRIDGE     > Root port..........................NONE
BRIDGE     > Frame filtered.....................0
BRIDGE     > Frame timeout discard..............0

BRG LAN1-1> Destination.........................LOCAL LAN1
BRG LAN1-1> port index..........................0001
BRG LAN1-1> State...............................FORWARD
BRG LAN1-1> Designated root.....................7FFF00200AB08DF1
BRG LAN1-1> Designated cost.....................0
BRG LAN1-1> Designated bridge...................7FFF00200AB08DF1
BRG LAN1-1> Designated port.....................8001
BRG LAN1-1> Transparent frame in................0
BRG LAN1-1> Transparent frame out...............0

BRG LAN2-1> Destination.........................LOCAL LAN2
BRG LAN2-1> port index..........................0002
BRG LAN2-1> State...............................FORWARD
BRG LAN2-1> Designated root.....................7FFF00200AB08DF1
BRG LAN2-1> Designated cost.....................0
BRG LAN2-1> Designated bridge...................7FFF00200AB08DF1
BRG LAN2-1> Designated port.....................8002
BRG LAN2-1> Transparent frame in................0
BRG LAN2-1> Transparent frame out...............0
```

All of these status messages are detailed in “Bridge Port Statistics (DB)” on page 11-7.

4

DHCP Client

4.1 About the DHCP

The NetPerformer supports the Dynamic Host Configuration Protocol (DHCP) Client function, which can retrieve the IP address of the Ethernet LAN interface automatically from the local DHCP server.

This alternative to manual configuration is particularly useful when the Ethernet LAN port provides WAN connectivity. Through DHCP Client, the link can come up as soon as a LAN connection is detected, allowing immediate remote access to the unit and minimizing the amount of manual intervention required.

4.1.1 DHCP Application Scenario

A typical application of DHCP is the following:

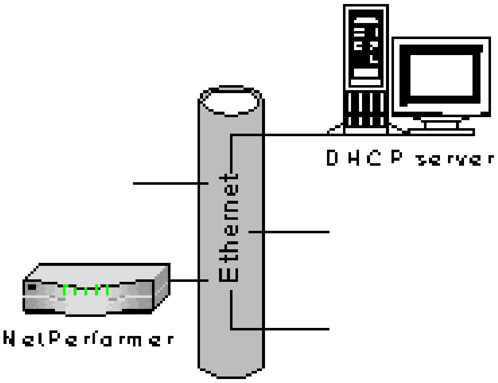


Figure 4-1: DHCP Application

In this scenario, the NetPerformer is connected directly to the Ethernet segment on which the DHCP server is located.

NOTE: This implementation assumes that the DHCP server is located directly on the LAN or is accessible through a BOOTP relay agent other than the requesting NetPerformer.

4.1.2 Retrieving the DHCP Interface Information

To retrieve DHCP interface information, the NetPerformer must be connected to a LAN on which one or more DHCP servers are located, or to a BOOTP relay agent that is able to access a DHCP server.

The NetPerformer requests the following information from the DHCP server:

- An IP address

  Only the first IP address configured on the Ethernet port can be received from the DHCP server (*IP address 1*). If a second address is desired, it must be configured manually (see next section). This would be the case when using the dual IP addressing capability of the NetPerformer.

- Local subnet mask
- Default gateway, if the parameter *Accept the default gateway from DHCP server* is set to **YES** (the default value)
- Interface MTU.

4.2 Configuring the LAN Interface as a DHCP Client

The NetPerformer LAN interface is an Ethernet port built into the base unit. To configure the Ethernet port for DHCP client functionality using the NetPerformer console:

- Enter the menu sequence: **SE ↵ PORT ↵ ETH**
- Press **<Enter>** three times to reach the *DHCP* parameter
- Set *DHCP* to **ENABLE** if you want the first Ethernet IP address on the NetPerformer unit to be allocated by a DHCP server.

  If *DHCP* is set to **ENABLE**, *IP address 1* and *Subnet mask 1* are not available, since they are automatically allocated by the DHCP server.
- Set *Accept the default gateway from DHCP server* to **YES** to retrieve the default gateway address.
- *IP address 2*, along with *Subnet mask 2*, can be used in a dual IP address application, but must be configured manually:
  - The two IP addresses must be set on two separate IP networks or sub-networks
  - Leave *IP address 2* at its default value (**000.000.000.000**) when the NetPerformer requires only a single IP address.

SE/PORT/ETH example: with DHCP enabled

```
UNIT2>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (ETH/CSL/1/2/3,def:ETH) ? ETH
PORT ETH> Protocol (def:ETH AUTO) ?
PORT ETH> Link integrity (def:YES) ?
PORT ETH> MAC address (def:000000000000) ?
PORT ETH> DHCP (def:DISABLE) ? ENABLE
PORT ETH> Accept the default gateway from DHCP server (def:YES) ?
PORT ETH> IP address 2 (def:000.000.000.000) ?
PORT ETH> Subnet mask 2 (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT ETH> Frame size (128-8192,def:1500) ?...
```

For complete descriptions of these parameters, refer to the appendix “SE/PORT/ETH Configuration Parameters” on page 8-1.

4.3 DHCP Status

The **PORT** option of the Display States (**DS**) command includes fields which show the current settings of the Ethernet port when it is DHCP-enabled.

To display the current status of the Ethernet port, including its DHCP information:

- Enter **DS** at the NetPerformer console command prompt
- Enter **PORT** at the *Item* prompt.

The Ethernet port is the first port that is listed.

DS/PORT example: with DHCP BOUND state

```
UNIT2>DS
DISPLAY STATES
Item (GLOBAL/PORT/PU/PVC/SLOT/SVC/VLAN,def:GLOBAL) ? PORT
PORT ETH> Protocol..............................ETHERNET
PORT ETH> Interface.............................10BASET
PORT ETH> Speed.................................10M
PORT ETH> Duplex mode...........................HALF
PORT ETH> Operating mode........................L-
PORT ETH> State.................................OPEN
PORT ETH> Network address.......................00200AB05A9A
PORT ETH> Burned-in address.....................00200AB05A9A
PORT ETH> Number of deferred transmissions......2
PORT ETH> Number of collision frames............0
PORT ETH> DHCP state............................BOUND
PORT ETH> IP address............................172.016.036.105
PORT ETH> Subnet mask...........................255.255.240.000
PORT ETH> MTU...................................0
PORT ETH> IP address lease......................1 d
```

The fields that provide DHCP information are the following:

4.3.1 DHCP state

Console	SNMP
DHCP state	statiflanDhcpState

Shows the current state of the DHCP request, which may be:

- **OFF**: DHCP is not configured on the port.
- **INIT**: First step when DHCP is activated. While in this state, the client sends requests in a server discovery frame to discover all available servers.
- **SELECTING**: The client is waiting for replies from all available servers. The client goes into this state after sending the server discovery frame.
- **REQUESTING**: After receiving offers from the available servers, the client selects one and sends a request to reserve this offer.
- **BOUND**: The client and the server have agreed on one offer. This is the final state of the DHCP configuration.

- **REBINDING**: The rebinding timer for the lease of the parameters has expired; trying to renew them with the server.
- **RENEWING**: The renewing timer for the lease of the parameters has expired; trying to renew them with the server.
- **INIT_REBOOT**: The DHCP sequence has restarted using parameters that were previously received from a server.
- **REBOOTING**: The client is waiting for the server to reply to a request using previously received parameters.
- **RELEASED**: The parameters provided by the server have been released by the client.

4.3.2 IP address

Console	SNMP
IP address	statiflanIpAddress

Shows the IP address that was assigned to the NetPerformer by the DHCP server.

4.3.3 Subnet mask

Console	SNMP
Subnet mask	statiflanSubnetMask

Shows the local subnet mask that was provided by the DHCP server.

4.3.4 MTU

Console	SNMP
MTU	statiflanMtu

Shows the interface MTU that was provided by the DHCP server.

4.3.5 IP address lease

Console	SNMP
IP address lease	statiflanIpAddressLease

Shows how much time remains in the current IP address lease. The lease time is determined by the server, and ranges from 1 second (**1 s**) to 136 years (**136 y**). The following notation is used:

- **s**: seconds
- **m**: minutes
- **h**: hours
- **d**: days
- **y**: years.

NOTE: The client can ask for a specific lease time, but this request may be overruled by the server.

4.4 Managing the DHCP IP Address

The **DHCP CLIENT** command is available from the NetPerformer console to manage the DHCP IP address. To execute this command:

- Enter **DHCP** at the console command line
- On a NetPerformer product with more than one LAN interface, select the Ethernet port (**ETH1** or **ETH2**)
- Select one of the following operations:
  - **RELEASE**: Clears the IP address that was learned from the DHCP server
  - **RENEW**: Gets a new IP address from the DHCP server

DHCP/ RELEASE example

```
SDM-9230>DHCP
DHCP CLIENT
Ethernet port (ETH1/ETH2,def:ETH1) ?
Operation (RELEASE/RENEW,def:RELEASE) ?
Release IP address, please confirm (NO/YES,def:NO) ? YES
```

DHCP/RENEW example

```
SDM-9230>DHCP
DHCP CLIENT
Ethernet port (ETH1/ETH2,def:ETH1) ?
Operation (RELEASE/RENEW,def:RELEASE) ? RENEW
Renew IP address, please confirm (NO/YES,def:NO) ? YES
```

4.5 DHCP Server

4.5.1 DHCP parameters with the addition of DHCP Server

Console	SNMP	Text-based Config
Port / ETH x / DHCP	iflanDhcpEnable	[iflan x] DhcpEnable

Description: DHCP operation mode.

Values: DISABLE, CLIENT, or SERVER.

Default: SERVER on ETH1 and DISABLE on ETH2.

NOTE: The DHCP SERVER mode is only available on Ethernet 1 port.

Console	SNMP	Text-based Config
Port / ETH 1 / Start of IP address range	iflanDhcpServerIpMin	[iflan 1] DhcpServerIpMin

Description: First IP address provided by the DHCP server from a range of IP addresses. This parameter will not appear at the console if the DHCP server is not enabled.

Values: Any IP addresses.

Default: 192.168.0.10.

Console	SNMP	Text-based Config
Port / ETH 1 / End of IP address range	iflanDhcpServerIpMax	[iflan 1] DhcpServerIpMax

Description: Last IP address provided by the DHCP server from a range of IP addresses. This parameter will not appear at the console if the DHCP server is not enabled.

Values: Any IP addresses.

Default: 192.168.0.254.

Console	SNMP	Text-based Config
Port / ETH 1 / Lease time (hours)	iflanDhcpServerLeaseTime	[iflan 1] DhcpServerLeaseTime

Description: Duration, in hours, of the lease time of an IP address provided by the DHCP server to a DHCP client. If the DHCP server does not receive a renew request within that lease time, it will expire the client's lease and reclaim its IP address for possible lease to another client.

Values: 1 to 1176.

Default: 1.

4.5.2 DHCP Server Display State Command

```
SERVER>DS
DISPLAY STATES
Item (DHCPSRV/GLOBAL/PORT/PVC/REDUNDANCY,def:DHCPSRV) ?
IP address........................................192.168.000.010
IP address status...................................OFFERED
Client's MAC address................................00200AB0CF34
Lease time remaining................................9 s

IP address........................................192.168.000.011
IP address status...................................AVAILABLE

IP address........................................192.168.000.012
IP address status...................................AVAILABLE

IP address........................................192.168.000.013
IP address status...................................AVAILABLE

IP address........................................192.168.000.014
IP address status...................................AVAILABLE
```

5

DNS Address Resolution

5.1 About the Domain Name Server

The NetPerformer uses the Domain Name Server (DNS) feature to resolve the IP address of a server based on its domain name, for example, *memotec.com*. This allows the NetPerformer to interconnect with Clarent Softswitch products using their host names instead of their IP addresses. DNS address resolution simplifies command entry, provides greater flexibility and allows for improved integration with the Clarent Softswitch product family.

As a DNS client, the NetPerformer is able to resolve the supplied domain name transparently, using standard DNS queries (see next section). To ensure proper functionality, the user must supply a list of potential DNS servers which can be used to resolve these queries (see “Defining the DNS Characteristics” on page 5-3).

NOTE: The NetPerformer supports only the A type of standard DNS queries, which carry out host name to IP address translation. In V9.2.0 DNS address resolution can be used only for the internal PING command. On a NetPerformer installed with the SIP VoIP licensed option, it can also be used for the SIP Global *Proxy server address* parameter.

5.1.1 DNS Queries

A DNS client can typically perform two types of queries:

- Standard query: Specifies a target domain name and asks for the resource records (RRs), that match this domain name. The resource record contains information related to the target name, and its content varies with the type of query that has been performed (see Table 5-1).
- Inverse query: The client tries to match a resource to a host name. These queries are used only for debugging and database maintenance, and are not supported by the NetPerformer.

Type	Definition
A	A host address related to a host name
PTR	A pointer to another part of a domain name space
CNAME	The canonical name of an alias host name
MX	Mail exchange information related to a domain
NS	The authoritative name server of a domain
SOA	The start of a zone of authority
HINFO	The CPU and OS used by a host

Table 5-1: Standard DNS Queries

These queries were designed to allow different kind of applications to get some insight into how a domain is managed:

- **A:** Performs host name to IP address translation. This is the only type of query supported by the NetPerformer in V9.2.0.
- **PTR:** Performs host address to host name translation.
- **CNAME:** Designed to get the real host name of an alias. This may be useful when different servers know a host by different names.
- **MX:** Obtains information about the mailing services present in the domain. It permits tracing an email address to its original mailing host, ensuring proper routing of Internet mail.
- **NS and SOA:** Obtain information on the authoritative name server of a domain. These queries allow the client to determine exactly who the authoritative server is, in the domain it is interested in.

  Since most DNS servers use caching, information from them cannot be asserted as being 100% accurate. The only server with reliable information is the authoritative server, which is generally located within the domain for which it is the authority.
- **HINFO:** Used by some clients (for example, FTP) to get information on the kind of environment they are connecting to, in order to adjust their mode of operation accordingly.

5.1.2 Defining the DNS Characteristics

The DNS characteristics must be defined to ensure correct IP address resolution. To configure the DNS characteristics using the NetPerformer console, enter the menu sequence: **SE** ↵ **IP** ↵ **DNS**. The following parameters are presented at the console:

SE/IP/DNS example

```
UNIT2>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? DNS
DNS> Primary server address (def:000.000.000.000) ? 198.68.0.1
DNS> Secondary server address (def:000.000.000.000) ?
DNS> Ignore DNS time to live (def:NO) ?
```

These parameters are detailed in “SE/IP/DNS Submenu” on page 9-29.

5.1.3 DNS Address Resolution Application Scenario

The most common scenario for DNS address resolution is with the NetPerformer SIP VoIP licensed software option (available with NetPerformer version 10.1 and higher), where the NetPerformer must connect to a remote server (gatekeeper or SIP proxy), and the user provides only a host name.

NOTE: The NetPerformer SIP VoIP option provides voice over IP using standard SIP. On the NetPerformer base product, DNS address resolution can be used only with the PING command. Contact your NetPerformer distributor for a product upgrade to SIP functionality.

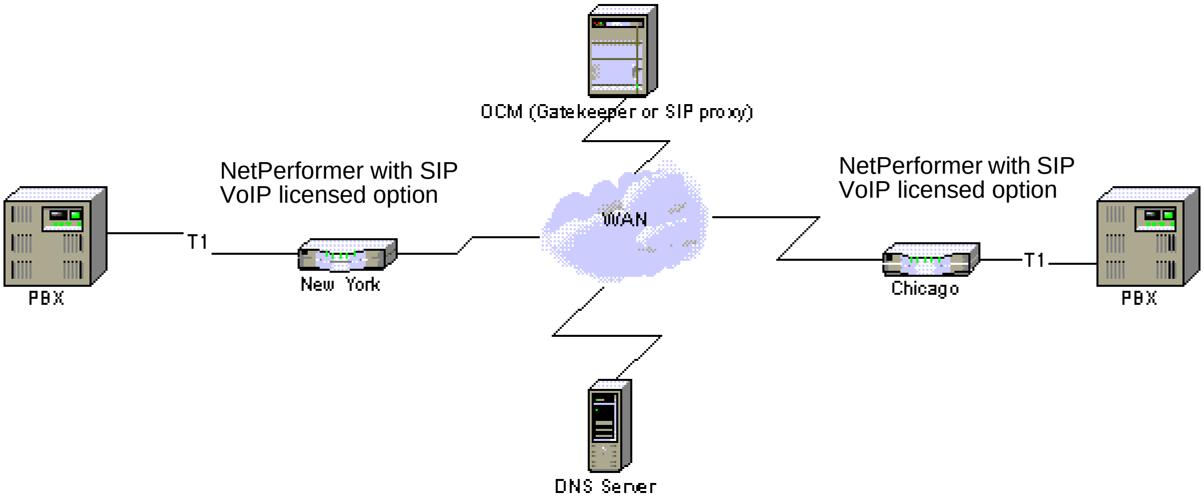

OCM (Gatekeeper or SIP proxy)
NetPerformer with SIP VoIP licensed option
NetPerformer with SIP VoIP licensed option
WAN
T1
T1
PBX
New York
Chicago
PBX
DNS Server

Figure 5-1: DNS Address Resolution Application

In this example, a WAN link is used to transmit voice over IP between one NetPerformer in New York and another in Chicago. The NetPerformer at the New York site must connect with the Clarent Class 5 Call Manager (C5CM) to correctly route the call to Chicago.

NOTE: The user has configured this NetPerformer unit with the address of one or more DNS servers, as well as the name of the Clarent Class 5 Call Manager host, for example:

```
SIP Global> Proxy server address (def:) ? sip-proxy.acme.com
```

The NetPerformer in New York issues a request to the DNS server, which returns the IP address of the Clarent Class 5 Call Manager. This provides the NetPerformer with the information it needs to establish the desired connection.

NOTE: This application is available only on NetPerformer units installed with the SIP VoIP licensed option.

5.1.4 DNS Client Statistics Counters

New statistics counters have been added to the NetPerformer console to track the efficiency of the DNS servers, including the number of requests and answers processed by each. These counters are grouped under the *Item* named **DNS**.

To display the **DNS** counters, enter the menu sequence: **DC** ↵ **DNS** at the NetPerformer console command prompt.

DC/DNS example

```
SDM-9380>DC
DISPLAY COUNTERS
Item (BOOTP/CONFIG/DNS/IP/NAT/PORT/PVC/Q922/Q933/QOS/SLOT/SVC/
TIMEP,
def:BOOTP) ? DNS
DNS SERVER 1> IP address........................000.000.000.000
DNS SERVER 1> Number of requests................0
DNS SERVER 1> Number of answers.................0
DNS SERVER 2> IP address........................000.000.000.000
DNS SERVER 2> Number of requests................0
DNS SERVER 2> Number of answers.................0
```

5.1.5 IP address

Console	SNMP
IP address	statDnsIp

The IP address of the DNS server.

5.1.6 Number of requests

Console	SNMP
Number of requests	statDnsRequest

The number of DNS requests received by the DNS server.

5.1.7 Number of answers

Console	SNMP
Number of answers	statDnsAnswer

The number of DNS replies sent by the DNS server.

5.1.8 Managing the DNS Entries

To manage the DNS entries, use the DNS Cache (**DNS**) command. This command lets you view or delete the DNS entries that have been defined with the **SETUP/IP/DNS** submenu (described on “Defining the DNS Characteristics” on page 5-3).

NOTE: The **DNS** command is available to users with **FULL** console access only.

To execute the **DNS** command:

- Enter **DNS** at the NetPerformer console command prompt.
- At the *Operation* prompt, enter:
  - **DISPLAY**, to view the current DNS characteristics, ***or***
  - **CLEAR**, to delete the DNS characteristics.

DNS Cache example

```
SDM-9380>DNS
DNS CACHE
Operation (DISPLAY/CLEAR,def:DISPLAY) ?

DESTINATION                                 IP ADDRESS        TTL

A.ACME.COM                                  192.168.001.001   30  s
TEST.ACME.COM                               172.016.035.233   1   h
```

6

Network Address Translation (NAT)

6.1 About Network Address Translation

Network Address Translation (NAT) provides a simple solution to IP address depletion in Internet applications by allowing IP addresses to be reused in different domains. This chapter explains how NAT works, describes how to configure NAT on the NetPerformer, and provides examples of NetPerformer-based NAT applications.

6.1.1 NAT Feature Overview

With the growth of the Internet, IP address depletion has become a significant problem for large scale Internet operations. The number of Internet IP addresses is not infinite and it is becoming more and more difficult to get one. For example, it is highly possible to have more users wanting to connect to the Internet than the number of Internet IP addresses available.

Network Address Translation (NAT) was originally proposed as a short-term solution to this problem, and has remained a viable option for ensuring efficient service in Internet applications. NAT allows hosts in a private network (or stub domain) to transparently communicate with destinations on an external network (or another stub domain) and vice versa. See Figure 6-1.

Another reason for providing address translation is that it allows the integration of two dissimilar networks. There is a growing number of very complex private IP networks today. It has become very difficult for a new network to be tied to existing networks without creating IP addressing problems. In many cases, both networks may even use the same pool of IP addresses (the popular private 10/8 class A network number is a prime example). Once again, the solution is to perform address translation.

6.1.2 The Need for IP Address Translation

IP address translation is required when a network’s internal IP addresses cannot be used outside the network because:

- They are invalid for use outside the network, for example, because they are already used, *or*
- The internal addressing system must be kept private from the external network for security or other reasons

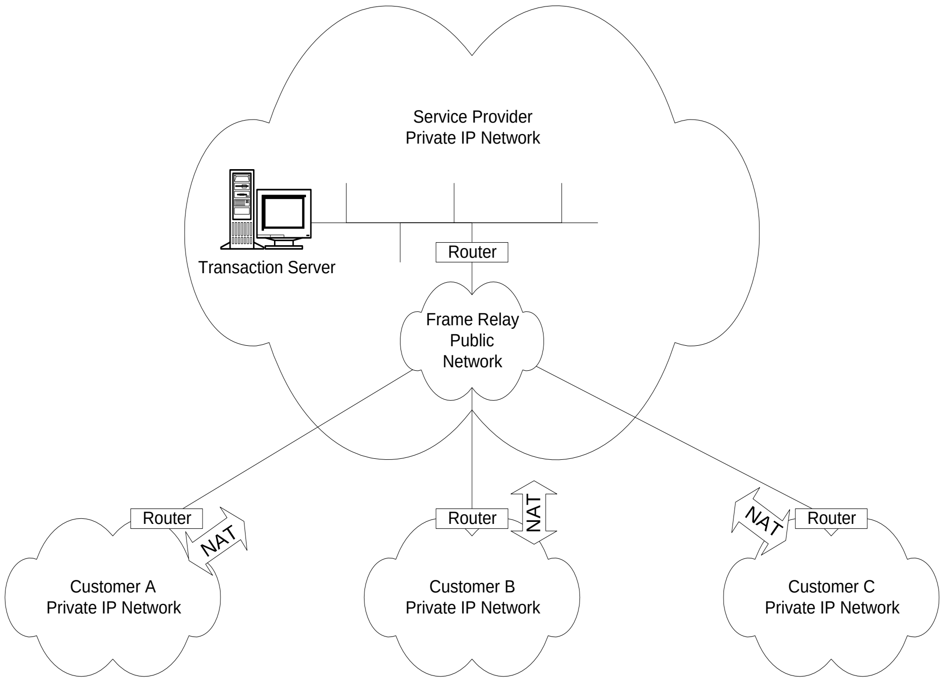


Figure 6-1: Integration of Public and Private Networks

6.1.3 Advantages of NAT

The main advantage of NAT over other solutions is its simplicity of installation and configuration.

- No changes are required to routers or hosts other than the definition of additional IP addresses for the purposes of address translation.
- Current Internet protocols can be used without alteration.
- The NAT solution takes advantage of the fact that at any given time, very few hosts in a stub domain will be sending or receiving traffic outside of their domain.

6.1.4 Principles of Operation

A typical NAT application is based on the following:

- A NAT router is installed at the border of each stub domain in the network.
- A Translation Table is defined on each NAT router. Each entry of the table defines a correspondence between a local IP address (IP) and a globally unique address (NAT IP).
- Local IP addresses are defined at the stub domain level (the private, or internal network), and do not need to be globally unique. This means that the same IP address can be reused in other domains.

- NAT IP addresses are defined according to Classless InterDomain Routing (CIDR) address allocation schemes, which permits globally unique addresses.
- To provide transparent routing to the hosts, IP addresses are mapped to NAT IP addresses when communication outside of the stub domain is required.
- Since most hosts inside a stub domain never communicate outside of their domain, only a small subset of the IP addresses will ever need to be translated into NAT IP addresses.

6.1.5 Translation Schemes

There are two basic types of address translation schemes:

- **Many-to-many Address Translation:** The NAT router maintains a pool of IP addresses that are available to all users.
    - As long as no users are accessing the network, the pool of IP addresses remains unallocated.
    - When a user tries to access the network, the NAT router allocates one of the free IP addresses to this user, and performs address translation until the session is over and a certain period of time has elapsed.
    - Then the IP address becomes free again.
- **Many-to-One Address Translation:** The NAT router maintains a pool of TCP/UDP port numbers and ICMP query IDs that are available to all users.
    - The pool of port addresses is reserved as a block for the translation function.
    - As long as no users are accessing the network, the translation table entries themselves remain unallocated.
    - When a user tries to access the network, the NAT router allocates one of the free TCP/UDP ports or ICMP query IDs to this user, and performs address translation until the session is over and a certain period of time has elapsed.
    - Then the translation table entry for the port becomes free again.

6.1.6 NAT Device Characteristics

All NAT devices have the following characteristics.

- Transparent address assignment.
- Transparent routing through address translation.

NOTE: *Routing* here refers to forwarding packets, and not the exchange of routing information.

- ICMP error packet payload translation, including the following errors:
  - **Destination-Unreachable**
  - **Source-Quench**
  - **Time-Exceeded**
  - **Parameter-Problem**.

6.1.7 Transparent Address Assignment

NAT binds addresses in the private network with addresses in the global network and vice versa. This provides transparent routing for datagrams that are sent from one address realm to another. Address binding is done at the start of a session. Address assignments can be of two types:

- **Static Address Assignment:** There is a one-to-one address mapping for hosts between a private network address and an external network address for the entire duration of NAT operations. With static address assignment, address binding is fixed. Use it when you do not want NAT to handle address management with the session flows.
- **Dynamic Address Assignment:** External addresses are dynamically assigned to private network hosts or vice versa, based on usage requirements and session flow. With dynamic address assignment, address binding is dynamic at session startup. When the last session using address binding is terminated, NAT frees up the global address for use in another session.

Examples on the NetPerformer of these address assignment techniques are provided in the section “NAT Addressing Techniques on the NetPerformer” on page 6-14.

6.1.8 Transparent Routing through Address Translation

Transparent routing permits a datagram to be routed from one address realm to another. A NAT router is located at the stub border between two address realms, and modifies the IP headers to that the source and destination addresses remain valid for each address realm the datagram passes through. This address translation is carried out in three stages:

- **Address Binding:** The local IP address is associated with an external address, or vice versa, for translation purposes. New address bindings are made at the start of a new session. Once the binding between two addresses is in place, all subsequent sessions originating from or directed to the same local address will use the same binding.
- **Address Lookup and Translation:** Once a state is established for a session, all packets belonging to the session are subject to address lookup and translation.
- **Address Unbinding:** At this stage a private address is no longer associated with a global address for translation purposes. NAT performs address unbinding when it detects that the last session using address binding has been terminated.

6.1.9 ICMP Error Packet Translation

When ICMP error messages pass through NAT (with the exception of the Redirect message type) they are modified to permit complete transparency to the end hosts. These modifications involve changes to parts or all of the original IP packet that is embedded in the payload of the ICMP error message, including:

- The IP address of the IP header
- Checksum field of the same IP header
- The accompanying transport header.

The ICMP header checksum is also modified to reflect changes made to the IP and transport headers in the payload.

6.1.10 A Simple NAT Scenario

The NAT router function can be set up as shown in Figure 6-2. Only the stub border router requires modifications to its configuration. For the NetPerformer, these modifications are described in the section “Configuring for NAT” on page 6-18.

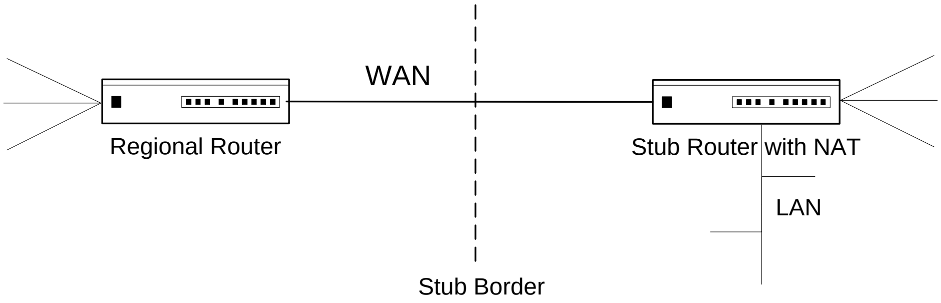


Figure 6-2: Simple NAT Scenario

The basic operation of NAT is as follows:

- Local IP addresses inside any stub domain can be reused by other stub domains.
- NAT must be installed at each exit point (or stub border) between a stub domain and the backbone.
- For each stub router with NAT, a translation table must be defined with all correspondences between local IP addresses and the globally unique NAT IP addresses.

  On the NetPerformer this is done by configuring a set of IP NAT Rules, as explained in the section “IP NAT Rule Parameters” on page 6-20.

NOTE: If there is more than one exit point from a particular stub domain, as in an Internet application, you must ensure that each NetPerformer has the same set of IP NAT Rules.

In the network shown in Figure 6-3, stubs A and B both use class A address 10.0.0.0 internally.

- For Stub A, NAT is assigned the class C address **205.81.34.0**. For Stub B, NAT is assigned the class C address **205.81.33.0**. These class C addresses are globally unique, and no other NAT routers can use them.

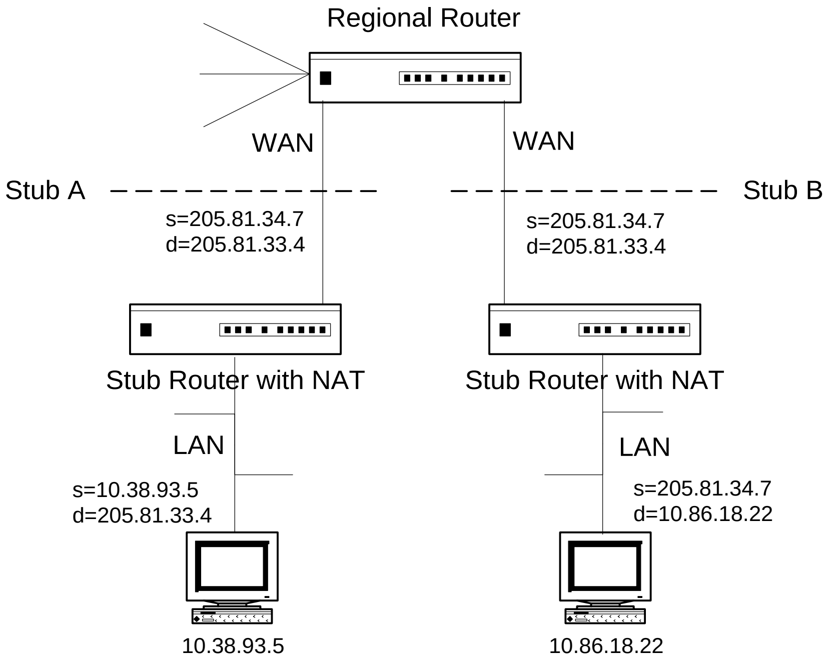


Figure 6-3: Basic NAT Operation

- In this example, when stub A host **10.38.93.5** wants to send a packet to stub B host **10.86.18.22**, it uses the globally unique address **205.81.33.4** as the destination in the IP header, and sends the packet to its primary router (the stub router with NAT).
- The stub router has a static route for network address **205.81.0.0**, so the packet is forwarded to the WAN link. Before transmission, NAT translates source address **10.38.93.5** with the globally unique **205.81.34.7** in the IP header.
- The stub B router with NAT receives the packet from the regional router via a second WAN link. Here the destination address in the IP header is translated back to **10.86.18.22** before being forwarded to its final destination, the stub B host.
- IP packets on the return path from the stub B to stub A hosts go through a similar translation of the source and destination addresses.

Even this simple example demonstrates the fact that no changes to hosts or routers are required for NAT to work. All address translations are carried out in a completely transparent way. For example, from the point of view of the stub A host, **205.81.33.4** is the address that is used by the host in stub B.

6.1.11 When NAT Should be Avoided

Although NAT provides a simple solution to IP address depletion, it is not always appropriate for all types of networks. NAT must be configured very carefully, or avoided altogether, for the following types of networks and protocols:

- Private networks (for example, a corporate network) that use a public backbone to communicate between locations. This type of application usually involves a large number of hosts communicating across the backbone, and requires a large translation table. Also, more applications in these networks will use configured addresses rather than going to a name server.

  A workaround using encapsulation at the border routers can be set up, with one global address at each NAT router used for tunneling through the backbone. This is supported on the NetPerformer using a NAPT configuration.

- Any network that does not have a sparse end-to-end traffic matrix. Once again, such networks require large NAT translation tables, which impedes overall performance.

- Encrypted protocols or applications with address content in the payload cannot be supported by NAT because of their dependence on the use of these addresses.

  If a protocol contains IP addresses or transport identifiers inside the packet payload, and the NetPerformer does not support the protocol used, the end results of NAT cannot be guaranteed. Refer to “Traffic Types Supported on the NetPerformer” on page 6-13.

- The fundamental role of NAT is to change the address in the IP header of a packet. Therefore, techniques that protect the content of IP headers (such as IPsec, AH and ESP) cannot be used in conjunction with NAT.

  Security techniques that do not depend on IP addresses will work correctly with NAT, for example, application layer techniques such as TLS, SSL and SSH. Also, end-to-end ESP-based transport mode authentication and confidentiality are permissible for packets such as ICMP, whose IP payload content is not affected by translation of the outer IP header.

6.1.12 Application Level Gateways (ALGs)

Application Level Gateways (ALGs) are application-specific translation agents that allow an application on a host in one address realm to transparently communicate with its counterpart on a host in a different realm. By itself, NAT cannot support all applications transparently and often must co-exist with one or more ALGs. For example, NAT cannot handle applications that include IP addresses or transport identifiers (TCP/UDP ports) in the payload. An ALG can interact with NAT for such things as setting up state, using NAT state information and modifying the payload.

Before installing a NAT-based solution, you must review all application requirements carefully and determine whether an ALG must be included with the configuration. An example NetPerformer application using DNS-ALGs is described in the section “Bidirectional NAT with DNS-ALGs” on page 6-45.

6.2 Varieties of NAT

Several varieties of NAT have been developed to handle various applications requiring address translation:

- Traditional, or Outbound NAT, including:
  - Basic NAT
  - Network Address Port Translation (NAPT)
- Bidirectional, or Two-way NAT
- Twice NAT
- Multihomed NAT

The main features of each of these NAT variations are described in the following sections, with the exception of Multihomed NAT, which is not supported by the NetPerformer.

The base model in Figure 6-4 illustrates the differences between these NAT types.

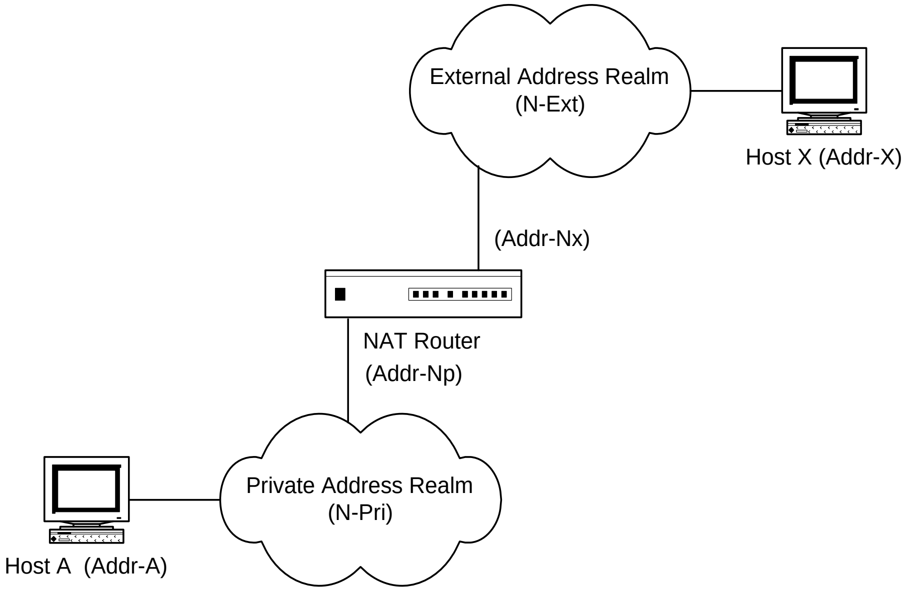


Figure 6-4: NAT Base Model

- Host A, with address Addr-A, is located in a private address realm (network N-Pri).
- A NAT router is located between N-Pri and the external address realm (N-Ext), and provides transparent routing between the two. It has two interfaces:
  - One interface to the private realm, with address Addr-Np,
  - One interface to the external realm, with address Addr-Nx.
- Host X, with address Addr-X, is located in the external realm N-Ext.
- Addresses Addr-A and Addr-Np belong to the private network N-Pri, and addresses Addr-X and Addr-Nx belong to the external network N-Ext.

6.2.1 Traditional NAT

Traditional NAT (also called Outbound NAT) allows hosts within a private network to transparently access hosts in the external network. It is primarily used by sites with private addresses that want outbound sessions from their site.

- Sessions are unidirectional, travelling outbound from the private network.
- The IP addresses of hosts in the external network are unique, and are valid in external as well as private networks.
- The addresses of hosts in the private network are unique only within their own domain, and may not be valid in the external network.
- NAT does not advertise private networks to the external realm, but networks from the external realm may be advertised within the private network.
- The addresses used within the private network must not overlap with the external addresses. Any given address cannot serve as both a private and external address.

A traditional NAT router in Figure 6-4 would allow Host A to initiate sessions to Host X, but not vice versa. Network N-Ext is routable from within N-Pri, but network N-Pri may not be routable from N-Ext.

6.2.2 Basic NAT

Basic NAT is a variation of traditional NAT in that many addresses are available for translation. However, only one IP address (source or destination) is actually translated before the packet leaves the NAT router.

NOTE: On the NetPerformer, this is accomplished by configuring only part of the IP NAT Rule submenu. Refer to "IP NAT Rule Parameters" on page 6-20.

- A block of external addresses is reserved for translating host addresses in the private network when sessions to the external network are initiated.
- Related fields such as the IP, TCP, UDP and ICMP header checksums are also translated.

A Basic NAT router in Figure 6-4 could be configured to translate N-Pri into a block of external addresses, for example, Addr-i through Addr-n, that are selected from external network N-Ext.

6.2.3 Network Address Port Translation (NAPT)

NAPT, also known as *Masquerading*, is another variation of traditional NAT. It carries the idea of translation one step further, by permitting the translation of transport identifiers such as TCP/UDP port numbers and ICMP query IDs.

- Only one *NAT source IP address* is used, and many IP addresses or transport identifiers can be "hidden" behind this single address.

  - The transport identifiers of a number of private hosts can be mapped to the transport identifiers of a single external address, which allows a set of hosts to share a single external address.
  - The greatest advantage of NAPT is that an entire network can directly access the Internet using only one official IP Address.
- A large number of connections are multiplexed using TCP port information.
  - The number of simultaneous connections is limited only by the number of TCP ports available and, on the NetPerformer, by the *NAPT pool size* parameter.

    The range of TCP ports and the *NAPT pool size* are configured on the NetPerformer with the IP NAT Port submenu. Refer to "IP NAT Port Parameters" on page 6-25.
  - With masquerading, incoming connections from an external network cannot be completed, even if a host has an entry in the translation table. This is because the entry is valid only for the currently active connection.

    For this reason, the NetPerformer supports NAPT on the source address only, not on the destination address.

NAPT Examples:

- A NAPT router in Figure 6-4 may be configured to translate sessions originated from N-Pri into a single external address, for example, Addr-i. Typically, the external interface address Addr-Nx of the NAPT router is used to map N-Pri.
- On the NetPerformer, a NAT Rule can be defined to masquerade an internal network, for example, **10.0.2.0** using the port 1 address.

  NAPT translation is accomplished by configuring the *NAT IP source address min* and *NAT IP source address max* parameters to the same value.
  - For each outgoing packet the source IP address is replaced by the NAT IP address (external) of port 1.
  - In addition, the source port is translated into an unused port from the range of ports reserved exclusively for masquerading on the router.
  - If the destination IP address of an incoming packet is the local port address (port 1) and the destination port is within the range of ports used for masquerading on the router, the NetPerformer checks its translation table to determine whether the packet belongs to a masqueraded session.
  - If the packet belongs to a masqueraded session, the destination IP address and port identifier of the internal host are inserted and the packet is forwarded to the internal host.

6.2.4 Bidirectional NAT

Bidirectional NAT permits sessions to be initiated from hosts in the public network as well as private networks. Hosts in the external address realm can access private realm hosts by using DNS for address resolution.

- Private network addresses are statically or dynamically bound to globally unique addresses as connections are established in either direction.
- The Fully Qualified Domain Names between hosts in the private and external networks must be unique from one end to the other.
- To permit name-to-address mapping, a DNS-ALG must be employed in conjunction with Bidirectional NAT. As DNS packets are sent between private and external realms, the DNS-ALG must be capable of translating private realm addresses in DNS queries and responses into their external realm address bindings, and vice versa.

A Bidirectional NAT router in Figure 6-4 would allow Host A to initiate sessions to Host X, and Host X to initiate sessions to Host A. As is the case with traditional NAT, external network N-Ext is routable from within N-Pri, but private network N-Pri may not be routable from N-Ext.

An example of Bidirectional NAT on the NetPerformer is provided in the section “Bidirectional NAT with DNS-ALGs” on page 6-45.

6.2.5 Twice NAT

Twice NAT allows both the source and destination addresses to be modified by NAT at the same time when a datagram is sent across address realms. This technique is typically used when the addresses used in a private network overlap with addresses used in the public network. It is required when private and external realms are subject to address collisions, for example, when internal nodes are improperly numbered with public addresses. It can also be used when a site changes from one Internet provider to another, and its users want to keep using its original internal addresses.

- The IP addresses of hosts in the external network are unique only within the external network.
- The addresses of hosts in the private network are unique only within their domain, and may not be valid in the external network.
- NAT must not advertise private networks to the external realm, or vice versa.

Twice-NAT operates in the following way:

- When Host A wants to initiate a session to Host X, it issues a DNS query for Host X.
- A DNS-ALG intercepts the DNS query, and replaces the address for Host X with one that is properly routable in the local site (the private, or internal network). The new address appears in the response returned to Host A.
- Host A then initiates communication with the new address for Host X. When packets are sent across the NAT router, the source IP address is translated as for

traditional NAT, and the destination address is translated to the actual address for Host X.

- A similar translation is performed on return packets from Host X.

A Twice NAT router in Figure 6-4 would allow Host A to initiate sessions to Host X, and Host X to initiate sessions to Host A. However, network N-Ext (or a subset of N-Ext) is not routable from within network N-Pri, and N-Pri is not routable from N-Ext.

6.2.6 Traffic Types Supported on the NetPerformer

The NetPerformer supports NAT translation of the following traffic types:

- TCP/UDP, unless it contains IP addresses inside the packet payload (see *Exceptions* below)
- HTTP and TFTP
- TELNET, archie, finger
- SIP VoIP
- NTP and NFS
- rlogin, rsh, rcp

Exceptions: The following protocols and functions are supported by the NetPerformer, even though IP addresses are contained in the application data stream:

- ICMP, including error packet payload translation of the following errors: **Destination-Unreachable**, **Source-Quench**, **Time-Exceeded**, **Parameter-Problem**
- FTP, including the **PORT** and **PASV** commands
- DNS **A** and **PTR** queries

NAT translation is *not supported* for the following traffic types:

- IP Multicast
- IP or IPX routing table updates
- SNMP
- BOOTP/DHCP
- DNS zone transfers
- talk, ntalk
- NetShow

Refer also to the section “When NAT Should be Avoided” on page 6-8.

6.3 NAT Addressing Techniques on the NetPerformer

The NetPerformer uses two basic NAT techniques (see also "Transparent Address Assignment" on page 6-5):

- Static NAT
- Dynamic NAT

6.3.1 Static NAT

With static address translation a specific, fixed, original IP address is always translated to the same NAT IP address at all times, and no other IP address is translated to that NAT IP address.

- This technique is ideal for translation between IP networks that have the same size, that is, networks that contain the same number of IP addresses (from 1 to 255).
- Static NAT is easy to implement, almost entirely transparent and can operate bidirectionally.
- With static NAT, a connection from a host in an external network to a local host in the internal network can be made without problem, as the two hosts simply appear to have different IP addresses.
- To configure the NetPerformer for Static NAT, set the *IP NAT Translation Type* parameter to STATIC. See "IP NAT Rule Parameters" on page 6-20.

SE/IP/NAT example: with STATIC Translation type

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 10
IP NAT RULE #10> Translation type (def:DYNAMIC) ? STATIC
IP NAT RULE #10> IP source address min (def:000.000.000.000) ?
10.0.1.0
IP NAT RULE #10> IP source address max (def:000.000.000.000) ?
10.0.1.255
IP NAT RULE #10> NAT IP source network (def:000.000.000.000) ?
205.168.43.0
IP NAT RULE #10> NAT IP source network mask (0-32,def:0) ? 0
{000.000.000.000}
IP NAT RULE #10> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #10> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #10> NAT IP destination network (def:000.000.000.000) ?
IP NAT RULE #10> NAT IP destination network mask (0-32,def:0) ?
{000.000.000.000}
IP NAT RULE #10> Pool size (0-2500,def:100) ?
```

NAT Rule: Statically translate all IP addresses **10.0.1.X** to NAT IP addresses **205.168.43.X**.

In this example, all packets with a source address of **10.0.1.X** will have their source address changed to the corresponding **205.168.43.X** address. For example, **10.0.1.15** is translated as **205.168.43.15**.

6.3.2 Dynamic NAT

With dynamic address translation, selection of the NAT IP address depends on various conditions, and each connection may have a different NAT IP address.

Dynamic address translation is required when:

- The number of IP addresses that must be translated does not equal the number of NAT IP addresses available for translation, *or*
- The same number of addresses is available, but static mapping is not desired for some other reason.

  NAPT is an extension of the dynamic NAT technique.

The number of hosts that can communicate at one time is limited by the number of NAT IP addresses that are available.

- When all NAT IP addresses are being used, no other IP address can be translated and the connection is rejected by the NetPerformer.
- Dynamic NAT is more complex to implement than static NAT, since the NetPerformer must keep track of which hosts are communicating and possibly the connections as well, which requires looking up TCP, UDP and ICMP information in the packets.

This technique may also be useful as a security measure for Internet applications, even in cases where there are enough NAT IP addresses to perform static mapping.

- It is highly unlikely that someone outside a network could get a useful IP address to connect to a host by observing a connection from a NAT router that is doing dynamic address translation, since the host may connect using a completely different IP address the next time.
- If security is an issue it may even make sense to have more NAT IP addresses than original IP addresses to be translated.

Connections originating from the external network are possible only when the internal host to be reached still has a NAT IP address assigned, that is, it still has an entry in the dynamic NAT translation table (where the NetPerformer keeps track of which internal IP address is mapped to which NAT IP address).

- If the internal host does not have an entry in the NAT translation table it is unreachable.
- If an entry exists in the NAT translation table that corresponds to the external host, but not to the correct internal host, the connection cannot be made.

NOTE: A bidirectional connection can also be set up using DNS (see "Bidirectional NAT with DNS-ALGs" on page 6-45).

Dynamic NAT Example:

- NAT Rule: Dynamically translate all IP addresses **132.212** (class B) to NAT IP addresses (class C) in network **205.168.100**.
- Each new connection from the internal host is assigned a NAT IP address from the pool of class C addresses, as long as unused addresses are still available.
- If a translation table entry already exists for a particular internal host it will be used again instead of assigning a new NAT IP address.
- As long as the address mapping exists as an entry in the translation table, the internal host can be reached from the external network via the NAT IP address that has been temporarily assigned to it.

SE/IP/NAT example: with DYNAMIC Translation type

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 10
IP NAT RULE #10> Translation type (def:DYNAMIC) ? DYNAMIC
IP NAT RULE #10> IP source address min (def:000.000.000.000) ?
132.212.0.0
IP NAT RULE #10> IP source address max (def:000.000.000.000) ?
132.212.255.255
IP NAT RULE #10> NAT IP source address min (def:000.000.000.000) ?
205.168.100.0
IP NAT RULE #10> NAT IP source address max (def:000.000.000.000) ?
205.168.100.255
IP NAT RULE #10> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #10> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #10> NAT IP destination address min
(def:000.000.000.000) ?
IP NAT RULE #10> NAT IP destination address max
(def:000.000.000.000) ?
IP NAT RULE #10> Pool size (0-2500,def:100) ?
```

6.3.3 NAT for SIP

As of version 10.1, the NetPerformer NAT techniques were extended to support SIP VoIP. The *NAT for SIP* feature allows a SIP connection to be opened between an external SIP proxy and a private User Agent (UA), with a media connection between an external UA and private UA. A typical network application is shown in Figure 6-5.

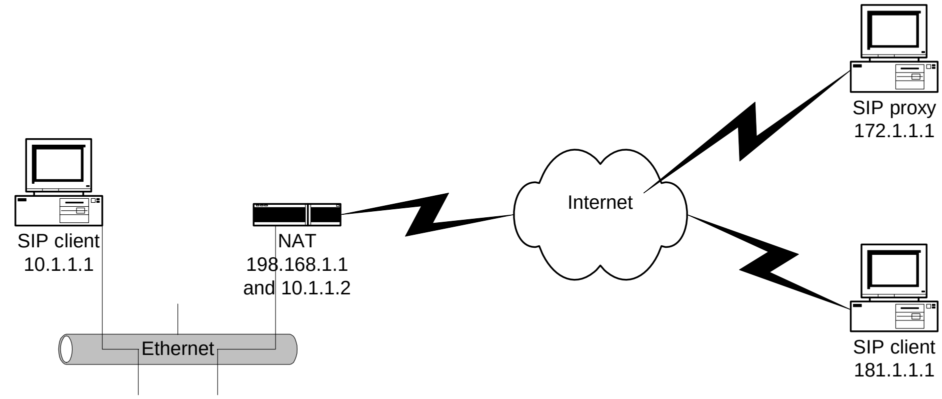


Figure 6-5: Typical NAT for SIP Application

In this setup, a SIP client (**10.1.1.1**) opens a connection with another SIP client (**181.1.1.1**) via the SIP proxy (**172.1.1.1**). To establish this connection the SIP Application Level Gateway (ALG) modifies all outgoing and incoming packets automatically.

NAT for SIP is configured in the same way as NAT for other traffic types. See the next section for complete configuration instructions.

⚠ **Caution:** In a NAT for SIP application, if you change the value of the *NAPT min port* or *NAPT max port* parameters **all voice lines participating in the address conversion will be dropped**. Refer to “Port Parameters for IP NAT Configuration” on page 6-25 for details on these parameters.

6.4 Configuring for NAT

6.4.1 Configuration Hints

The following points should be kept in mind when you set up your network for NAT.

- There must be no overlap between the local IP addresses on the internal network and the globally unique NAT IP addresses on the external network. Any given address must be clearly identifiable as either a local address or a global address.
- You must take great care to define all addresses and correspondences between IP and NAT IP addresses correctly, to avoid the routing problems associated with misaddressing.
- You can introduce NAT incrementally, if desired. If an existing stub domain is running out of unique internal addresses, you can change the addresses subnet by subnet to local IP addresses, and let NAT use the freed up addresses for communication outside the domain.
- The router running NAT should never advertise the internal networks to the backbone. Only those networks that have been assigned global addresses may be known outside the stub.

  Global information that NAT receives from a stub border router can be advertised in the stub domain in the usual way.

6.4.2 Port/PVC NAT Parameters

NAT parameters are defined on the LAN Port, the WAN Port (PVCR and PPP protocols only) and on PVCs (PVCR and RFC-1490 modes only). The following table summarizes the parameters that have been added to the Setup Slot/Port and Setup PVC menus of the NetPerformer console for configuration of LAN/WAN ports and PVCs on a NAT router.

Parameter	Range of Values	Default	Function
NAT enable *ifwanNat-Enable,* *iflanNat-Enable,* *pvcNatEnable*	YES, NO	NO	Enable (YES) or disable (NO) Network Address Translation on this port or PVC.

Table 6-1: Port/PVC Parameters for NAT Configuration

Parameter	Range of Values	Default	Function
NAT rule *ifwanNatRule,* *iflanNatRule,* *pvcNatRule*	1 to 10	undefined	If NAT is enabled, use this parameter to select the number (or numbers) of the rule that should be used to translate the address information for traffic to and from this port/PVC. • A rule defines the correspondence between internal IP addresses and external, globally unique NAT IP addresses. • Select multiple rules by entering a comma between the rule numbers, for example: 1,3,4.
NAT side *ifwanNatSide,* *iflanNatSide,* *pvcNatSide*	INTERNAL, EXTERNAL	INTERNAL	If NAT is enabled, use this parameter to determine with which address realm this port or PVC is associated. • Select INTERNAL if the port/ PVC connects to equip-ment on the internal, or local side (the private network). • Select EXTERNAL if the port/ PVC connects to equipment on the external network side.

Table 6-1: Port/PVC Parameters for NAT Configuration

6.4.3 IP NAT Parameters

A new NAT submenu of the Setup IP menu includes all NAT parameters that are required for IP NAT configuration. To access this submenu:

- Enter **SE**
- Enter **IP**
- Enter **NAT**.

There are four submenus in the IP NAT configuration:

- Enter **RULE** to define the NAT rules
- Enter **PORT** to define the NAPT parameters
- Enter **TIMEOUT** to define all required timeout parameters
- Enter **SERVICE** to define the NAT services used.

The parameters of each of these submenus are described in the following sections.

IP NAT Rule Parameters

The IP NAT Rules determine:

- The type of translation that takes place,
- Address ranges before and after translation,
- If more than one translation rule applies to a single packet, which rule will be executed.

  IP NAT Rules are applied in ascending numerical order. For example, an attempt to match a packet to **RULE #1** is executed before **RULE #10**. Although more than one IP NAT Rule may apply to a single frame, only one can be executed – the lowest numbered rule.

The following parameters are displayed when you select the Setup **IP/NAT RULE** item:

SE/IP/NAT/ RULE example: with DYNAMIC Translation type

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ?
IP NAT RULE #1> Translation type (def:DYNAMIC) ? DYNAMIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ?
IP NAT RULE #1> IP source address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP source address min (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP source address max (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination address min
(def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination address max
(def:000.000.000.000) ?
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

The NAT IP source and destination parameters are slightly different when the **STATIC** Translation Type is selected, as they refer to network rather than a smaller range of IP addresses:

SE/IP/NAT/ RULE example: with STATIC Translation type

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
```

```
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ?
IP NAT RULE #1> Translation type (def:DYNAMIC) ? STATIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ?
IP NAT RULE #1> IP source address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP source network (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP source network mask (0-32,def:0) ?
{000.000.000.000}
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network mask (0-32,def:0) ?
{000.000.000.000}
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

Refer to Table 6-2.

Parameter	Range of Values	Default	Function
IP NAT rule number *natIndex*	1 to 10	1	Use this parameter to select the number of the rule you want to configure. • A rule defines the correspondence between internal IP addresses and external, globally unique NAT IP addresses. • You can configure up to 10 NAT rules.
Translation type *natTransl-Type*	DYNAMIC, STATIC	DYNAMIC	Determines the translation type to be used by NAT. • Select STATIC for static address translation (see “Static NAT” on page 6-14). • Select DYNAMIC for dynamic address translation (see “Dynamic NAT” on page 6-15).
IP source address min *natIpSource-Min*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	Defines the lowest IP address in the range of IP source addresses that must be translated by NAT.
IP source address max *natIpSource-Max*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	Defines the highest IP address in the range of IP source addresses that must be translated by NAT.

Table 6-2: Rule Parameters for IP NAT Configuration

Parameter	Range of Values	Default	Function
NAT IP source address min *natNATIp-SourceMin*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For DYNAMIC Translation Type only.* Defines the lowest IP address in the range of NAT IP source addresses that are used to translate local (or internal) IP addresses.
NAT IP source address max *natNATIp-SourceMax*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For DYNAMIC Translation Type only.* Defines the highest IP address in the range of NAT IP source addresses that are used to translate internal IP addresses.
NAT IP source network *natNATIp-SourceNetwork*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For STATIC Translation Type only.* Defines the NAT IP address of the source network that is used to translate local (or internal) IP addresses.
NAT IP source network mask *natNATIp-SourceNetworkMask*	IP address (0.0.0.0 to 255.255.255.255) **Note:** Entered as 0 to 32	0 (0.0.0.0)	*For STATIC Translation Type only.* Defines the mask for the NAT IP address of the source network.
IP destination address min *natIpDest-Min*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	Defines the lowest IP address in the range of IP destination addresses that must be translated by NAT. **Note:** For basic NAT where the destination address is not translated, leave this parameter at 0.0.0.0.

Table 6-2: Rule Parameters for IP NAT Configuration

Parameter	Range of Values	Default	Function
IP destination address max *natIpDest-Max*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	Defines the highest IP address in the range of IP destination addresses that must be translated by NAT. **Note:** For basic NAT where the destination address is not translated, leave this parameter at 0.0.0.0.
NAT IP destination address min *natNATIp-DestMin*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For DYNAMIC Translation Type only.* Defines the lowest IP address in the range of NAT IP destination addresses that are used to translate local (or internal) IP addresses. **Note:** For basic NAT where the destination address is not translated, leave this parameter at 0.0.0.0.
NAT IP destination address max *natNATIp-DestMax*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For DYNAMIC Translation Type only.* Defines the highest IP address in the range of NAT IP destination addresses that are used to translate internal IP addresses. **Note:** For basic NAT where the destination address is not translated, leave this parameter at 0.0.0.0.

Table 6-2: Rule Parameters for IP NAT Configuration

Parameter	Range of Values	Default	Function
NAT IP destination network *natNATIp-DestNetwork*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	*For STATIC Translation Type only.* Defines the NAT IP address of the destination network that is used to translate local (or internal) IP addresses. **Note:** For basic NAT where the destination address is not translated, leave this parameter at 0.0.0.0.
NAT IP destination network mask *natNATIp-DestNetwork-Mask*	IP address (0.0.0.0 to 255.255.255.255) **Note:** Entered as 0 to 32	0 (0.0.0.0)	*For STATIC Translation Type only.* Defines the mask for the NAT IP address of the destination network.
Pool size *natPoolSize*	0 to 2500	100	Defines the maximum number of entries that can be maintained in the translation table for this particular rule.

Table 6-2: Rule Parameters for IP NAT Configuration

Configuration Notes for Defining NAT Rules:

- When a frame is received on a port that has the NAT function enabled, then the frame is parsed through all applicable NAT rules. If there is a match, then the frame is translated as indicated.
    - The number of the IP NAT rule is used to determine the priority of application, in ascending order. For example, if a packet matches both **RULE #1** and **RULE #10**, only **RULE #1** will be used for translation.
    - Although all IP NAT rules are scanned, a maximum of one rule can be applied in each direction, that is, one translation of the source address and one of the destination address.
- If the range of NAT IP Source Addresses is limited to one IP address (*NAT IP source address min = NAT IP source address max*), then the NetPerformer will automatically perform port translation (NAPT) and will overload the new IP address (only the source port is changed).
- If the range of NAT IP Source Addresses is greater than one IP address, no port translation will be performed.

- If both the *NAT IP source address min* and *NAT IP source address max* are set to 0.0.0.0, then the NetPerformer will use the IP address that is negotiated by default during link establishment.

  This method is available for a PPP port only.

- If the local IP Destination Addresses (*IP destination address min* and *IP destination address max*) are both set to **0.0.0.0**, no translation will be performed on the destination port.

IP NAT Port Parameters

The following parameters are displayed when you select the Setup **IP/NAT PORT** item:

SE/IP/NAT/ PORT example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? PORT
IP NAT PORT> NAPT min port (1-65534,def:60000) ?
IP NAT PORT> NAPT max port (1-65534,def:61000) ?
IP NAT PORT> NAPT pool size (0-2500,def:1000) ?
```

Refer to Table 6-3.

Parameter	Range of Values	Default	Function
NAPT min port *natNaptMin-Port*	1 to 65534	60000	Sets the minimum TCP/UDP port number or ICMP query ID to be used for NAPT translation. **Note:** NAPT translates the source address only. See “Network Address Port Translation (NAPT)” on page 6-10 for details. **Caution:** In a NAT for SIP application, if you change the value of this parameter **all voice lines participating in the address conversion will be dropped**.

Table 6-3: Port Parameters for IP NAT Configuration

Parameter	Range of Values	Default	Function
NAPT max port *natNapt-Max-Port*	1 to 65534	61000	Sets the maximum TCP/UDP port number or ICMP Query ID to be used for NAPT translation. ⚠ **Caution:** In a NAT for SIP application, if you change the value of this parameter **all voice lines participating in the address conversion will be dropped**.
NAPT pool size *natNapt-Pool-Size*	0 to 2500	1000	Defines the maximum number of entries that can be maintained in the translation table. **Note:** When ports are allocated, the *NAPT pool size* is used only if the range defined by the *NAPT min port* and *NAPT max port* parameters is greater than this value. Otherwise, the number of available ports determines the actual pool size.

Table 6-3: Port Parameters for IP NAT Configuration

If you are using a PPP port only, and the *NAT IP source address* parameters (*min* and *max*) for the NAT Rule are set to **0.0.0.0**, then the PPP port address will be used for NAPT translation.

IP NAT Timeout Parameters

The following parameters are displayed when you select the Setup **IP/NAT TIMEOUT** item:

SE/IP/NAT/ TIMEOUT example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? TIMEOUT
IP NAT TIMEOUT> Timeout (min) (1-10000,def:1440) ?
IP NAT TIMEOUT> TCP timeout (min) (1-10000,def:1440) ?
IP NAT TIMEOUT> TCP FIN/RST timeout (min)  (1-10000,def:1) ?
IP NAT TIMEOUT> TCP SYN timeout (min) (1-10000,def:2) ?
IP NAT TIMEOUT> DNS timeout (min) 1-10000,def:2) ?
IP NAT TIMEOUT> UDP timeout (min) (1-10000,def:5) ?
IP NAT TIMEOUT> ICMP timeout (min) (1-10000,def:2) ?
```

These parameters include the following:

Parameter	Range of Values	Default	Function
Timeout (min) *natTimeout*	1 to 10000	1440 (24 hours)	If NAPT is not configured (with the IP NAT Rule parameters, see "IP NAT Rule Parameters" on page 6-20), this timeout determines how long a NAT entry can remain inactive before it is removed from the translation table.
TCP timeout (min) *natTcpTimeout*	1 to 10000	1440 (24 hours)	If NAPT is configured, this timeout determines how long a TCP NAT entry (NAPT) can remain inactive before it is removed from the translation table.
TCP FIN/RST timeout (min) *natTcpFinRstTimeout*	1 to 10000	1	If NAPT is configured, this timeout determines how long a TCP NAT entry can remain in the translation table after an RST or FIN bit is detected. **Note:** The end of a TCP session is detected when FIN is acknowledged by both sides of the session, or when either side receives an RST bit in the TCP flags field.
TCP SYN timeout (min) *natTcpSynTimeout*	1 to 10000	2	If NAPT is configured, this timeout determines how long a TCP NAT entry can remain in the translation table after a SYN bit is detected and no further data is received. The presence of a SYN bit in the TCP flags indicates the first packet of a TCP session.

Table 6-4: Timeout Parameters for IP NAT Configuration

Parameter	Range of Values	Default	Function
DNS timeout (min) *natUdpTimeout*	1 to 10000	2	If NAPT is configured, this timeout determines how long a DNS NAT entry (TCP or UDP) can remain inactive before it is removed from the translation table.
UDP timeout (min) *natDnsTimeout*	1 to 10000	5	If NAPT is configured, this timeout determines how long a UDP NAT entry can remain inactive before it is removed from the translation table.
ICMP timeout (min) *natIcmp-Timeout*	1 to 10000	2	If NAPT is configured, this timeout determines how long an ICMP NAT entry can remain inactive before it is removed from the translation table.

Table 6-4: Timeout Parameters for IP NAT Configuration

IP NAT Service Parameters

The IP NAT Service parameters are used in NAPT to enable specific static port translation. This allows for a direct association between an external connection on a specific port and a specific internal address.

- Up to 10 IP NAT Services can be defined.
- A Service is used only when it is associated with a specific NAPT rule, and only on incoming external traffic.

  Twice NAT using NAPT is not supported.
- When a frame arrives, if an entry is not found in the translation table the NetPerformer verifies whether the frame corresponds to an IP NAT Service. If it does, it performs the port translation defined by the Service.
- This translation method allows access to, for example, an FTP or Internet server that is hidden behind a NAT (NAPT) translation process.

The following parameters are displayed when you select the Setup **IP/NAT SERVICE** item:

SE/IP/NAT/ SERVICE example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? SERVICE
```

```
IP NAT service (1/2/3/4/5/6/7/8/9/10,def:1) ?
IP NAT SERVICE #1> Associated rule number (0-10,def:0) ?
IP NAT SERVICE #1> Internal service address (def:000.000.000.000) ?
IP NAT SERVICE #1> Internal service port (1-65534,def:1) ?
IP NAT SERVICE #1> External service port (1-65534,def:1) ?
```

These parameters include the following:

Parameter	Range of Values	Default	Function
IP NAT service *natServIndex*	1 to 10	1	Use this parameter to select the number of the service that is defined with the following parameters. You can configure up to 10 IP NAT services.
Associated rule number *natAssociate-Rule*	0 to 10	0	Use this parameter to select the number of the NAPT rule that is associated with this IP NAT service. Select 1 to 10 for a specific rule number, or 0 to disable any association with a rule.
Internal service address *natIntAddr*	IP address (0.0.0.0 to 255.255.255.255)	0.0.0.0	Defines the IP address of the internal service to be reached by the external service port.
Internal service port *natIntPort*	1 to 65534	1	Defines the port that corresponds to the internal service address.
External service port *natExtPort*	1 to 65534	1	Defines the external port to be used to reach the internal service address.

Table 6-5: Service Parameters for IP NAT Configuration

Once NAT is configured in NAPT, each incoming frame is processed by the NAT services.

- If no entry is found in the translation table, the NAT service parses the frame and tries to detect a matching situation.
- If a match is found, the NAT service creates an entry in the translation table and performs the translation.
- If there is no match, the frame is discarded.

6.5 NAT Status and Statistics

NOTE: No special capture procedures are required to capture NAT transmissions. To observe the results of network address translation, use the Port or PVC option of the Start Capture (**SC**) command, and view with the View Capture (**VC**) command.

6.5.1 Display NAT Table Information (DN)

The Display NAT Table Information (**DN**) command displays the NAT Translation Table. To execute this command enter **DN** at the NetPerformer console command prompt.

The translation table includes:

- The source and destination IP and NAT IP addresses
- The source and destination TCP/UDP and NAT (NAPT) ports
- The current timeout for each address mapping.

In the following example, the first two entries are in NAT, and the last is in NAPT mode.

DN example

```
NATROUTER>DN
DISPLAY NAT TABLE INFORMATION
IP NAT rule number (1/2/3/4/5/6/7/8/9/10/ALL,def:1) ? 1

Active translation entries:

             Rule        IP              Nat IP         Port   Nat Port  Timeout

Source     :   1   010.000.000.001   205.168.043.010       0        0
Destination:       206.001.001.001   206.001.001.001       0        0      1440

Source     :   1   010.000.000.003   205.168.043.012       0        0
Destination:       206.001.001.002   206.001.001.002       0        0      1439

Source     :   1   010.000.000.003   205.168.043.012      21    60000
Destination:       206.001.001.003   206.001.001.003      21       21      1439

NATROUTER>
```

6.5.2 Clear NAT Entry (CN)

The Clear NAT Entry (**CN**) command removes one or more entries from the NAT Translation Table. To execute this command enter **CN** at the NetPerformer console command prompt, then select **SINGLE**, **RULE** or **ALL** entries.

NOTE: If your selection does not include any translation table entries, the NetPerformer console returns the message **Entry(ies) not found**.

Clearing a Single Entry

Enter **SINGLE** at the *Entry type* prompt to clear a single NAT entry from the translation table. You must define the entry you want to delete by specifying the following:

- *Rule number*
- *Source IP address* and *NAT source IP address*
- *Source port* and *NAT Source port*
- *Destination IP address* and *NAT destination IP address*
- *Destination port* and *NAT destination port*.

Confirm deletion of one or more entries from the translation table by entering **YES** at the confirmation prompt.

CN/SINGLE example

```
NATROUTER>CN
CLEAR NAT ENTRY
Entry type (SINGLE/RULE/ALL,def:ALL) ? SINGLE
Rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
Source IP address (def:000.000.000.000) ? 10.0.0.100
NAT source IP address (def:000.000.000.000) ? 205.168.43.100
Source port (0-65534,def:0) ? 512
NAT Source port (0-65534,def:0) ? 60004
Destination IP address (def:000.000.000.000) ? 145.256.1.10
NAT destination IP address (def:000.000.000.000) ? 145.256.1.10
Destination port (0-65534,def:0) ? 512
NAT destination port (0-65534,def:0) ? 60004
Clear NAT entries, please confirm (NO/YES,def:NO) ? YES
Entry(ies) deleted
```

Clearing a Rule

Enter **RULE** at the *Entry type* prompt to clear all entries that have been created with a particular rule. You must define the rule you want to delete by specifying the *IP NAT rule number*, and confirm deletion of the NAT entries, as in the following example:

CN/RULE example

```
NATROUTER >CN
CLEAR NAT ENTRY
Entry type (SINGLE/RULE/ALL,def:SINGLE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
Clear NAT entries, please confirm (NO/YES,def:NO) ? YES
Entry(ies) deleted
```

Clearing All Entries

Enter **ALL** at the *Entry type* prompt to clear all entries from the NAT Translation Table. You must confirm the clearing of all NAT entries.

CN/ALL example

```
NATROUTER >CN
CLEAR NAT ENTRY
Entry type (SINGLE/RULE/ALL,def:SINGLE) ? ALL
Clear NAT entries, please confirm (NO/YES,def:NO) ? YES
Entry(ies) deleted
```

⚠ **Caution:** In a NAT for SIP application, be very careful when executing the *Clearing All Entries* command, as **it will clear all lines participating in the address conversion.**

6.5.3 Display Counters (DC)

To view the counters pertaining to NAT translation, enter **NAT** after the Display Counters (**DC**) command *Item* prompt, as in the following example.

DC/NAT example

```
NATROUTER>DC
DISPLAY COUNTERS
Item (PORT/PVC/IP/BOOTP/TIMEP/SLOT/NAT/SVC/Q922/Q933,def:PORT) ?
NAT
IP NAT rule number (1/2/3/4/5/6/7/8/9/10/ALL,def:1) ? 1
Rule 1> Number of translation entries currently free          90
Rule 1> Number of translation entries currently active        10
Rule 1> Number of DNS entries currently free                  99
Rule 1> Number of DNS entries currently active                 1
Rule 1> Number of translation requests accepted             2676
Rule 1> Number of translation requests discarded               2
```

These statistics indicate the following:

Statistic	Indication
Number of translation entries currently free *statIfnatTranslEntryFree*	The number of entries that are available to perform translation, but are not currently used. **Note:** This number is defined by the *Pool size* parameter for the IP NAT Rule.
Number of translation entries currently active *statIfnatTranslEntryActive*	The number of entries that are used for this rule. **Note:** To view these entries, use the DN command (see “Display NAT Table Information (DN)” on page 6-30).
Number of DNS entries currently free *statIfnatDnsEntryFree*	The number of DNS temporary bindings that are available for future translation. **Note:** This number is defined by the *Pool size* parameter for the IP NAT Rule.
Number of DNS entries currently active *statIfnatDnsEntryActive*	The number of temporary bindings that are currently being used.
Number of translation requests accepted *statIfnatTranslReqAccept*	The number of frames that have been correctly translated using NAT.
Number of translation requests discarded *statIfnatTranslReqDiscard*	The number of frames that have been discarded during NAT translation. **Note:** To find out why frames have been discarded, use the DE command (see next section).

Table 6-6: NAT Counters

6.5.4 Display Errors (DE)

To view the errors that have occurred during NAT translation, enter **NAT** after the Display Errors (**DE**) command *Item* prompt, as in the following example.

DE/NAT example

```
NATROUTER>DE
DISPLAY ERRORS
Item (PORT/PVC/PU/GROUP/CHANNEL/DICT/BOOTP/TIMEP/SLOT/SVC/Q922/
NAT,def:PORT) ? NAT
IP NAT rule number (1/2/3/4/5/6/7/8/9/10/ALL,def:1) ? 1
Rule 1> No more source address                              0
Rule 1> No more destination address                         0
Rule 1> No more source port                                 0
Rule 1> No match entry found                                2
Rule 1> Entry maximum pool reached                          0
Rule 1> DNS entry maximum pool reached                      0
Rule 1> Entry allocation failed                             0
Rule 1> Fragment entry allocation failed                    0
Rule 1> Sequence entry allocation failed                    0
Rule 1> DNS entry allocation failed                         0
```

These statistics indicate the following:

Statistic	Indication
No more source address *statIfnatNoSourceAddr*	The number of times that the NAT source address pool has gone empty. To solve this problem: • Increase the value of the *Pool size* parameter, ***and*** • Adjust the Rule parameters *NAT IP source address min* and *NAT IP source address max* if the range of addresses they define is smaller than the value of the *Pool size* parameter. Refer to "IP NAT Rule Parameters" on page 6-20.

Table 6-7: NAT Errors

Statistic	Indication
No more destination address *statIfnatNoDestAddr*	The number of times that the NAT destination address pool has gone empty. To solve this problem: • Increase the value of the *Pool size* parameter, ***and*** • Adjust the Rule parameters *NAT IP destination address min* and *NAT IP destination address max* if the range of addresses they define is smaller than the value of the *Pool size* parameter. Refer to "IP NAT Rule Parameters" on page 6-20.
No more source port *statIfnatNoSourcePort*	The number of times that the NAPT port pool has gone empty. To solve this problem: • Increase the value of the *NAPT pool size* parameter, ***and*** • Adjust the NAT IP Port parameters *NAPT min port* and *NAPT max port* if the range of addresses they define is smaller than the value of the *NAPT pool size* parameter. Refer to "IP NAT Port Parameters" on page 6-25.
No match entry found *statIfnatNoMatchEntry*	The number of times that no match was found in the translation table and the packet was discarded.
Entry maximum pool reached *statIfnatMaxPoolReach*	The number of times that the maximum allowable number of entries for this rule was reached and the packet was discarded.
DNS entry maximum pool reached *statIfnatDnsEntryMaxPoolReach*	The number of times that the maximum allowable number of DNS temporary bindings was reached and the packet was discarded.
Entry allocation failed *statIfnatEntryAllocFail*	The number of times that an entry could not be allocated. In most cases this is due to a lack of memory.
Fragment entry allocation failed *statIfnatFragEntryAllocFail*	The number of times that an entry could not be allocated. In most cases this is due to a lack of memory.

Table 6-7: NAT Errors

Statistic	Indication
Sequence entry allocation failed *statIfnatSeqEntryAllocFail*	The number of times that an entry could not be allocated. In most cases this is due to a lack of memory.
DNS entry allocation failed *statIfnatDnsEntryAllocFail*	The number of times that an entry could not be allocated. In most cases this is due to a lack of memory.

Table 6-7: NAT Errors

6.6 Extended Parameters

A new type of extended parameters, **EP NAT**, has been added to the Extended Parameters (**EP**) command. **EP NAT** includes the following extended parameters.

6.6.1 DNS

Use the **DNS** extended parameter to set the timeout of a special DNS entry used for temporary binding.

Syntax:

To set the **DNS** extended parameter enter the following at the NetPerformer console command prompt:

EP NAT DNS *n*

where: *n* is the number of minutes for the timeout

Range of Values:

1 to **10000** minutes

Default Value:

1 minute

6.6.2 FRAGMENT

Use the **FRAGMENT** extended parameter to set the timeout of a special translation table entry that is created by fragmented packets. These special entries are not displayed with the Display NAT Table Information (**DN**) command and are intended for use during troubleshooting procedures by NetPerformer technical personnel only.

Syntax:

To set the **FRAGMENT** extended parameter enter the following at the NetPerformer console command prompt:

EP NAT FRAGMENT *n*

where: *n* is the number of minutes for the timeout

Range of Values:

1 to **10000** minutes

Default Value:

4 minutes

6.6.3 SEQUENCE

Use the **SEQUENCE** extended parameter to set the timeout of a special translation table entry that is created during translation of an FTP PORT command packet. These special entries are not displayed with the **DN** command and are intended for use during troubleshooting procedures by NetPerformer technical personnel only.

Syntax:

To set the **SEQUENCE** extended parameter enter the following at the NetPerformer console command prompt:

EP NAT SEQUENCE *n*

where: *n* is the number of minutes for the timeout

Range of Values:

1 to **10000** minutes

Default Value:

1440 minutes

6.7 Application Examples

This section describes three example applications using NAT: Internet Sharing, Service Extension Sharing and Bidirectional NAT with DNS-ALGs, and provides the required NetPerformer parameter configuration for each.

6.7.1 Internet Sharing

In this application multiple users share an Internet connection using a single Internet IP address, thus overloading the IP address using NAPT.

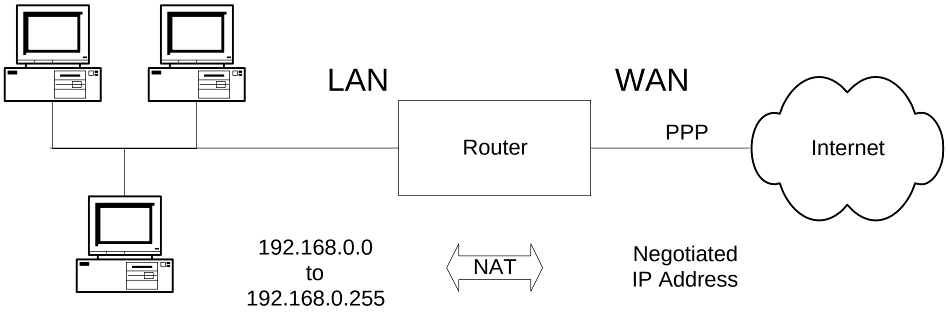


Figure 6-6: Internet Sharing Application

Frames coming from the LAN are automatically routed to the PPP port if the Destination IP is defined for the Internet, so NAT needs to be configured on the PPP port only. The LAN port does not require NAT configuration.

All frames outgoing to PPP port are intercepted by NAT to verify if the *Source IP Address* of the frame matches the Rule configured on the PPP port. This address is translated to the *NAT source IP address* if there is a match. After translation, PPP negotiation determines the IP address used on the Internet side.

In the opposite direction, when the PPP port receives frames from the Internet, the NetPerformer searches its translation table for a match between the *Destination IP address* of the frame and the *NAT source IP address* in the pool. If there is a match, the destination IP address of the frame is translated to the original *Source IP address*.

NOTE: This example shows translation being applied on a single address only. In fact, the NetPerformer scans the translation table for all possible address/port matches, including the source and destination IP address and the source and destination port (for TCP/UDP or ICMP query ID).

6.7.2 Configuring the PPP Port

Here is how the PPP port should be configured for this application:

SE/PORT/#/ PPP example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (ETH/CSL/1/2,def:ETH) ? 1
PORT #1> Protocol (def:PPP) ? PPP
...
PORT #1> IP address (def:000.000.000.000) ?
PORT #1> Subnet mask (number of bits) (0-32,def:8) ?  {255.000.000.000}
PORT #1> IP RIP (def:V1) ?
PORT #1> IP RIP TX/RX (def:DUPLEX) ?
PORT #1> NAT enable (def:NO) ? YES
PORT #1> NAT rule (1-10) (def:) ? 1
PORT #1> NAT side (def:INTERNAL) ? EXTERNAL
PORT #1> Silent (def:SEND REQUEST) ?
...
IPDIAL 1> Authentication timeout (s) (1-255,def:10) ?
```

6.7.3 Configuring the IP NAT Rule

A single IP NAT rule needs to be configured, as follows:

SE/IP/NAT/ RULE example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
IP NAT RULE #1> Translation type (def:DYNAMIC) ? DYNAMIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ? 192.168.0.0
IP NAT RULE #1> IP source address max (def:000.000.000.000) ? 192.168.0.255
IP NAT RULE #1> NAT IP source address min (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP source address max (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

- By setting the *NAT IP source address min* and *NAT IP source address max* parameters to **0.0.0.0**, the NAT function is able to determine that the PPP negotiated address must be used on the Internet side.

  This will work only if the port is defined with the PPP protocol.

- All packets coming from the LAN with addresses **192.168.0.0** to **192.168.0.255**, and that have the Internet as their destination will have their IP Source addresses translated to the *Negotiated IP Address* of the PPP port.

- Since there is address overloading, the transport identifier (for example, the TCP and UDP port numbers and ICMP query identifiers) are translated too.

For example, if the *Negotiated IP Address* of the PPP port is **205.168.43.10**, then a packet coming from the LAN with source address **192.168.0.10**, destination address **132.212.1.2** and TCP source/destination port **21** will be translated by NAT to a packet with source address **205.168.43.10**, destination address **132.212.1.2**, TCP source port **60000** and TCP destination port **21**.

NOTE: The TCP source port number (**60000**) is an arbitrary number. It can be changed using the IP NAT port parameters *NAPT min port* and *NAPT max port*. Refer to "IP NAT Port Parameters" on page 6-25.

6.7.4 Service Extension Sharing

In this application the server **10.1.1.1** is located on the service provider premises and must be reachable by different subscribers via a dedicated Frame Relay network.

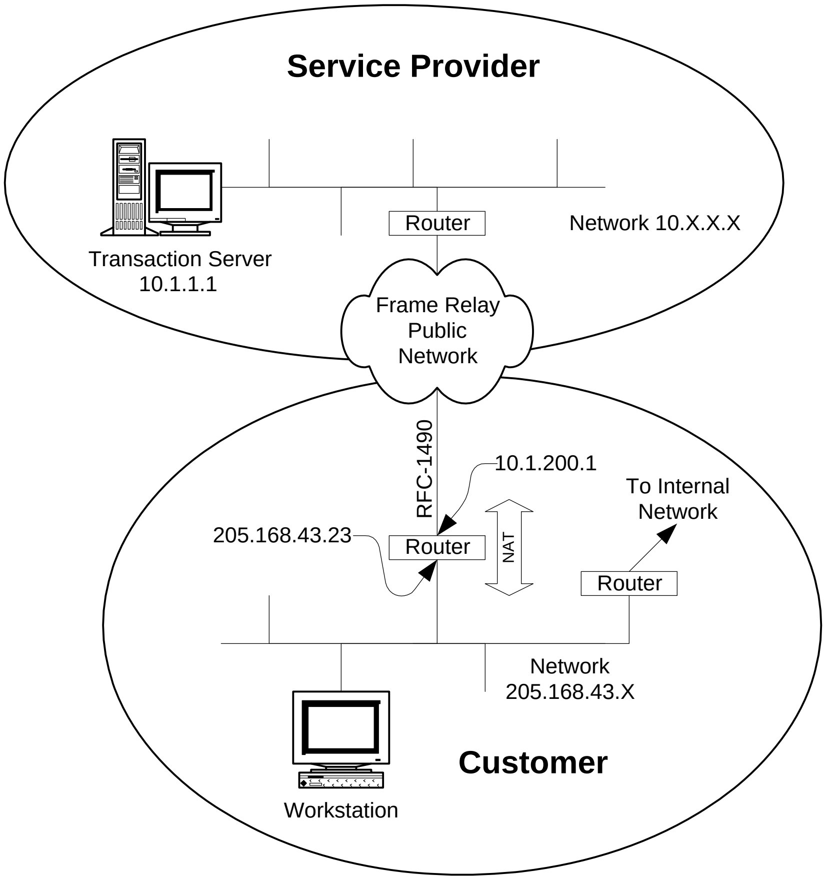


Figure 6-7: Service Extension Sharing

NOTE: The configuration provided here is one possibility among many. It should be taken as an example approach.

In the application portrayed in Figure 6-7, each subscriber has its own private or public IP network. The IP address of the point of presence at the customer location must be translated into the final IP address of the server located in the service provider computer center.

Local users access the service provider’s transaction server using IP address **205.168.43.23**. The NetPerformer translates both the source and destination addresses to ensure that the transaction server can be reached.

Configuring the LAN Port

The LAN port is configured so that NAT is enabled and operates according to **RULE #1**.

SE/PORT/ETH example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (ETH/CSL/1/2,def:ETH) ? ETH
PORT ETH> Protocol (def:ETH AUTO) ?
PORT ETH> MAC address (def:000000000000) ?
PORT ETH> IP address (def:000.000.000.000) ? 205.168.43.23
PORT ETH> Subnet mask (number of bits) (0-23,def:8) ?  24 {255.255.255.000}
...
PORT ETH> IP multicast protocol (def:NONE) ?
PORT ETH> NAT enable (def:NO) ? YES
PORT ETH> NAT rule (0-10,def:) ? 1
PORT ETH> NAT side (def:INTERNAL) ? INTERNAL
PORT ETH> IPX RIP (def:DISABLE) ?
...
PORT ETH> IPX encapsulation (def:ETH 802.2) ?
```

Configuring the Frame Relay Port

The Frame Relay port is defined with the **FR-USER** protocol, following the usual NetPerformer configuration procedure.

SE/PORT/#/FR-USER example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (ETH/CSL/1/2,def:ETH) ? 1
PORT #1> Protocol (def:PPP) ? FR-USER
PORT #1> Port speed (bps) (1200-2048000,def:56000) ?
...
```

Configuring the PVC

The PVC is configured so that NAT is enabled, operates according to **RULE #2**, and is associated with the external side of the NAT router.

SE/PVC/#/RFC1490 example

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PVC
PVC number (1-300,def:1) ?
PVC #1> Mode (def:RFC1490) ?
PVC #1> Port (def:1) ?
PVC #1> DLCI address (0-1022,def:0) ? 100
...
PVC #1> IP address (def:000.000.000.000) ? 10.1.200.1
PVC #1> Subnet mask (number of bits) (0-32,def:8) ?  16  {255.255.000.000}
...
PVC #1> IP multicast protocol (def:NONE) ?
PVC #1> NAT enable (def:NO) ? YES
PVC #1> NAT rule (0-10,def:1) ? 2
PVC #1> NAT side (def:INTERNAL) ? EXTERNAL
PVC #1> IPX RIP (def:DISABLE) ?
...
PVC #1> LLC connection (def:YES) ?
```

Configuring the Rules

NAT **RULE #1** must be configured so that all frames going to the NetPerformer LAN port and coming from any PC (with *Destination IP address* **205.168.43.23**) on the customer's LAN will have their destination address translated to **10.1.1.1**. The NAT router redirects this new frame to port 1 (the **10.0.0.0** route). Before transmission, NAT intercepts the frame again to verify the *Source IP address*, and translates it to **10.1.200.1** using NAT Rule 2.

In the opposite direction, when incoming frames arrive on port 1 NAT verifies whether the frame destination matches any *NAT source IP address* in the translation table, and then changes it to the corresponding *Source IP address*. Frames are routed to the correct port (LAN), and then intercepted by NAT before transmission to verify whether their *Source IP* *address* matches any *NAT destination IP address*. If a match is found, the *Source IP address* is changed to the corresponding *Destination IP address*.

The configuration of **RULE #1** for this example application is as follows:

SE/IP/NAT/ RULE example: for Rule #1

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
IP NAT RULE #1> Translation type (def:DYNAMIC) ? DYNAMIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #1> IP source address max (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #1> NAT IP source address min (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #1> NAT IP source address max (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ? 205.168.43.23
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ? 205.168.43.23
IP NAT RULE #1> NAT IP destination address min (def:000.000.000.000) ? 10.1.1.1
IP NAT RULE #1> NAT IP destination address max (def:000.000.000.000) ? 10.1.1.1
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

Here is the configuration for NAT **RULE #2**, which is applied to the PVC:

SE/IP/NAT/ RULE example: for Rule #2

```
NATROUTER>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 2
IP NAT RULE #2> Translation type (def:DYNAMIC) ? DYNAMIC
IP NAT RULE #2> IP source address min (def:000.000.000.000) ? 205.168.43.1
IP NAT RULE #2> IP source address max (def:000.000.000.000) ? 205.168.43.255
IP NAT RULE #2> NAT IP source address min (def:000.000.000.000) ? 10.1.200.1
IP NAT RULE #2> NAT IP source address max (def:000.000.000.000) ? 10.1.200.1
IP NAT RULE #2> IP destination address min (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #2> IP destination address max (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #2> NAT IP destination address min (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #2> NAT IP destination address max (def:000.000.000.000) ? 0.0.0.0
IP NAT RULE #2> Pool size (0-2500,def:100) ?
```

6.7.5 Bidirectional NAT with DNS-ALGs

In the application portrayed in Figure 6-8, host **x.abc.com** wants to communicate bidirectionally with host **y.xyz.com** via DNS-ALGs. The goal of the NAT configuration is to:

- Dynamically translate all internally originated traffic from domain **abc.com** having a class A source address of the form **10/8** to a class C address of the form **195.16.1/24** for transmission to the Internet.
- Dynamically translate all internally originated traffic from domain **xyz.com** having a class A source address of the form **10/8** to a class C address of the form **196.168.1/24** for transmission to the Internet.
- With this configuration, all hosts in the domain **abc.com** will be able to communicate with all hosts in the domain **xyz.com** and vice-versa using their DNS servers.

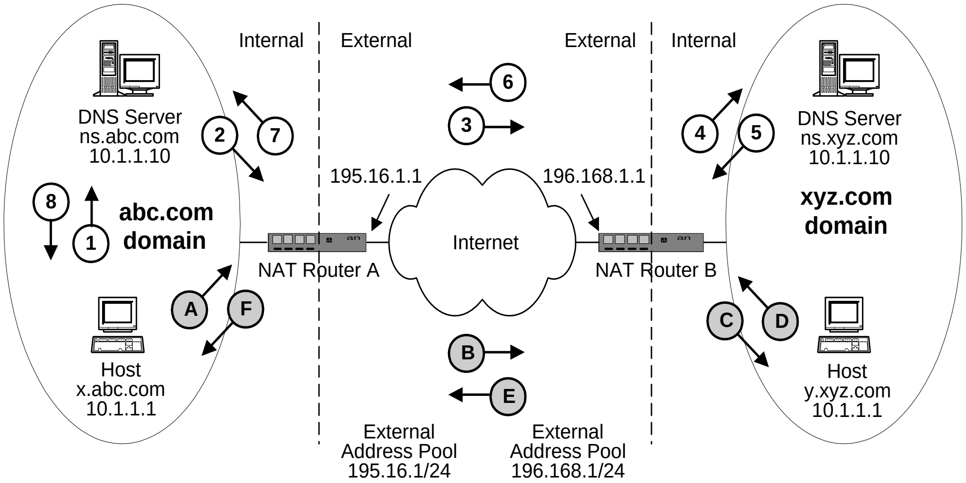


Figure 6-8: Bidirectional NAT

NOTE: To simplify the explanation of how a packet travels through this network, it is assumed that the two DNS routers (labelled **NAT Router A** and **NAT Router B** in Figure 6-8) can communicate directly with each other.

DNS Name Lookup Query

Before host **x.abc.com** can communicate with host **y.xyz.com**, it must resolve the host name *y.xyz.com* using a name lookup query sent via the NAT routers. The path of the DNS name lookup query has 8 distinct stages, represented in Figure 6-8 as numbers 1 through 8.

1. First, host **x.abc.com** sends a name lookup query to DNS server **ns.abc.com** to resolve the host name *y.xyz.com*:

Source:	10.1.1.1
Destination:	10.1.1.10
DNS Payload:	y.xyz.com

2. DNS server **ns.abc.com** sends the name lookup query to the remote site DNS server, **ns.xyz.com**, for host **y.xyz.com**:

Source:	10.1.1.10
Destination:	196.168.1.254
DNS Payload:	y.xyz.com

3. Before sending the packet on the Internet, **NAT Router A** uses a static rule to change the source address of the DNS server **ns.abc.com** to **195.16.1.254**, and creates an entry in its translation table:

Source:	195.16.1.254
Destination:	196.168.1.254
DNS Payload:	y.xyz.com

The DNS-ALG does not change the packet contents.

D. When **NAT Router B** receives the packet, it uses a static rule for DNS server **ns.xyz.com** to change the destination address to **10.1.1.10**, creates an entry in its translation table, and routes the packet to domain **xyz.com**:

Source:	195.16.1.254
Destination:	10.1.1.10
DNS Payload:	y.xyz.com

5. DNS server **ns.xyz.com** answers the name lookup query with the address of host **y.xyz.com** in the DNS payload:

Source:	10.1.1.10
Destination:	195.16.1.254
DNS Payload:	10.1.1.1

6. Before sending the packet on the Internet, **NAT Router B** uses the static rule for the DNS server **ns.xyz.com** to change the packet source address to **196.168.1.254**. The DNS-ALG also intercepts the packet to create a temporary binding, using a free address from the external address pool. At this stage, no entry is created in the translation table, only a binding:

Source:	196.168.1.254
Destination:	195.16.1.254
DNS Payload:	196.168.1.25 (free address in the pool)

7. Before **NAT Router A** sends the packet to domain **abc.com**, it changes the destination address back to **10.1.1.10** using the entry in its translation table that was created by the static rule in step **3**:

Source:	196.168.1.254
Destination:	10.1.1.10
DNS Payload:	196.168.1.25

8. Finally, DNS server **ns.abc.com** forwards the answer for the name lookup query to host **x.abc.com**:

Source:	10.1.1.10
Destination:	10.1.1.1
DNS Payload:	196.168.1.25

Packet Flow

Once the answer for the name lookup query has been received by host **x.abc.com**, it is able to determine that the host name *y.xyz.com* corresponds to destination address **196.168.1.25**. It can then proceed with the packet flow, which is represented in Figure 6-8 as letters **A** through **F**.

A. Host **x.abc.com** sends a packet to host **y.xyz.com**:

Source:	10.1.1.1
Destination:	196.168.1.25

B. Before **NAT Router A** sends the packet on the Internet, it changes the source address using a dynamic address taken from the pool of available addresses, in this case, **195.16.1.12**. An entry is created in the translation table using the new allocated address **195.16.1.12**:

Source:	195.16.1.12
Destination:	196.168.1.25

C. When **NAT Router B** receives the packet, the temporary binding for address **196.168.1.25** is changed into a normal entry using the temporary binding information and packet. When this entry is created, the packet destination is reverted to the real address of host **y.xyz.com** (**10.1.1.1**):

Source:	195.16.1.12
Destination:	10.1.1.1

D. Host **y.xyz.com** now sends a packet to host **x.abc.com** (**195.16.1.12**):

Source:	10.1.1.1
Destination:	195.16.1.12

E. Before **NAT Router B** sends the packet on the Internet, it changes the source address to **196.168.1.25** using the translation table entry created in step **C**:

Source:	196.168.1.25
Destination:	195.16.1.12

F. **NAT Router A** receives the packet and translates the destination address back to **10.1.1.1** using the translation table entry created in step **B**:

Source:	196.168.1.25
Destination:	10.1.1.1

Translation Tables

The final translation table for **NAT Router A** would look like the following:

DN/ALL example: for NAT Router A

```
NATROUTER_A>DN
DISPLAY NAT TABLE INFORMATION
IP NAT rule number (1/2/3/4/5/6/7/8/9/10/ALL,def:1) ? ALL

Active translation entries:

             Rule        IP              Nat IP         Port   Nat Port  Timeout

Source     :   1   010.001.001.010   195.016.001.254       0        0
Destination:       196.168.001.254   196.168.001.254       0        0      1440

Source     :   2   010.001.001.001   195.016.001.012       0        0
Destination:       196.168.001.025   196.168.001.025       0        0      1440
```

The final translation table for **NAT Router B** would look like the following:

DN/ALL example: for NAT Router B

```
NATROUTER_B>DN
DISPLAY NAT TABLE INFORMATION
IP NAT rule number (1/2/3/4/5/6/7/8/9/10/ALL,def:1) ? ALL

Active translation entries:

             Rule        IP              Nat IP         Port   Nat Port  Timeout

Source     :   1   010.001.001.010   196.168.001.254       0        0
Destination:       196.016.001.254   195.016.001.254       0        0      1440

Source     :   2   010.001.001.001   196.168.001.025       0        0
Destination:       195.016.001.012   195.016.001.012       0        0      1440
```

DNS Server Configuration

In this application example, DNS Server **ns.abc.com** has the following configuration:

ns.abc.com	10.1.1.10
x.abc.com	10.1.1.1
nat-ns.abc.com	195.16.1.254

DNS Server **ns.xyz.com** has the following configuration:

ns.xyz.com	10.1.1.10
y.xyz.com	10.1.1.1
nat-ns.xyz.com	196.168.1.254

NAT Router A Configuration

NAT Router A must be configured so that NAT is enabled on the WAN link to the Internet, with 2 rules defined for address translation.

Configuring the PPP Port:

The PPP port is configured so that NAT is enabled and operates according to **RULE #1** and **RULE #2**.

SE/PORT/#/ PPP example: with NAT enabled

```
NATROUTERA>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (CSL/1/2/3,def:1) ? 1
PORT #1> Protocol (def:PVCR) ? PPP
...
PORT #1> IP address (def:000.000.000.000) ? 195.16.1.1
PORT #1> Subnet mask (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT #1> IP RIP (def:V1) ?
PORT #1> IP RIP TX/RX (def:DUPLEX) ? RX ONLY
PORT #1> NAT enable (def:NO) ? YES
PORT #1> NAT rule (def:) ? 1,2
PORT #1> NAT side (def:INTERNAL) ? EXTERNAL
PORT #1> Silent (def:SEND REQUEST) ?
PORT #1> LCP timeout (s)  (1-255,def:3) ?
PORT #1> LCP retries, 255 = forever  (0-255,def:255) ?
PORT #1> Negotiate MRU (def:NO) ?
PORT #1> Use MRU proposed by peer (def:NO) ?
PORT #1> Request Magic Number (def:YES) ?
PORT #1> Accept Magic Number Request (def:YES) ?
PORT #1> Accept Addresses Old Negotiation (def:NO) ?
PORT #1> Request IP-Address (def:NO) ?
PORT #1> Accept IP-Address Request (def:NO) ?
PORT #1> Remote IP-Address (def:000.000.000.000) ?
PORT #1> PPP dial index (def:NONE) ?
```

Configuring the Rules:

RULE #1 uses the **STATIC** translation type, and is defined to reach the DNS Server **ns.abc.com** (**10.1.1.10**).

SE/IP/NAT/ RULE example: for Rule #1

```
NATROUTERA>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
```

```
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
IP NAT RULE #1> Translation type (def:DYNAMIC) ? STATIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ?
10.1.1.10
IP NAT RULE #1> IP source address max (def:000.000.000.000) ?
10.1.1.10
IP NAT RULE #1> NAT IP source network (def:000.000.000.000) ?
195.16.1.254
IP NAT RULE #1> NAT IP source network mask (0-32,def:0) ? 8
{255.000.000.000}
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network mask (0-32,def:0) ?
{000.000.000.000}
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

RULE #2 uses the **DYNAMIC** translation type, and is configured as follows:

SE/IP/NAT/ RULE example: for Rule #2

```
NATROUTERA>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 2
IP NAT RULE #2> Translation type (def:STATIC) ? DYNAMIC
IP NAT RULE #2> IP source address min (def:000.000.000.000) ?
10.0.0.0
IP NAT RULE #2> IP source address max (def:000.000.000.000) ?
10.255.255.255
IP NAT RULE #2> NAT IP source address min (def:000.000.000.000) ?
195.16.1.2
IP NAT RULE #2> NAT IP source address max (def:000.000.000.000) ?
195.16.1.253
IP NAT RULE #2> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #2> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #2> NAT IP destination address min
(def:000.000.000.000) ?
IP NAT RULE #2> NAT IP destination address max
(def:000.000.000.000) ?
IP NAT RULE #2> Pool size (0-2500,def:100) ?
```

NAT Router B Configuration

NAT Router B must be configured so that NAT is enabled on the WAN link to the Internet, with 2 rules defined for address translation.

Configuring the PPP Port:

The PPP port is configured so that NAT is enabled, and operates according to **RULE #1** and

RULE #2.

SE/PORT/#/ PPP example: with NAT enabled

```
NATROUTERB>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (CSL/1/2/3,def:1) ? 1
PORT #1> Protocol (def:PVCR) ? PPP
...
PORT #1> IP address (def:000.000.000.000) ? 196.168.1.1
PORT #1> Subnet mask (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT #1> IP RIP (def:V1) ?
PORT #1> IP RIP TX/RX (def:DUPLEX) ? RX ONLY
PORT #1> NAT enable (def:NO) ? YES
PORT #1> NAT rule (def:) ? 1,2
PORT #1> NAT side (def:INTERNAL) ? EXTERNAL
PORT #1> Silent (def:SEND REQUEST) ?
PORT #1> LCP timeout (s)  (1-255,def:3) ?
PORT #1> LCP retries, 255 = forever  (0-255,def:255) ?
PORT #1> Negotiate MRU (def:NO) ?
PORT #1> Use MRU proposed by peer (def:NO) ?
PORT #1> Request Magic Number (def:YES) ?
PORT #1> Accept Magic Number Request (def:YES) ?
PORT #1> Accept Addresses Old Negotiation (def:NO) ?
PORT #1> Request IP-Address (def:NO) ?
PORT #1> Accept IP-Address Request (def:NO) ?
PORT #1> Remote IP-Address (def:000.000.000.000) ?
PORT #1> PPP dial index (def:NONE) ?
```

Configuring the Rules:

RULE #1 uses the **STATIC** translation type, and is defined to reach the DNS Server **ns.xyz.com** (**10.1.1.10**):

SE/IP/NAT/ RULE example: for Rule #1

```
NATROUTERB>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 1
IP NAT RULE #1> Translation type (def:DYNAMIC) ? STATIC
IP NAT RULE #1> IP source address min (def:000.000.000.000) ?
10.1.1.10
IP NAT RULE #1> IP source address max (def:000.000.000.000) ?
10.1.1.10
IP NAT RULE #1> NAT IP source network (def:000.000.000.000) ?
196.168.1.254
```

```
IP NAT RULE #1> NAT IP source network mask (0-32,def:0) ? 8
{255.000.000.000}
IP NAT RULE #1> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #1> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network (def:000.000.000.000) ?
IP NAT RULE #1> NAT IP destination network mask (0-32,def:0) ?
{000.000.000.000}
IP NAT RULE #1> Pool size (0-2500,def:100) ?
```

RULE #2 uses the **DYNAMIC** translation type, and is configured as follows:

SE/IP/NAT/RULE example: for Rule #2

```
NATROUTERB>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? NAT
Item (RULE/PORT/TIMEOUT/SERVICE,def:RULE) ? RULE
IP NAT rule number (1/2/3/4/5/6/7/8/9/10,def:1) ? 2
IP NAT RULE #2> Translation type (def:STATIC) ? DYNAMIC
IP NAT RULE #2> IP source address min (def:000.000.000.000) ?
10.0.0.0
IP NAT RULE #2> IP source address max (def:000.000.000.000) ?
10.255.255.255
IP NAT RULE #2> NAT IP source address min (def:000.000.000.000) ?
196.168.1.2
IP NAT RULE #2> NAT IP source address max (def:000.000.000.000) ?
196.168.1.253
IP NAT RULE #2> IP destination address min (def:000.000.000.000) ?
IP NAT RULE #2> IP destination address max (def:000.000.000.000) ?
IP NAT RULE #2> NAT IP destination address min
(def:000.000.000.000) ?
IP NAT RULE #2> NAT IP destination address max
(def:000.000.000.000) ?
IP NAT RULE #2> Pool size (0-2500,def:100) ?
```

7

OSPF Network Support

7.1 About the Open Shortest Path First

The Open Shortest Path First (OSPF) protocol is a link-state routing protocol designed for a single *Autonomous System* (AS), which is a group of routers that exchange routing information via a common routing protocol.

The OSPF protocol runs directly over IP. It is considered an *Interior Gateway Protocol* (IGP), since it distributes routing information between routers that belong to the AS. Routing data may also be derived externally to the AS, that is, via an *Exterior Gateway Protocol* (EGP). This data is passed transparently across the AS, and is kept separate from the OSPF protocol's link state data.

OSPF was designed for the TCP/IP Internet environment and supports:

- IP subnetting,
- Routing based on the IP Type of Service (TOS),
- Authentication of routing updates,
- Tagging of externally-derived routing information.

OSPF is a dynamic routing protocol, in that it quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence. This period of convergence is short and involves a minimum of routing traffic.

7.1.1 How OSPF Routing Works

Each router running the OSPF protocol in the AS is assigned a unique *router ID*. The connection between a router and one of its attached networks is called an *interface* or *link*. An interface has state information associated with it, which is obtained from the underlying lower-level protocols and the routing protocol itself.

OSPF routes IP packets based on the destination IP address and the IP Type of Service (TOS). Both of these are found in the IP packet header, and no further encapsulation is required. Each interface to a network is assigned a single IP address and mask (unless the network is an unnumbered point-to-point network). Each network in the OSPF topology likewise has an IP address and associated network mask representing the number of nodes on the network. Details on how OSPF supports IP subnetting are given in *IP and Variable Length Subnetting*, later in this document.

Each router in the AS maintains a database that portrays the topology of the Autonomous System. For a particular router, each database entry describes the router's local state, such as the available interfaces and neighbors. Each router broadcasts its local state information throughout the AS in a *link state advertisement*. When taken together, the link state advertisements of all routers and networks in the AS comprise the OSPF *topological database*, which each participating router uses to construct an identical picture of the current AS topology.

From its database information, each router designs a tree of shortest paths to each destination in the AS, using itself as the root. With OSPF, separate routes can also be calculated for each Type of Service that is used.

A configurable *cost* is associated with the output side of each router interface. The lower the cost, the more likely the interface will be used to forward data traffic. Costs are also associated with externally derived routing data such as EGP, static routes and default routes. The total cost of a particular route is determined from the cost of all hops along the route.

NOTE: Links leading from networks to routers always have cost 0.

7.1.2 Types of Networks Supported

In OSPF terminology, the term *network* refers to an IP network, subnet or supernet. One physical network can have more than one IP network/subnet number, in which case each network number is considered a separate network (with the exception of point-to-point physical networks, as noted below).

OSPF supports the following types of physical networks:

- **Point-to-point Network**: A network that joins a single pair of routers. A point-to-point network is considered a single network no matter how many IP network/subnet numbers are assigned to it. Example: a serial line running at 64 Kbps.
- **Broadcast Network**: A network that supports more than two attached routers (multi-access), and that can address a single message to all attached routers (broadcast). Example: an Ethernet or Token-Ring LAN.
- **Non-broadcast Network**: A multi-access network that does not have broadcast capability. Example: a Frame Relay or X.25 network.

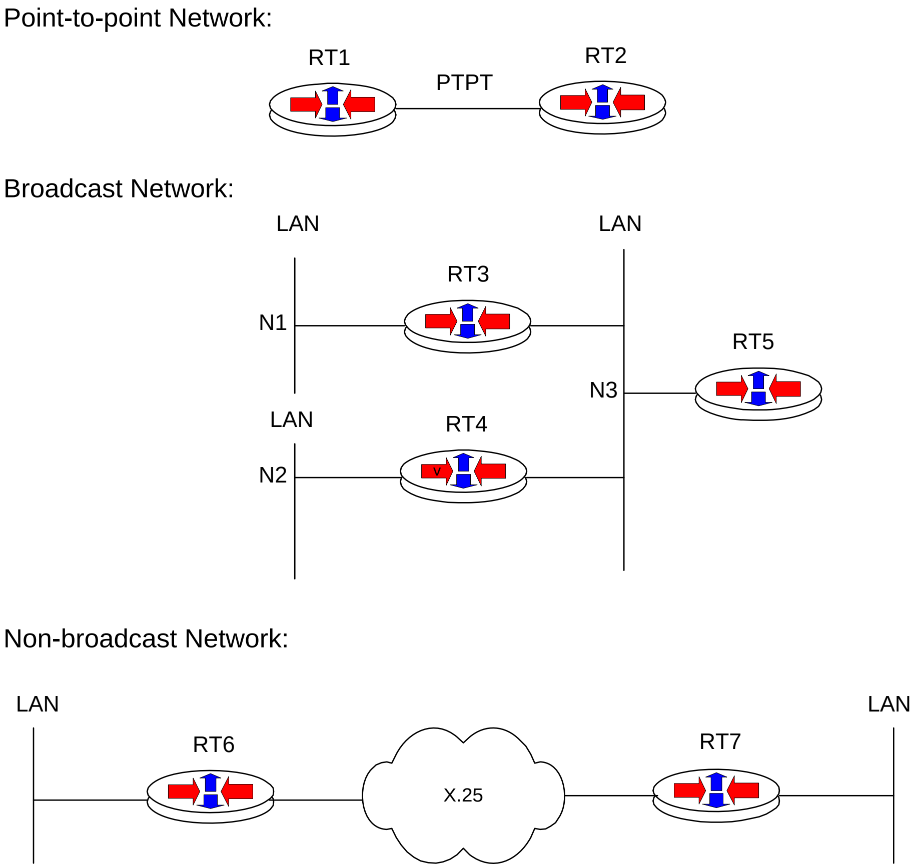
These network types are depicted in Physical Networks Supported by OSPF on the next page.

Each pair of routers on a multi-access network (both broadcast and non-broadcast) can communicate directly. A multi-drop network is thus considered to be a type of point-to-point network, rather than a multi-access network.

Multi-access networks can be *transit* or *stub* networks, depending on their function:

- **Transit network**: routes traffic that is neither locally originated nor locally destined.
- **Stub network**: has a local endpoint, and is not bidirectionally connected in the OSPF database topology.

For example, if only one router is attached to a multi-access network, that network will appear in the topology as a stub connection. Hosts attached directly to routers (referred to as *host routes*) appear in the topology as stub networks. Interface addresses are also modelled as stub routes, with each router having a stub connection to the other router’s interface address.



Point-to-point Network:

Broadcast Network:

Non-broadcast Network:

Figure 7-1: Physical Networks Supported by OSPF

7.2 Neighboring, Adjacent and Designated Routers

Two routers that have interfaces to the same network are considered *neighboring routers*. On multi-access networks, neighboring routers are dynamically discovered using the OSPF Hello Protocol. The Hello Protocol takes advantage of the broadcast and multicast capabilities of the networks it serves. For non-broadcast networks some configuration information is also required, and OSPF packets that would normally be multicast must be sent to each neighboring router in turn.

Some pairs of neighboring routers are selected for the purpose of exchanging routing information. This special relationship is called an *adjacency*. Not all neighboring routers are adjacent, which reduces the amount of routing traffic in the AS.

Adjacencies control the distribution of routing protocol packets and updates to the database topology. OSPF routing packets (with the exception of Hellos) are sent and received only on adjacencies. This means that all OSPF protocol packets travel a single IP hop, unless the adjacency is created from a virtual link (see "The AS Backbone and Virtual Links" on page 7-7).

Each multi-access network has a *designated router* that is elected by the Hello Protocol. The designated router has special administrative functions such as generating all link state advertisements for the multi-access network. This reduces the number of adjacencies required on the network, thereby reducing the amount of routing protocol traffic and the size of the topological database.

7.2.1 Splitting the AS into Areas

OSPF allows a set of contiguous networks and hosts within the AS to be grouped together as an *area*. An area also includes all routers that interface with any of the included networks. The address specifications of an OSPF area are described later in "The Address Range of an Area" on page 7-9.

The area concept is similar to that of the IP subnetted network. However, the topology of an area is hidden from the rest of the AS, which permits a significant reduction in routing traffic. Each area runs a separate copy of the basic link-state routing algorithm and maintains its own topological database. Routers inside a given area know nothing of the detailed topology outside of the area.

The area data structure contains all the information used to run the basic OSPF routing algorithm. A network belongs to a single area, and a router interface connects to a single area. Each router adjacency also belongs to a single area.

The area topological (or link state) database consists of the collection of router links, network links and summary link advertisements that have originated from the area's routers. This information is flooded throughout a single area only. The list of AS external link advertisements is also considered to be part of each area's topological database.

When an AS is split into areas, routers in different areas no longer have identical topological databases. A router constructs a separate topological database for each area it is connected to. (Routers connected to multiple areas are called *area border routers*; see "Types of OSPF Routers" on page 7-8.) Two routers belonging to the same area, however, will have identical topological databases for that area.

Routing in the AS areas may be of two types: *intra-area* and *inter-area*:

- **Intra-area routing**: used when the source and destination of a packet reside in the same area.
- **Inter-area routing**: used when the source and destination are in different areas. This type of routing requires a common backbone (see next section).

For intra-area routing, the packet is routed solely on information obtained within the area. No routing information obtained from outside the area can be used. This protects against bad routing information.

7.3 The AS Backbone and Virtual Links

The *backbone* of an AS consists of all networks that are not contained in any area, along with their attached routers and all area border routers. The backbone components must be contiguous.

The backbone is responsible for distributing routing information between areas (inter-area routing). Its properties are similar to those of an area, in that the backbone topology is invisible to each of the areas and, conversely, the backbone knows nothing of the area topologies. For this reason the backbone is represented by an area data structure.

For inter-area routing, the path that the packet travels can be divided into three contiguous pieces: an intra-area path from the source device to an area border router, a backbone path between the source and destination areas, and then another intra-area path from destination area border router to the destination device. The OSPF algorithm always uses the set of paths that have the smallest cost.

If new areas are defined in the AS such that the backbone is no longer contiguous, one or more *virtual links* must be configured to restore backbone connectivity. Virtual links belong to the backbone. They can be defined between any two backbone routers that interface with a common non-backbone area.

Although a virtual link has an associated IP interface address, the OSPF protocol treats two routers joined by a virtual link as if they were connected by an unnumbered point-to-point network. The cost of this connection is determined from the intra-area distances between the two routers. The routing protocol traffic that flows along a virtual link uses intra-area routing only.

7.4 Types of OSPF Routers

Routers participating in AS areas can be classed into four overlapping categories:

- **Internal Router**: a router that is directly connected to networks belonging to a single area only. Routers that interface only with the backbone also belong in this category. These routers run a single copy of the OSPF routing algorithm.
- **Area Border Router**: a router that attaches to multiple areas. Area border routers run multiple copies of the basic algorithm (one copy for each attached area and an additional copy for the backbone). Area border routers condense the topological information of their attached areas for distribution to the backbone. The backbone in turn distributes this information to the other areas of the AS.
- **Backbone Router**: a router that has an interface to the backbone. By definition this includes all area border routers, although some backbone routers are not area border routers. Note that routers with all interfaces connected to the backbone are considered internal routers rather than backbone routers.
- **AS Boundary Router**: a router that exchanges routing information with routers belonging to other Autonomous Systems. An AS boundary router uses AS external routes that are advertised throughout the Autonomous System. Every router in the AS knows the path to each AS boundary router.

  The definition of an AS boundary router cuts across the other router classifications. That is, an AS boundary router may also be an internal or area border router, and may or may not participate in the backbone. It can be configured to prevent the forwarding of RIP entries and static routes (see "OSPF AS boundary router" on page 9-2).

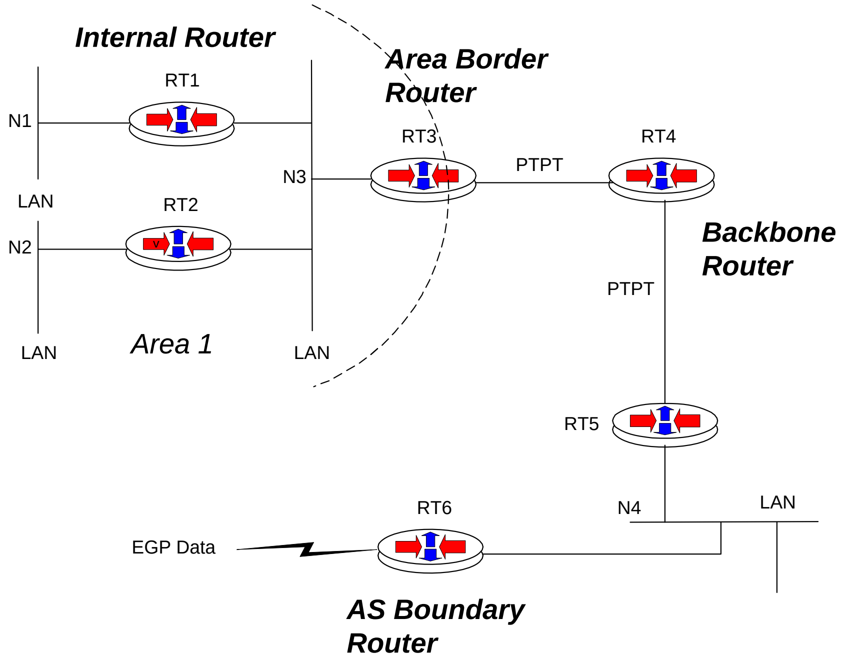


Figure 7-2: OSPF Router Types

7.5 Variable-Length Subnetting

OSPF attaches an IP address mask to each advertised route. The mask indicates the range of addresses included in the route. For example, a summary advertisement for the destination 128.185.0.0 with a mask of 0xffff0000 (255.255.0.0) actually describes a single route to all destinations from 128.185.0.0 to 128.185.255.255. Similarly, hosts attached directly to routers (host routes) are always advertised with a mask of 0xffffffff (255.255.255.255), indicating only one destination.

With OSPF, the configuration of IP subnets is flexible. A single IP network number can have two or more subnets of different sizes, in other words, subnets with different masks. This is commonly referred to as *variable-length subnetting*.

There are many ways to implement variable-length subnetting, since a single IP class A, B, or C network number can be divided into many subnets of various sizes. However, when routing a packet, the longest (most specific) match is chosen. In other words, a packet is always forwarded to the network that is the best match for its destination. It is important, therefore, to assign subnet masks so that the best match for any IP destination is unambiguous. For example, a default route with destination 0.0.0.0 and mask 0x00000000 is always a match for every IP destination. Yet this route is always less specific than any other match, and will seldom be used.

7.6 The Address Range of an Area

The OSPF area concept is modelled after an IP subnetted network. Earlier we defined an area to be a set of contiguous networks and hosts within the AS. In fact, an OSPF area is a list of *address ranges*, where each range is defined as an [address,mask] pair. Just as a subnetted network is composed of many separate subnets, a single address range can contain many separate networks. Area border routers summarize the area contents for distribution to the backbone by advertising a single route for each address range. The cost of the route is the minimum cost to any of the networks falling within the specified range.

For example, an IP subnetted network can be configured as a single OSPF area. This area would be defined as a single address range: a class A, B, or C network number along with its natural IP mask. Inside the area, any number of variable sized subnets could be defined. An area border router distributes a single route for the entire subnetted network. This advertisement is external to the area, hiding the fact that the network is subnetted at all. The cost of this route is the minimum of the set of costs to the component subnets.

7.7 TOS-based Routing

The OSPF algorithm is able calculate a separate set of routes for each IP Type of Service. This means that for any single destination there may be multiple routing table entries, one for each IP TOS. Separate interface costs can also be configured for each TOS (the cost for TOS 0 is always specified). TOS values are represented in OSPF exactly as they appear in the IP packet header.

NetPerformer routes all packets on the TOS 0 path, eliminating the need to calculate non-zero TOS paths. This conserves both routing table space and processing resources.

The NetPerformer sends all OSPF routing protocol packets with the IP Type of Service (TOS) field set to 0. It does not change the value of the TOS field on packets received or forwarded, thereby preserving TOS distinctions for other routers in the AS. NetPerformer units can thus be mixed with routers that perform TOS-based routing, with no loss of routing information.

7.8 Authentication

All OSPF protocol exchanges are authenticated. This means that only authorized routers can route packets over the AS. Each area can be configured separately as to its authentication scheme, which permits defining some areas with a higher level of authentication than others.

The OSPF packet header includes an authentication type field and 64 bits of data for use by the authentication scheme. Three authentication types are available:

- **Type 0: No authentication.** Routing exchanges in the area are not authenticated. The authentication field in the OSPF header can contain anything, and is not examined on packet reception.
- **Type 1: Simple password.** The 64-bit authentication field is configured on a per-network basis. All packets sent on the network must carry this value in the OSPF header. A router must be configured with the password of its attached network before it can participate in the routing domain.

  With simple password authentication the password is not encoded, and can easily be determined by taking a traffic capture on the port.
- **Type 2: Cryptographic authentication.** A shared secret key is configured in all routers that are attached to a common network. This key is used to generate a message digest that is appended to the end of each OSPF packet. The secret key itself is never sent over the network, which prevents intruders from decoding the password from a traffic capture.

  The NetPerformer uses the MD5 algorithm to encrypt the shared secret key.

Each NetPerformer interface is configured individually for OSPF authentication, including LAN and WAN ports, Frame Relay PVCs and OSPF virtual links. A complete description of the parameters required appears in this document on “OSPF Authentication type” on page 8-15 (LAN configuration) and “Authentication type” on page 9-20 (VLINK configuration).

7.9 IP RIP/OSPF Integration

The NetPerformer can integrate communication between the RIP (Routing Information Protocol) and OSPF (Open Shortest Path First) protocols. This permits creation of a RIP/OSPF network, typically composed of smaller systems using the RIP protocol interconnected with a larger system using the OSPF protocol.

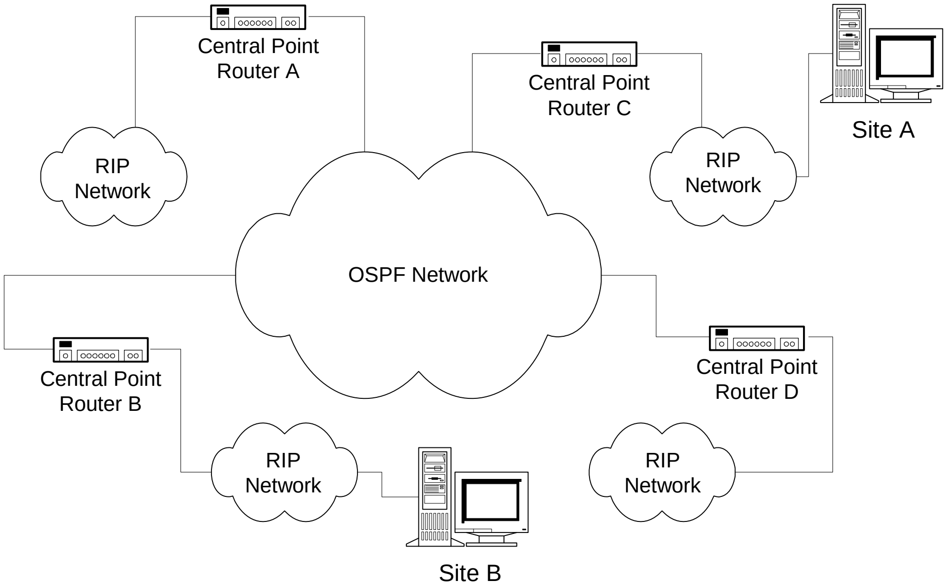


Figure 7-3: Integration of RIP and OSPF in a Large Network

In a combined RIP/OSPF network, one NetPerformer unit is designated as the Central Point Router:

- This unit serves as the connection point between the RIP and OSPF systems
- It has at least one port that uses RIP frames and one port that uses OSPF
- It is configured to forward the entries learned by the other side, thereby making communication between the RIP and OSPF systems possible.

The NetPerformer routing table can store either RIP or OSPF information, as it is independent of the protocol responsible for maintaining it.

- Entries that are learned by either RIP or OSPF do not overwrite entries learned by the other algorithm.
- The metric cost of each route is converted from RIP to OSPF values using a configurable parameter.

The parameters required to set up an integrated RIP/OSPF network are:

- *OSPF AS boundary router* and, if set to **YES**, the following parameters:
  - *RIP to OSPF metric conversion cost*
  - *OSPF AS forwards RIP entries*

- *OSPF AS forwards STATIC entries*

- *RIP AS boundary router.*

For information on configuring an integrated RIP/OSPF network, refer to the description of these parameters on "OSPF AS boundary router" on page 9-2.

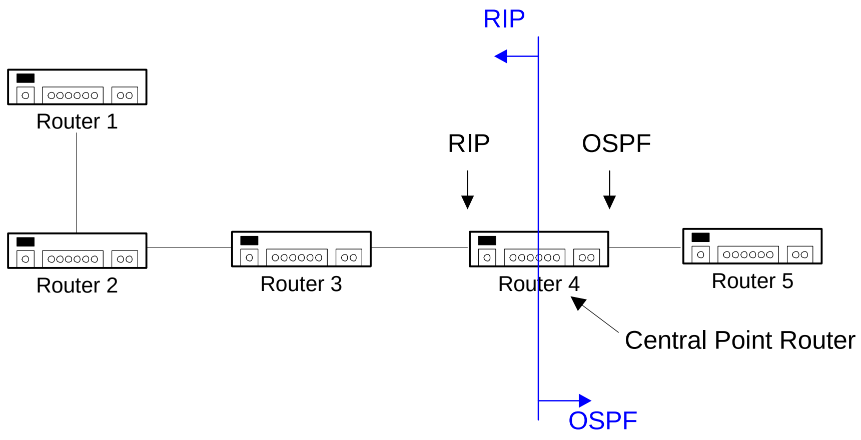


Figure 7-4: Central Point Router in an Integrated RIP/OSPF Network

7.10 OSPF Configuration

The OSPF protocol can be enabled on the following NetPerformer interfaces:

- **LAN port**: Ethernet (ETH) LAN port
- **WAN link**: Serial port configured with the PVCR protocol
- **PVC link**: PVC configured in PVCR or RFC1490 mode.

Several areas of the NetPerformer configuration need to be adjusted to support OSPF. To configure all required parameters, use the following procedure:

1. Enable OSPF on the ports and PVCs that will route OSPF traffic (see next section)
2. Define the OSPF interface characteristics on the ports/PVCs (see "Defining OSPF Interface Characteristics" on page 7-14)
3. Define the OSPF global characteristics (see "Defining OSPF Global Characteristics" on page 7-17)
4. Define the OSPF area characteristics (see "Defining OSPF Area Characteristics" on page 7-17)
5. Define the address range of all areas (see "Defining the Range of all Areas" on page 7-18)
6. Define all required OSPF virtual links (see "Defining OSPF Virtual Links" on page 7-19)

NOTE: Virtual links are required only if the global *Auto virtual link* parameter is set to **NO**.

7.10.1 Enabling OSPF

OSPF must be enabled on all ports and PVCs that participate in OSPF routing. It is preferable to disable IP RIP on these ports and PVCs, although this is not mandatory. If both IP RIP and OSPF data are received at a NetPerformer interface, the OSPF data will be given higher priority.

As mentioned in the previous section, a NetPerformer LAN port, WAN link or PVC link can serve as an OSPF interface. The required parameters for an Ethernet or Token-Ring LAN port are listed under the Setup Slot menu (SNMP *iflan* category). Those for a WAN link (PVCR protocol) can be accessed by entering the port number at the Setup Ports menu prompt (SNMP *ifwan* category). Use the Setup PVC menu to enable OSPF on a PVC in either PVCR or RFC1490 mode.

If you want to disable IP RIP, enter carriage returns at the NetPerformer console until you reach the IP RIP parameter. If you are using SNMP, select the appropriate variable name: *iflanIpRip* for a LAN port, *ifwanIpRip* for a WAN port, or *pvcIpRip* for a PVC. Disable IP RIP routing on the port/PVC by entering the value DISABLE.

To enable OSPF, use the **OSPF** parameter on the console listing for the link interface

(Ethernet port or WAN connection):

SE/PORT/ETH example: enabling OSPF

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? PORT
Port number (ETH1/ETH2/CSL/1,def:ETH1) ?
PORT ETH 1> Protocol (def:ETH AUTO) ?
PORT ETH 1> LAN speed (mbps) (def:AUTO) ?
PORT ETH 1> MAC address (def:000000000000) ?
PORT ETH 1> DHCP (def:DISABLE) ?
PORT ETH 1> IP address 1 (def:000.000.000.000) ?
PORT ETH 1> Subnet mask 1 (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT ETH 1> IP address 2 (def:000.000.000.000) ?
PORT ETH 1> Subnet mask 2 (number of bits) (0-32,def:8) ?
{255.000.000.000}
PORT ETH 1> Frame size (128-8192,def:1500) ?
PORT ETH 1> IP RIP (def:V1) ?
PORT ETH 1> IP RIP TX/RX (def:DUPLEX) ?
PORT ETH 1> OSPF (def:DISABLE) ? ENABLE
...
```

NOTE: Details on the OSPF parameters that are defined on the link interface (Ethernet port or WAN connection) are detailed with the Ethernet parameters, starting from "OSPF" on page 8-12.

7.10.2 Defining OSPF Interface Characteristics

An OSPF interface is the connection between a router and a network, and belongs to the area that contains the attached network. There is a single OSPF interface structure for each attached network, so each interface structure has at most one IP interface address. All routing protocol packets originated by the NetPerformer over an OSPF interface are labelled with the interface's Area ID.

All OSPF interface characteristics are configured from the Setup Slot, Setup Port and Setup PVC menus. On the console, the required parameters are displayed after the OSPF Enable parameter (see previous section). If you are using SNMP, you can access each variable directly. Select the *iflan-* variable when configuring a LAN port, the *ifwan-* variable for a WAN port, and the *pvc-* variable for a PVC.

The following parameters define an OSPF interface on a NetPerformer. Note that some of these parameters actually define the attached network. The value of these parameters must be the same for all routers connected to this network. This requirement is noted where applicable.

SE/PORT/ETH example: OSPF interface parameters

```
...
PORT ETH 1> OSPF (def:DISABLE) ? ENABLE
PORT ETH 1> OSPF Area ID (def:000.000.000.000) ?
PORT ETH 1> OSPF Router priority (0-255,def:1) ?
PORT ETH 1> OSPF Transit delay (1-360,def:1) ?
PORT ETH 1> OSPF Retransmit interval (1-360,def:5) ?
PORT ETH 1> OSPF Hello interval (1-360,def:10) ?
PORT ETH 1> OSPF Dead interval (1-2000,def:40) ?
PORT ETH 1> OSPF Authentication type (def:NONE) ?
PORT ETH 1> OSPF Metric cost (1-65534,def:10) ?
...
```

Refer to “OSPF” on page 8-12 for details on these parameters.

7.10.3 The Setup OSPF Menu

Console	SNMP	Text-based Config
SE/IP/OSPF	*ospf* (category)	*[ospf]* (heading)

The Setup OSPF menu lets you configure all parameters required to define the OSPF global, area, range and virtual link characteristics. For SNMP, these configuration variables are found under the *ospf* category. At the console, first enter **SE** followed by **IP** to reach the Setup IP menu, then enter **OSPF** at the second **Item** prompt to reach the Setup OSPF menu. A third **Item** prompt is displayed to select one of the OSPF submenus, as in the following example.

SE/IP/OSPF example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? OSPF
Item (GLOBAL/AREA/RANGE/VLINK,def:GLOBAL) ?
...
```

The four OSPF submenu options are the following:

NOTE: The SNMP equivalents given below are the group name prefixes of the included variables. Each variable is accessed directly from the MIB. For the full variable names refer to the individual parameter descriptions later in this chapter.

Console	SNMP	Text-based Config
Item ? **GLOBAL**	ospfGlobal (prefix)	[ospfGlobal]

Setup OSPF Global: Enter GLOBAL on the console command line after the OSPF menu **Item** prompt. For SNMP, select the desired variable in the *ospf* category. All configurable OSPF Global parameters are addressed in “Defining OSPF Interface Characteristics” on page 7-14.

Console	SNMP	Text-based Config
Item ? **AREA**	ospfArea (prefix)	[ospfArea]

Setup OSPF Area: Enter AREA on the console command line after the OSPF menu **Item** prompt. For SNMP, select the desired variable in the *ospf* category. All configurable OSPF Area parameters are described later in “Defining OSPF Area Characteristics” on page 7-17.

Console	SNMP	Text-based Config
Item ? **RANGE**	ospfRange (prefix)	[ospfRange]

Setup OSPF Range: Enter **RANGE** on the console command line after the OSPF menu **Item** prompt. For SNMP, select the desired variable in the *ospf* category. We will examine all configurable OSPF Range parameters later in “Defining the Range of all Areas” on page 7-18.

Console	SNMP	Text-based Config
Item ? **VLINK**	ospfVLink (prefix)	[ospfLink]

Setup OSPF Virtual Link: Enter **VLINK** on the console command line after the OSPF menu **Item** prompt. For SNMP, select the desired variable in the *ospf* category. Turn to “Defining OSPF Virtual Links” on page 7-19 for a discussion of all configurable OSPF Virtual Link parameters.

7.10.4 Defining OSPF Global Characteristics

All OSPF Global characteristics are configured from the Setup OSPF Global submenu. For console access, select the **GLOBAL** option of the Setup OSPF menu, as described in the previous section. The SNMP variables can be accessed directly by selecting the desired variable name. The Setup OSPF Global submenu includes the following parameters:

SE/IP/OSPF/ GLOBAL example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? OSPF
Item (GLOBAL/AREA/RANGE/VLINK,def:GLOBAL) ?
OSPF GLOBAL> Router ID (def:000.000.000.000) ?
OSPF GLOBAL> Auto virtual link (def:NO) ? Y
OSPF GLOBAL> Global area ID (def:000.000.000.000) ?
```

The OSPF Global parameters are detailed in the section “SE/IP/OSPF/GLOBAL Parameters” on page 9-10.

7.10.5 Defining OSPF Area Characteristics

All OSPF Area characteristics are configured from the Setup OSPF Area submenu. For console access, select the AREA option of the Setup OSPF menu, as described in “The Setup OSPF Menu” on page 7-15. The SNMP variables can be accessed directly by selecting the desired variable name.

NOTE: The OSPF backbone has all the properties of an area and is represented by an area data structure. However, you do not need to configure an area for the backbone. The NetPerformer automatically assigns Area ID 000.000.000.000 to the backbone.

When you first access the Setup OSPF Area submenu on the console, select a specific Area entry (1 to 10) using the *OSPF Area entry number* parameter.

SE/IP/OSPF/ AREA example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? OSPF
```

```
Item (GLOBAL/AREA/RANGE/VLINK,def:GLOBAL) ? AREA
OSPF Area entry number (1-10,def:1) ?
OSPF AREA 1> Area ID (def:000.000.000.000) ?
OSPF AREA 1> Enable (def:ENABLE) ?
OSPF AREA 1> Authentication type (def:NONE) ?
OSPF AREA 1> Import AS extern (def:YES) ?
OSPF AREA 1> Stub metric (1-255,def:1) ?
```

These parameters are detailed in the section "SE/IP/OSPF/AREA Parameters" on page 9-12.

7.10.6 Defining the Range of all Areas

Ranges are address/mask pairs that let you group subnetted networks residing in the same area. With this feature, the NetPerformer (or other router) can generate a single network summary advertisement for the entire group, rather than one for each subnet in the area.

NOTE: Ranges are required only for those areas that connect to the backbone via an area border router.

All OSPF Range characteristics are configured from the Setup OSPF Range submenu. For console access, select the **RANGE** option of the Setup OSPF menu, as described earlier in "The Setup OSPF Menu" on page 7-15. The SNMP variables can be accessed directly by selecting the desired variable name.

When you first access the Setup OSPF Range submenu on the console, select a specific Range entry (1 to 10) with the *OSPF Range entry number* parameter.

SE/IP/OSPF/ RANGE example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:BOOTP) ? OSPF
Item (GLOBAL/AREA/RANGE/VLINK,def:AREA) ? RANGE
OSPF Range entry number (1-10,def:1) ?
OSPF RANGE 1> Range net (def:000.000.000.000) ?
OSPF RANGE 1> Range mask (def:000.000.000.000) ?
OSPF RANGE 1> Enable (def:ENABLE) ?
OSPF RANGE 1> Status (def:ADVERTISE) ?
OSPF RANGE 1> Add to area (def:000.000.000.000) ?
```

These parameters are detailed in the section "SE/IP/OSPF/RANGE Parameters" on page 9-15.

Address Range Configuration Example:

As an example, suppose that an IP subnetted network is defined as a single OSPF area. A single address range (Range Net and Range Mask) should be configured for that area.

- Set the Range Net value to the IP address of the subnetted network, and the Range Mask value to the natural class A, B or C address mask.
- Enable the range entry, and add it to the area to which the range belongs.
- The Status parameter should be set to **ADVERTISE**, to allow the advertisement of a single route external to the area. This external route would describe the entire subnetted network.

7.10.7 Defining OSPF Virtual Links

A virtual link can be configured for the following purposes:

- To restore backbone connectivity when the AS is divided into non-contiguous areas
- For backup support, in case of link failure between routers
- Where the virtual link would permit a shorter path to an area that advertises a good route to an external network

To allow for virtual link configuration, **the NetPerformer must be an area border router**. All virtual links are configured using the NetPerformer as one endpoint. Each virtual link must also be configured in the other endpoint router.

For each virtual link you must identify:

- The Router ID of the other endpoint. This must be a neighboring area border router.
- The virtual link’s transit area. This is the common area to which both endpoint routers are attached.
- Properties of the virtual link interface, including the authentication type used.

Define a virtual link wherever you find that the NetPerformer may need to route OSPF data to another backbone router via a common non-backbone area (the virtual link’s transit area). The virtual links you define will be brought up and down depending on the current shortest-path tree for the transit area.

The OSPF Virtual Link characteristics can be manually configured from the Setup OSPF Virtual Link submenu. For console access, select the VLINK option of the Setup OSPF menu, as described in “The Setup OSPF Menu” on page 7-15. The SNMP variables can be accessed directly by selecting the desired variable name.

NOTE: Auto Virtual Link, an OSPF Global parameter, allows the NetPerformer to automatically configure and enable virtual links as they are required. Refer to “Defining OSPF Global Characteristics” on page 7-17 for details.

When you first access the Setup OSPF Virtual Link submenu on the console, select a specific Virtual link entry (1 to 10) using the OSPF Virtual link entry number.

SE/IP/OSPF/ VLINK example

```
SDM-9230>SE
SETUP
Item (BRIDGE/CALLER ID/CLASS/CUSTOM/FILTER/GLOBAL/HUNT/IP/IPX/MAP/
PHONE/
PORT/PU/PPPOE/PPPUSER/PVC/REDUNDANCY/SCHEDULE/SLOT/USER/VLAN,
def:BRIDGE) ? IP
Item (BOOTP/DNS/FTP/GLOBAL/NAT/OSPF/RADIUS/SNMP/SOURCE-STATIC/
STATIC/
TELNET/TIMEP,def:OSPF) ? OSPF
Item (GLOBAL/AREA/RANGE/VLINK,def:RANGE) ? VLINK
OSPF Virtual link entry number (1-10,def:1) ?
OSPF VLINK 1> Transit area ID (def:000.000.000.000) ?
OSPF VLINK 1> Neighbor’s router ID (def:000.000.000.000) ?
OSPF VLINK 1> Enable (def:ENABLE) ?
OSPF VLINK 1> Transit delay (1-360,def:1) ?
OSPF VLINK 1> Retransmit interval (1-360,def:5) ?
OSPF VLINK 1> Hello interval (1-360,def:10) ?
OSPF VLINK 1> Dead interval (1-2000,def:60) ?
OSPF VLINK 1> Authentication type (def:NONE) ?
```

These parameters are detailed in the section “SE/IP/OSPF/VLINK Parameters” on page 9-17.

7.11 Routing Table for OSPF Routes

To view the status of all destinations reached through the OSPF protocol, use the IP Routing Table, which includes information on OSPF routes.

Console	SNMP
DR/IP	MIB II - *mgmt* - ipRouteTable

To access the IP Routing Table from the console, enter **DR** on the command line, and then select **IP**. For SNMP, open the Standard MIB (MIB II) and bring up the *ipRouteTable* table under the *mgmt* section.

DR/IP/ UNICAST/RIP example

```
SDM-9230>DR
DISPLAY ROUTING TABLE
Item (IP/IPX,def:IP) ?
Item (UNICAST/MULTICAST,def:UNICAST) ?
Item (RIP/MULTIHOMED,def:RIP) ?

The routing table has 8 entry(ies)

DESTINATION     VAL COST  INTF NEXT HOP        AGE   MASK
TYPE PROT

199.170.000.000 YES 120   LAN  200.160.001.003 687 s
255.255.000.000 RNGE OSPF
200.160.001.000 YES 0    LAN  200.160.001.001 0   s 255.255.255.000
NET  LOCAL
200.160.001.001 YES 0    LAN  200.160.001.001 0   s 255.255.255.255
HOST LOCAL
202.150.000.000 YES 300   2    201.150.003.002 688 s
255.255.000.000 RNGE OSPF
202.150.001.000 YES 200  2    201.150.003.002 0   s 255.255.255.000
NET  OSPF
202.150.001.001 YES 300   2    201.150.003.002 688 s
255.255.255.255 HOST OSPF
202.150.001.002 YES 200   2    201.150.003.002 688 s
255.255.255.255 HOST OSPF
220.150.000.000 YES 210   2    201.150.003.002 688 s
255.255.000.000 RNGE OSPF
```

Two columns of the IP routing table indicate the nature of the OSPF connections: **TYPE** and **PROT**.

7.11.1 TYPE

Console	SNMP
TYPE	MIB II - *mgmt* - ipRouteTable

This entry specifies the destination type. The following types may be displayed:

- **NET**: network,
- **SUB**: a subnet,
- **HOST**: a host,
- **DGTW**: a default gateway,
- **RNGE**: a range, that is, an entry that can be recognized by OSPF.

7.11.2 PROT

Console	SNMP
PROT	MIB II - *mgmt* - ipRouteTable

This entry specifies the type of routing used to reach the destination. The following protocols may be displayed:

- **LOCAL**: when the destination is the IP address of a NetPerformer interface,
- **OSPF**: when the destination is reached through OSPF routing,
- **RIP**: when the destination is reached through IP RIP routing,
- **STATIC**: when the destination is reached via a static route,
- **UNK**: unknown protocol, for example, an invalid static route.

7.12 Application Example

The figure below is an example of an OSPF network application, using NetPerformers as routers RT1 to RT6.

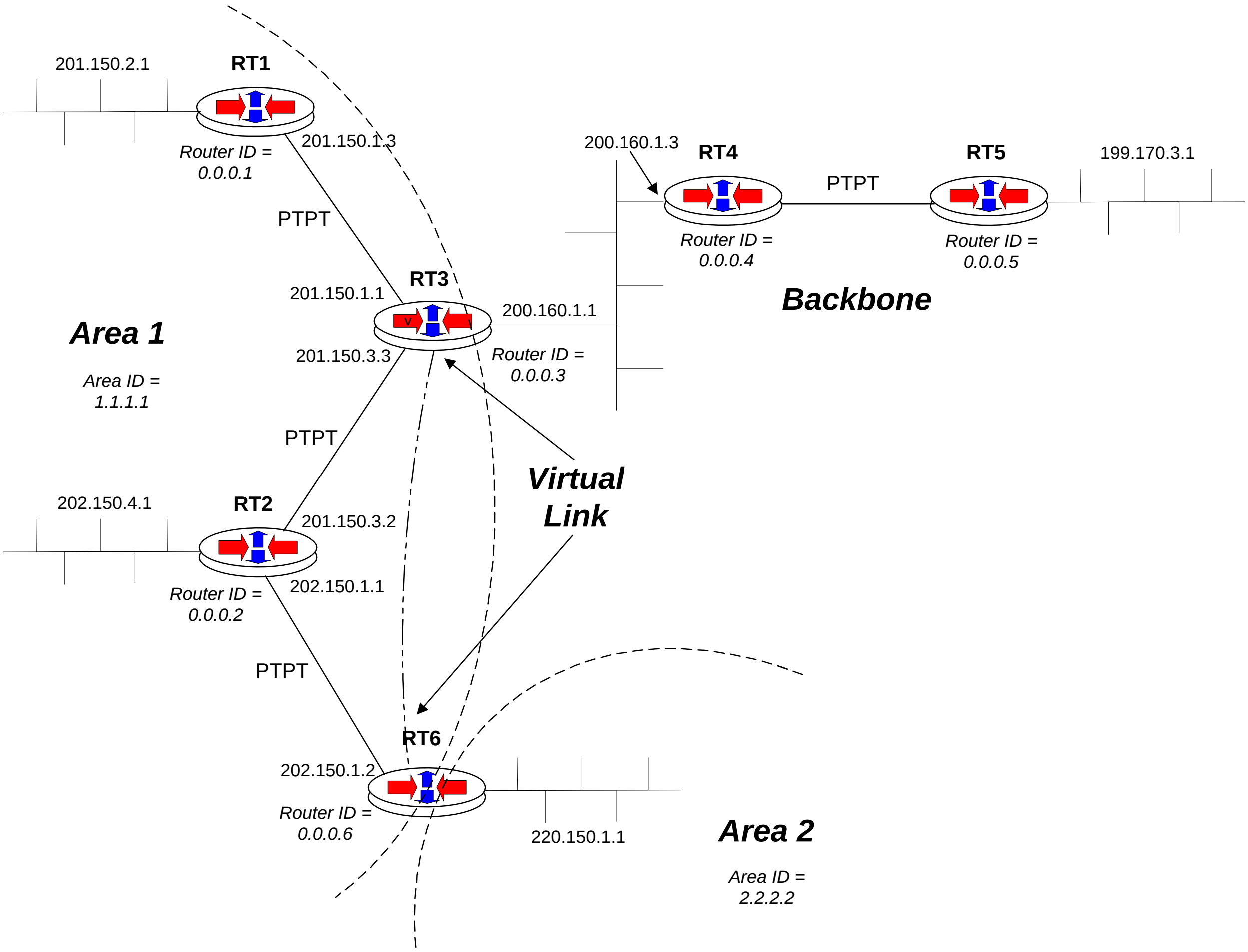


Figure 7-5: An Example OSPF Network

In this example, the following OSPF entities are defined:

- **Backbone**: includes routers RT3, RT4, RT5 and RT6 (through virtual link)
- **Area 1**: includes routers RT1, RT2, RT3 and RT6,
- **Area 2**: includes router RT6,
- **Ranges**: for area border routers RT3 and RT6,
- **Virtual Link**: between routers RT3 and RT6.

To demonstrate how the various ports and OSPF features should be configured in this network, we will list the required parameter values for:

- The area border router with a LAN connection to the backbone (RT3), in the next section
- The area border router with a virtual link connection to the backbone (RT6), on "Area Border Router with Virtual Link to the Backbone" on page 7-26
- The first backbone router that communicates with these area border routers (RT4), on "Backbone Router" on page 7-28.

7.12.1 Area Border Router with LAN Connection to the Backbone

In the example in Figure 7-5, RT3 is an area border router belonging to both Area 1 and the backbone. It has a LAN connection to the backbone, point-to-point connections to routers RT1 and RT2, and a virtual link connection to router RT6.

To configure the OSPF characteristics of router RT3:

The Ethernet port is used to access router RT4, and is part of the backbone. Define the Ethernet port with the following parameter values:

```
PORT ETH> IP address............................200.160.001.001
PORT ETH> Subnet mask (number of bits)..........24
{255.255.255.000}
...
PORT ETH> IP RIP................................DISABLE
PORT ETH> OSPF..................................ENABLE
PORT ETH> OSPF Area ID..........................000.000.000.000
PORT ETH> OSPF Router priority..................1
PORT ETH> OSPF Transit delay....................1
PORT ETH> OSPF Retransmit interval..............5
PORT ETH> OSPF Hello interval...................10
PORT ETH> OSPF Dead interval....................40
PORT ETH> OSPF Password.........................(8 characters)
PORT ETH> OSPF Metric cost......................10
```

Port 1 is used to access router RT1, and is part of Area 1. Define Port 1 with the following parameter values:

```
PORT 1> Protocol................................PVCR
...
PORT 1> IP address..............................201.150.001.001
PORT 1> Subnet mask (number of bits)............24
{255.255.255.000}
PORT 1> IP RIP..................................DISABLE
PORT 1> OSPF....................................ENABLE
PORT 1> OSPF Area ID............................001.001.001.001
PORT 1> OSPF Transit delay......................1
PORT 1> OSPF Retransmit interval................5
PORT 1> OSPF Hello interval.....................10
PORT 1> OSPF Dead interval......................40
PORT 1> OSPF Password...........................(8 characters)
PORT 1> OSPF Metric cost........................100
...
PORT 1> Remote unit name........................RT1
```

Port 2 is used to access router RT2, and is part of Area 1. Define Port 2 with the following parameter values:

```
PORT 2> Protocol................................PVCR
....
PORT 2> IP address..............................201.150.003.003
PORT 2> Subnet mask (number of bits)............24
{255.255.255.000}
PORT 2> IP RIP..................................DISABLE
PORT 2> OSPF....................................ENABLE
PORT 2> OSPF Area ID............................001.001.001.001
PORT 2> OSPF Transit delay......................1
PORT 2> OSPF Retransmit interval................5
PORT 2> OSPF Hello interval.....................10
PORT 2> OSPF Dead interval......................40
PORT 2> OSPF Password...........................(8 characters)
PORT 2> OSPF Metric cost........................100
...
PORT 2> Remote unit name........................RT2
```

Define the following OSPF Global characteristics for router RT3:

```
OSPF GLOBAL> Router ID..........................000.000.000.003
OSPF GLOBAL> Auto virtual link..................NO
```

Define Area 1 with the following OSPF Area parameter values (the backbone does not need to be configured):

```
OSPF AREA 1> Area ID............................001.001.001.001
OSPF AREA 1> Enable.............................ENABLE
OSPF AREA 1> Authentication type................NONE
OSPF AREA 1> Import AS extern...................YES
OSPF AREA 1> Stub metric........................1
```

Define the range of Area 1 with the following OSPF Range parameter values:

```
OSPF RANGE 1> Range net.........................201.150.000.000
OSPF RANGE 1> Range mask........................255.255.000.000
OSPF RANGE 1> Enable............................ENABLE
OSPF RANGE 1> Status............................ADVERTISE
OSPF RANGE 1> Add to area.......................001.001.001.001
```

Define the RT3 end of the virtual link with the following OSPF Virtual Link parameter values:

```
OSPF VLINK 1> Transit area ID...................001.001.001.001
OSPF VLINK 1> Neighbor's router ID..............000.000.000.006
OSPF VLINK 1> Enable............................ENABLE
OSPF VLINK 1> Transit delay.....................1
OSPF VLINK 1> Retransmit interval...............5
OSPF VLINK 1> Hello interval....................10
OSPF VLINK 1> Dead interval.....................60
OSPF VLINK 1> Password..........................(8 characters)
```

7.12.2 Area Border Router with Virtual Link to the Backbone

In the example in Figure 7-5, RT6 is an area border router belonging to both Area 1 and Area 2. It has a point-to-point connection to router RT2, and a virtual link connection to area border router RT3 (through which it can access the backbone).

To configure the OSPF characteristics of router RT6:

The Ethernet port can be used to access LAN-connected devices, and is part of Area 2. Define the Ethernet port with the following parameter values:

```
PORT ETH> IP address............................220.150.001.001
PORT ETH> Subnet mask (number of bits)..........24
{255.255.255.000}
...
PORT ETH> IP RIP................................DISABLE
PORT ETH> OSPF..................................ENABLE
PORT ETH> OSPF Area ID..........................002.002.002.002
PORT ETH> OSPF Router priority..................1
PORT ETH> OSPF Transit delay....................1
PORT ETH> OSPF Retransmit interval..............5
PORT ETH> OSPF Hello interval...................10
PORT ETH> OSPF Dead interval....................40
PORT ETH> OSPF Password.........................(8 characters)
PORT ETH> OSPF Metric cost......................10
```

Port 1 is used to access router RT2, and is part of Area 1. Define Port 1 with the following parameter values:

```
PORT 1> Protocol................................PVCR
...
PORT 1> IP address..............................202.150.001.002
PORT 1> Subnet mask (number of bits)............24
{255.255.255.000}
PORT 1> IP RIP..................................DISABLE
PORT 1> OSPF....................................ENABLE
PORT 1> OSPF Area ID............................001.001.001.001
PORT 1> OSPF Transit delay......................1
PORT 1> OSPF Retransmit interval................5
PORT 1> OSPF Hello interval.....................10
PORT 1> OSPF Dead interval......................40
PORT 1> OSPF Password...........................(8 characters)
PORT 1> OSPF Metric cost........................100
...
PORT 1> Remote unit name........................RT2
```

Define the following OSPF Global characteristics for router RT6:

```
       OSPF GLOBAL> Router ID..........................000.000.000.006
       OSPF GLOBAL> Auto virtual link..................NO
```

Define Area 1 with the following OSPF Area parameter values:

```
       OSPF AREA 1> Area ID............................001.001.001.001
       OSPF AREA 1> Enable.............................ENABLE
       OSPF AREA 1> Authentication type................NONE
       OSPF AREA 1> Import AS extern...................YES
       OSPF AREA 1> Stub metric........................1
```

Define Area 2 with the following OSPF Area parameter values:

```
       OSPF AREA 2> Area ID............................002.002.002.002
       OSPF AREA 2> Enable.............................ENABLE
       OSPF AREA 2> Authentication type................NONE
       OSPF AREA 2> Import AS extern...................YES
       OSPF AREA 2> Stub metric........................1
```

Define the range of Area 1 with the following OSPF Range parameter values for Range 1:

```
       OSPF RANGE 1> Range net.........................202.150.000.000
       OSPF RANGE 1> Range mask........................255.255.000.000
       OSPF RANGE 1> Enable............................ENABLE
       OSPF RANGE 1> Status............................ADVERTISE
       OSPF RANGE 1> Add to area.......................001.001.001.001
```

Also define a second range for Area 1, with the following OSPF Range parameter values:

```
       OSPF RANGE 2> Range net.........................201.150.000.000
       OSPF RANGE 2> Range mask........................255.255.000.000
       OSPF RANGE 2> Enable............................ENABLE
       OSPF RANGE 2> Status............................ADVERTISE
       OSPF RANGE 2> Add to area.......................001.001.001.001
```

Define the range of Area 2 with the following OSPF Range parameter values for Range 3:

```
       OSPF RANGE 3> Range net.........................220.150.000.000
       OSPF RANGE 3> Range mask........................255.255.000.000
       OSPF RANGE 3> Enable............................ENABLE
       OSPF RANGE 3> Status............................ADVERTISE
       OSPF RANGE 3> Add to area.......................002.002.002.002
```

Define the RT6 end of the virtual link with the following OSPF Virtual Link parameter values:

```
OSPF VLINK 1> Transit area ID...................001.001.001.001
       OSPF VLINK 1> Neighbor's router ID..............000.000.000.003
OSPF VLINK 1> Enable............................ENABLE
OSPF VLINK 1> Transit delay.....................1
OSPF VLINK 1> Retransmit interval...............5
OSPF VLINK 1> Hello interval....................10
OSPF VLINK 1> Dead interval.....................60
OSPF VLINK 1> Password..........................(8 characters)
```

7.12.3 Backbone Router

In the example in Figure 7-5, RT4 is a backbone router with a LAN connection to area border router RT3, and a point-to-point connection with backbone router RT5. It is able to communicate with router RT6 via the virtual link connection between routers RT3 and RT6.

To configure the OSPF characteristics of router RT4:

The Ethernet port is used to access router RT3, and is part of the backbone. Define the Ethernet port with the following parameter values:

```
PORT ETH> IP address............................200.160.001.003
PORT ETH> Subnet mask (number of bits)..........24
{255.255.255.000}
...
PORT ETH> IP RIP................................DISABLE
PORT ETH> OSPF..................................ENABLE
PORT ETH> OSPF Area ID..........................000.000.000.000
PORT ETH> OSPF Router priority..................10
PORT ETH> OSPF Transit delay....................1
PORT ETH> OSPF Retransmit interval..............5
PORT ETH> OSPF Hello interval...................10
PORT ETH> OSPF Dead interval....................40
PORT ETH> OSPF Password.........................(8 characters)
PORT ETH> OSPF Metric cost......................10
```

Port 2 is used to access router RT5, and is part of the backbone. Define Port 2 with the following parameter values:

```
PORT 2> Protocol................................PVCR
...
PORT 2> IP address..............................000.000.000.000
PORT 2> Subnet mask (number of bits)............24
{000.000.000.000}
PORT 2> IP RIP..................................DISABLE
PORT 2> OSPF....................................ENABLE
PORT 2> OSPF Area ID............................000.000.000.000
PORT 2> OSPF Transit delay......................1
PORT 2> OSPF Retransmit interval................5
PORT 2> OSPF Hello interval.....................10
PORT 2> OSPF Dead interval......................40
PORT 2> OSPF Password...........................(8 characters)
PORT 2> OSPF Metric cost........................100
...
PORT 2> Remote unit name........................RT5
```

Define the following OSPF Global characteristics for router RT4:

```
OSPF GLOBAL> Router ID..........................000.000.000.004
OSPF GLOBAL> Auto virtual link..................NO
```

Since RT4 is part of the backbone, no OSPF Area or Range characteristics need to be defined.

8

SE/PORT/ETH Configuration Parameters

NOTE: These parameters apply to LAN ports identified as **ETH**, **ETH1** or **ETH2** at the NetPerformer console.

8.1 Port number

Console	SNMP	Text-based Config
Port number	iflanIndex	[iflan#]

Select **ETH** to configure the Ethernet port. Select **ETH1** or **ETH2** on products that support more than one Ethernet port.

Values:	On products with one Ethernet port: ETH
	On products with two Ethernet ports: ETH1, ETH2
Default:	The last value selected

8.2 Protocol

Console	SNMP	Text-based Config
Protocol	iflanProtocol	[iflan#] Protocol

The operating protocol for the Ethernet port. Two Ethernet frame formats are recognized: IEEE 802.3 which is based on frame length, and V2.0 (Blue Book) based on frame type. Both of these formats are recognized automatically by the NetPerformer when the Ethernet operating protocol is set to **ETH AUTO**. The **ETH AUTO** value is thus the usual value for this parameter, and the value that must be selected if your Ethernet LAN uses mixed frame formats.

Select the ETH 802.3 value for this parameter if the frame format is exclusively 802.3 across the LAN, or if the frame length exceeds 1500 bytes (the **ETH AUTO** setting interprets all long frames as being in V2.0 format). The **ETH V2** value can be used when the format is exclusively V2.0 across the LAN. Use the **OFF** value when the Ethernet port is not used.

Values:	OFF, ETH AUTO, ETH 802.3, ETH V2
Default:	ETH AUTO

8.3 Link integrity

Console	SNMP	Text-based Config
Link integrity	iflanEth-LinkIntegrity	[iflan#] Eth-LinkIntegrity

Determines whether the Link Integrity feature is enabled or disabled. When Link Integrity is enabled (**YES**), the NetPerformer supports the Link Integrity mode. This is required when the Ethernet 10Base-T port is connected to a hub. When this parameter is set to **NO**, the NetPerformer does not support Link Integrity.

NOTE: The **LI** or **LNK** LED on the Ethernet port lights red when Link Integrity is established.

Values: NO, YES

Default: YES

8.4 LAN speed (mbps)

Console	SNMP	Text-based Config
LAN speed (mbps)	iflanSpeed	[iflan#] Speed

The speed of the Ethernet port. Use the **AUTO** setting for auto-detection of the LAN speed.

Values: AUTO, 10 MBPS, 100 MBPS

Default: AUTO

8.5 MAC address

Console	SNMP	Text-based Config
MAC address	iflanPhysAddr	[iflan#] PhysAddr

Provides the physical (or MAC) address of the NetPerformer. This address can be set to any value using 12 hexadecimal digits (6 bytes). However, the NetPerformer automatically sets the first byte to **02** for an Ethernet LAN, which forces the use of a locally administered address. When this parameter is set to **000000000000**, the NetPerformer uses its burned-in address.

NOTE: The burned-in address can be displayed using the Display States command, described later in this chapter.

Values: 000000000000 - FFFFFFFFFFFF

Default: 000000000000

8.6 Redundancy MAC address

Console	SNMP	Text-based Config
Redundancy MAC address	iflanRedunMacAddrress	[iflan#] RedunMacAddr

Determines the redundant MAC address that the Ethernet port will use as source address when the unit is operating in a redundant system.

Values:	00005E000000 - 00005E0000FF
Default:	00005E000000

8.7 Redundancy MAC address active

Console	SNMP	Text-based Config
Redundancy MAC address	iflanRedunMacAddressActive	[iflan#] RedunMacAddrActive

Allows for configuration of a redundant MAC address on this Ethernet port. This MAC address will be used as the source address instead of the default BIA or configured physical MAC address. If you set this parameter to NO, a redundant MAC address is not available for this Ethernet port.

NOTE: Set this parameter to YES in a redundant system. This allows the two redundant units to have the same MAC address and be on the same LAN. The unit that is currently active is the one that uses the redundant MAC address.

Values:	NO, YES
Default:	NO

8.8 DHCP

Console	SNMP	Text-based Config
DHCP	iflanDhcpEnable	[iflan#] DhcpEnable

When enabled, allows the first Ethernet IP address on the NetPerformer unit to be allocated by a DHCP server. *IP address 1* and *Subnet mask 1* are not available for configuration in this case, since they are automatically allocated by the DHCP server.

IP address 2, along with *Subnet mask 2,* can be used in a dual IP address application, but must be configured manually. The two IP addresses must be set on two separate IP networks or sub-networks.

NOTE: Leave *IP address 2* at its default value (**000.000.000.000**) when the NetPerformer requires only a single IP address.

Values: DISABLE, ENABLE

Default: DISABLE

8.9 Accept the default gateway from DHCP server

Console	SNMP	Text-based Config
Accept the default gateway from DHCP server	iflanDhcpAcceptGateway	[iflan#] DhcpAcceptGateway

For DHCP enabled only

Accepts (**YES**) or refuses (**NO**) the default gateway address provided by a DHCP server as the default gateway for the entire NetPerformer unit.

The IP address of the default gateway is determined from the following (in decreasing order of importance):

- The *Default gateway* parameter configured with the **GLOBAL** submenu of the Setup (**SE**) command
- The address of the gateway received from a DHCP server via the first LAN port (**ETH1**)
- The address of the gateway received from a DHCP server via the second LAN port (**ETH2**).

NOTE: If there is no known IP address on the LAN port, the source address in the IP header is set to **0.0.0.0** instead of any valid IP address on the NetPerformer unit. When a release is done or when the lease has expired, the default gateway and the local IP address of the LAN port are flushed.

You can also set a specific port, PVC, SVC or PPPoE connection as the default gateway, with the *Use this port as default gateway* parameter. Once the connection is up, it becomes the route for the default gateway, overriding both the global *Default gateway* parameter and the gateway address received from a DHCP server. This applies to the following network connections only:

- A WAN port or channel with *Protocol* set to **PVCR** or **PPP** (refer to the *WAN/Leased Lines* and *WAN/Point-to-Point Protocol (PPP)* fascicles of this document series)

- A Frame Relay PVC with *Mode* set to **PVCR** or **RFC1490** (refer to the *WAN/ Frame Relay* fascicle of this document series)
- An ATM PVC or SVC with *Mode* set to **ATMPVCR**, **ATMPPP** or **RFC1483** (refer to the *ATM Option* fascicle of this document series)
- Any PPPoE connection (refer to the *WAN/Point-to-Point Protocol (PPP)* fascicle of this document series).

Values: YES, NO

Default: YES

8.10 IP address 1

Console	SNMP	Text-based Config
IP address 1	iflanIpAddress	[iflan#] IpAddress

Available only when DHCP is disabled.
Provides the IP address of the local port. It is a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte, for example 1**28.128.128.122**. When this parameter is set to **000.000.000.000**, no IP address is defined for the Ethernet port.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

8.11 IP address 2

Console	SNMP	Text-based Config
IP address 2	iflanIpAddress2	[iflan#] IpAddress2

This parameter is used for dual IP address applications to provide the IP address of the second local LAN port. Like *IP address 1*, it is a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

NOTE: Leave *IP address 2* at its default value (**000.000.000.000**) when the NetPerformer requires only a single IP address.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

8.12 Redundancy IP address 1

Console	SNMP	Text-based Config
Redundancy IP address	iflanRedunIPAddrress1	[iflan#] RedunIPAddress1

Specifies the first redundant IP address that is activated by the Redundancy feature when the unit is operative. The IP address value is class A, class B or class C. The IP address is in x.x.x.x format.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

8.13 Redundancy IP address 2

Console	SNMP	Text-based Config
Redundancy IP address 2	iflanRedunIPAddrress2	[iflan#] RedunIPAddress2

Specifies the second redundant IP address that is activated by the Redundancy feature when the unit is operative. The IP address value is class A, class B or class C. The IP address is in x.x.x.x format.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

8.14 Subnet mask 1 (number of bits)

Console	SNMP	Text-based Config
Subnet mask 1 (number of bits)	iflanSubnetMask	[iflan#] SubnetMask

Available only when DHCP is disabled.
The subnet mask associated with the Ethernet port's IP Address. It is configured like the IP address: a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The subnet mask identifies which bits of the IP address correspond to the physical network, and which bits correspond to host identifiers. For example, in address 255.255.000.000 all network bits are set to **1** and all host bits are set to **0**.

To change the value of the subnet mask using the console, enter the number of bits of that mask.

- For example, select **17** bit to define the mask 255.255.128.000; select **23** bits to define 255.255.254.000

- When you enter the number of bits at the console, the NetPerformer provides the resulting mask in dotted decimal notation to the right of the bits value
- If the value of the subnet mask is not valid for the IP address configured, it will be rejected by the NetPerformer and the IP address will be invalid.

NOTE: As of V10.2 the NetPerformer supports supernetting as well as subnetting. To accommodate this the subnet mask of any IP address can now be set to any whole integer value from **0** to **32**.

Values: 0 - 32 (equivalent to 000.000.000.000 - 255.255.255.255)

Default: 8 (equivalent to 255.000.000.000)

8.15 Subnet mask 2 (number of bits)

Console	SNMP	Text-based Config
Subnet mask 2 (number of bits)	iflanSubnetMask2	[iflan#] SubnetMask2

The subnet mask associated with the second Ethernet port's IP Address for dual IP address applications. It is configured like the IP address: a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte. For more information about this parameter, see the description for Subnet mask 1 (number of bits) above.

Values: 0 - 32 (equivalent to 000.000.000.000 - 255.255.255.255)

Default: 8 (equivalent to 255.000.000.000)

8.16 Redundancy Subnet mask 1 (number of bits)

Console	SNMP	Text-based Config
Redundancy Subnet mask 1 (number of bits)	iflanRedunSNMask1	[iflan#] RedunSNMask1

Specifies the subnet mask for the first redundant IP address, in number of bits. For example, the value **8** corresponds to **255.0.0.0**, and 16 corresponds to **255.255.0.0**.

Values: 0 - 32 (equivalent to 000.000.000.000 - 255.255.255.255)

Default: 8 (equivalent to 255.000.000.000)

8.17 Redundancy Subnet mask 2 (number of bits)

Console	SNMP	Text-based Config
Redundancy Subnet mask 2 (number of bits)	iflanRedunSNMask2	[iflan#] RedunSNMask2

Specifies the subnet mask for the second redundant IP address, in number of bits. For example, the value **8** corresponds to **255.0.0.0**, and 16 corresponds to **255.255.0.0**.

Values:	0 - 32 (equivalent to 000.000.000.000 - 255.255.255.255)
Default:	8 (equivalent to 255.000.000.000)

8.18 Allow routing between IP network #1 and #2

Console	SNMP	Text-based Config
Allow routing between IP network #1 and #2	iflanAllowNetToNetIp-Routing	[iflan#] AllowNetToNet-IpRouting

Permits (**YES**) or prevents (**NO**) routing between the two IP networks that are accessed via the two IP addresses configured on a single LAN port (*IP address 1* and *IP address 2* parameters).

NOTE: This parameter is listed at the console only when **both** LAN IP addresses are configured. It does **not** affect routing between the two LAN ports on a single unit. This is controlled with the **IP/GLOBAL** *Allow LAN-to-LAN routing* parameter (see page 5).

Values:	YES, NO
Default:	YES

8.19 Frame size

Console	SNMP	Text-based Config
Frame size	iflanMaxFrame	[iflan#] MaxFrame

Determines the largest datagram, in octets, that can be sent or received on the interface in one IP frame. Datagrams larger than the maximum frame size are divided into fragments before transmission, then reassembled at the remote end.

Values:	128 - 8192
Default:	1500

8.20 IP RIP

Console	SNMP	Text-based Config
IP RIP	iflanIpRip	[iflan#] IpRip

Enables or disables the Routing Information Protocol (RIP) on the LAN interface. Three settings are available to enable IP RIP:

- **V1**: The Ethernet port uses RIP Version 1. With this version the subnet mask for an IP address in a routing table entry is determined using the mask of the port on which the frame was received.
- **V2 BROADCAST**: The port uses RIP Version 2 in Broadcast mode. In Version 2 a subnet mask is transmitted for each address contained in the RIP frame. In Broadcast mode each RIP V2 frame is sent with IP address 255.255.255.255, which permits routers running RIP Version 1 to receive and analyze those frames.
- **V2 MULTICAST**: The port uses RIP Version 2 in Multicast mode. In this mode each RIP V2 frame is sent with IP address 224.000.000.009, which prevents routers running RIP Version 1 from receiving those frames.

For details on the differences between IP RIP routing using RIP Version 1 and RIP Version 2, consult Table 1, Processing of RIP Frames Received, on page 5.

Set the IP RIP parameter to **DISABLE** to prevent the NetPerformer from transmitting or receiving RIP frames on the Ethernet port. The NetPerformer will discard all RIP frames received.

NOTE: If you disable IP RIP on the NetPerformer, but require IP routing to a particular destination (for management under SNMP, for example) you can configure a static IP address using the IP Static menu, described in the section Configuring the Static IP Parameters on page 21.

Values:	DISABLE, V1, V2 BROADCAST, V2 MULTICAST
Default:	V1

NOTE: When *IP RIP* is set to V2 **BROADCAST** or **V2 MULTICAST**, additional IP RIP parameters are also available, as shown in this example:

```
PORT ETH 1> IP RIP (def:V1) ? V2 BROADCAST
PORT ETH 1> IP RIP TX/RX (def:DUPLEX) ?
PORT ETH 1> IP RIP Authentication (def:NONE) ?
PORT ETH 1> IP RIP Password (def:) ?
```

8.21 IP RIP TX/RX

Console	SNMP	Text-based Config
IP RIP TX/RX	iflanIpRipTxRx	[iflan#] IpRipTxRx

Sets the directionality of the RIP version used on the LAN interface. Configure the port for two-way IP RIP routing by setting this parameter to **DUPLEX** (the default value). The NetPerformer can generate IP routing tables, and both receive and transmit RIP frames on this port.

You can also enable *IP RIP* in a single direction. Select the **TX ONLY** value to allow the NetPerformer to transmit RIP frames only. The NetPerformer will discard all RIP frames received at the LAN interface. Select **RX ONLY** to allow the NetPerformer to receive RIP frames only. In this case, the Ethernet port cannot transmit a RIP frame.

Values:	DUPLEX, TX ONLY, RX ONLY
Default:	DUPLEX

8.22 IP RIP authentication

Console	SNMP	Text-based Config
IP RIP authentication	iflanIpRipAuthType	[iflan#] IpRipAuthType

For IP RIP Version 2 only.
Enables or disables password authentication for the interface. Select **SIMPLE** to have the password included in all RIP frames sent from the Ethernet port. Frames containing authentication that are received at this port will be accepted only if the password is valid. The password is defined using the IP RIP Password parameter, described below.

Values:	NONE, SIMPLE
Default:	NONE

8.23 IP RIP password

Console	SNMP	Text-based Config
IP RIP password	iflanIpRipPassword	[iflan#] IpRipPassword

For IP RIP Version 2 only.
Defines the password to be used on the interface. The password allows the authentication procedure to generate and/or verify the authentication field in the RIP header.

NOTE: The value of the *IP RIP Password* parameter must be the same for each interface on both sides of the network.

Values:	alphanumeric string, maximum 8 characters
Default:	none

8.24 OSPF

Console	SNMP	Text-based Config
OSPF	iflanOspfEnable	[iflan#] OspfEnable

Enables or disables the OSPF protocol on the interface.

NOTE: By default, OSPF is **disabled** on all NetPerformer slots, ports and PVCs.

- **ENABLE:** Enables OSPF on this port. Each slot, port and PVC must be enabled separately.
  - If OSPF is enabled, the LAN connection will be advertised as an internal route to an area of the Autonomous System (AS).
  - You must also define the following parameters, which are presented on the console immediately after the OSPF parameter:

```
PORT ETH 1> OSPF (def:DISABLE) ? ENABLE
PORT ETH 1> OSPF Area ID (def:000.000.000.000) ?
PORT ETH 1> OSPF Router priority (0-255,def:1) ?
PORT ETH 1> OSPF Transit delay (1-360,def:1) ?
PORT ETH 1> OSPF Retransmit interval (1-360,def:5) ?
PORT ETH 1> OSPF Hello interval (1-360,def:10) ?
PORT ETH 1> OSPF Dead interval (1-2000,def:40) ?
PORT ETH 1> OSPF Authentication type (def:NONE) ?
PORT ETH 1> OSPF Metric cost (1-65534,def:10) ?
```

These additional OSPF parameters are detailed further below.

- **DISABLE:** Disables OSPF on this port. Select this value if you do not want the WAN link to participate in an OSPF network.
  - In this case, the other parameters related to OSPF configuration are not presented on the console.

Values: DISABLE, ENABLE

Default: DISABLE

8.25 OSPF Area ID

Console	SNMP	Text-based Config
OSPF Area ID	iflanOspfAreaId	[iflan#] OspfAreaId

Identifies the area to which this interface belongs. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte. All routing protocol packets originating from the interface are labelled with this Area ID.

The value of this parameter must be the same as the Area ID of the area to which the attached network belongs (see Defining OSPF Area Characteristics on page 17). If you want to define subnetted networks as separate areas, you can use the IP network number as the Area ID.

NOTE: Area ID 000.000.000.000 indicates that this interface is included in the OSPF backbone.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

8.26 OSPF Router priority

Console	SNMP	Text-based Config
OSPF Router priority	iflanOspfPriority	[iflan#] OspfPriority

For LAN ports only.

The priority assigned to this interface for determining the designated router of the attached multi-access network. The value of this parameter is an 8-bit unsigned integer that is advertised in Hello Packets sent out from this interface.

If two routers attached to the same network both attempt to become the designated router, the one with the highest Router Priority value will take precedence. If they have the same priority, the router with the highest *Router ID* (an OSPF Global parameter) will take precedence.

Set this parameter to **0** if you do not want the NetPerformer to become the designated router of the attached network.

Values: 0 - 255

Default: 1

8.27 OSPF Transit delay

Console	SNMP	Text-based Config
OSPF Transit delay	iflanOspfTransitDelay	[iflan#] OspfTransitDelay

The estimated number of seconds required to transmit a Link State Update packet over this interface. Link state advertisements contained in the Link State Update packet will have their age incremented by this amount before transmission.

When configuring the Transit Delay, take into account the transmission and propagation delays that occur on this port. For example, you should increase the value of the Transit Delay for low-speed serial connections. The default value, 1 second, is appropriate for a LAN connection.

Values: 1 - 360

Default: 1

8.28 OSPF Retransmit interval

Console	SNMP	Text-based Config
OSPF Retransmit interval	iflanOspfRetransmitInt	[iflan#] OspfRetransmi-tInt

The number of seconds that elapse between retransmissions of link state advertisements. This parameter is used for adjacencies that belong to this interface, and for retransmissions of OSPF Database Description and Link State Request packets.

Set this parameter to a value that is higher than the expected round-trip delay between any two routers on the network attached to this port. Otherwise, needless retransmissions will occur. The default value, 5 seconds, is appropriate for a LAN connection. Low-speed links require a higher value.

Values: 1 - 360

Default: 5

8.29 OSPF Hello interval

Console	SNMP	Text-based Config
OSPF Hello interval	iflanOspfHelloInt	[iflan#] OspfHelloInt

The length of time, in seconds, between the Hello Packets that the NetPerformer sends on this interface. The value of the Hello Interval parameter is advertised in Hello Packets sent out from this interface, and must be the same on all other routers having a connection with the network attached to this interface.

If you set the Hello Interval to a short length of time, changes to the OSPF topological database will be detected more quickly. However, a short Hello Interval creates more OSPF routing protocol traffic. The default value, 10 seconds, is appropriate for a LAN connection, whereas a PVC may require a Hello Interval of 30 seconds.

Values:	1 - 360
Default:	10

8.30 OSPF Dead interval

Console	SNMP	Text-based Config
OSPF Dead interval	iflanOspfDeadInt	[iflan#] OspfDeadInt

The length of time, in seconds, before neighboring routers declare a router down when they stop hearing its Hello Packets. The value of the Dead Interval parameter is advertised in Hello Packets sent out from this interface, and must be the same on all other routers having a connection with the network attached to this interface.

Set the Dead Interval to a multiple of the Hello Interval (described above).

Values:	1 - 2000
Default:	40

8.31 OSPF Authentication type

Console	SNMP	Text-based Config
OSPF Authentication type	iflanOspfAuthType	[iflan#] OspfAuthType

Determines the type of authentication that will be performed to generate and/or verify OSPF protocol packets sent over this LAN connection.

- **SIMPLE:** The *OSPF Password* (see page 17) is used for all OSPF packets sent on the network. All OSPF packets sent on this network must contain this value in the

Authentication field of the OSPF header. The remaining contents of each OSPF packet are also verified with a checksum operation.

Caution: The **SIMPLE** authentication type does not protect the OSPF routing domain from passive attacks via the Internet, as anyone with physical access to the network can learn the *OSPF Password* that is used.

•**CRYPTOGRAPHIC:** A shared secret key, configured with the *OSPF Cryptographic auth. key* parameter (see page 17), is used to generate and/or verify a one-way *message digest* that is appended to each OSPF packet sent over this LAN connection. **CRYPTOGRAPHIC** authentication also includes the use of:

- The MD5 algorithm, to generate the message digest
- A sequence number in the *Authentication* field of each OSPF packet, to protect against replay attacks.

NOTE: This is a more secure authentication method, as the *OSPF Cryptographic auth. key* itself is never sent over the LAN connection.

- **NONE:** No authentication is performed on OSPF protocol packets.

Values:	NONE, SIMPLE, CRYPTOGRAPHIC
Default:	NONE

When the *OSPF Authentication type* parameter is set to **CRYPTOGRAHPIC**, three additional parameters are displayed at the console (described below):

```
...
PORT ETH 1> OSPF Authentication type (def:NONE) ? CRYPTOGRAPHIC
PORT ETH 1> OSPF Cryptographic auth. ID (0-255,def:1) ?
PORT ETH 1> OSPF Cryptographic auth. key (def:) ?
```

When the *OSPF Authentication type* parameter is set to **SIMPLE**, one additional parameter is displayed at the console (described on page 17):

```
...
PORT 1> OSPF Authentication type (def:NONE) ? SIMPLE
PORT 1> OSPF Password (def:) ?
```

8.32 OSPF Cryptographic auth. ID

Console	SNMP	Text-based Config
OSPF Cryptographic auth. ID	iflanOspfCryptoAuthId	[iflan#] OspfCryptoAuthId

For CRYPTOGRAPHIC Authentication type only

Identifies which algorithm and shared secret key will be used to create the message digest appended to an OSPF packet. This ID is particular to this LAN connection only.

Values:	0 - 255
Default:	1

8.33 OSPF Cryptographic auth. key

Console	SNMP	Text-based Config
OSPF Cryptographic auth. key	iflanOspfCryptoAuthKey	[iflan#] OspfCryptoAuth-Key

For CRYPTOGRAPHIC Authentication type only

Defines the value of the shared secret key that will be used to create the message digest for OSPF packets sent over this LAN connection.

Values:	maximum 16-character string
Default:	no value

8.34 OSPF Password

Console	SNMP	Text-based Config
OSPF Password	iflanOspfPassword	[iflan#] OspfPassword

For SIMPLE Authentication type only

Defines the 64-bit value that will appear in the authentication field of all OSPF packets sent or received on this LAN connection. The *OSPF Password* allows the authentication procedure to generate and/or verify the *Authentication* field in the OSPF header.

⚠ **Caution:** The value of the *OSPF Password* parameter must be the same as that configured on all other routers having a connection with the network on this LAN. In other words, all routers in the same area must have the same password (or no authentication at all).

NOTE: Since the *OSPF Password* is configured separately for each interface, there can be a separate password for each network in the AS.

Values: Maximum 8-character string

Default: none

8.35 OSPF Metric cost

Console	SNMP	Text-based Config
OSPF Metric cost	iflanOspfMetricCost	[iflan#] OspfMetricCost

The cost of sending a packet on this interface, expressed in the link state metric. The value of this parameter is determined from:

100,000,000 ÷ *interface speed*

The metric cost is advertised in the router links advertisement as the link cost for this interface.

Values: 1 - 65534

Default: **10**

8.36 IGMP enable

Console	SNMP	Text-based Config
IGMP enable	iflanIpIgmpEnable	[iflan#] IpIgmpEnable

Enables (**YES**) or disables (**NO**) IGMP to allow the NetPerformer to keep track of membership in a multicast group. IGMP must be enabled to define the IP multicast addresses.

Values: NO, YES

Default: **NO**

NOTE: When *IGMP* is set to **ENABLE**, additional IGMP parameters are also available, as shown in this example:

```
PORT ETH 1> IGMP enable (def:NO) ? YES
PORT ETH 1> IGMP version (1-2,def:2) ?
PORT ETH 1> IGMP send report (def:NO) ?
```

8.37 IGMP version 1,2

Console	SNMP	Text-based Config
IGMP version 1,2	iflanIpIgmpVersion	[iflan#] IpIgmpVersion

Determines which version of the IGMP protocol will be used.

Values: 1 - 2

Default: **2**

8.38 IGMP send report

Console	SNMP	Text-based Config
IGMP send report	iflanIpIgmpSendReport	[iflan#] IpIgmpSendReport

Determines whether an IGMP membership report should be sent for each IP multicast address. This can be helpful when interfacing with a router that uses a protocol different from PIM-DM on its LAN port.

Values: NO, YES

Default: **NO**

8.39 IP multicast active

Console	SNMP	Text-based Config
IP multicast active	iflanIpMulticastActive	[iflan#] IpMulticastActive

Enables (**YES**) or disables (**NO**) IP Multicast on this LAN connection. When *IP multicast active* is enabled, the port becomes an IP Multicast client.

Values: NO, YES

Default: NO

NOTE: When *IP multicast active* is set to **YES**, additional parameters are available so you can configure the IP addresses of all multicast groups that will be recognized by the LAN port:

```
PORT ETH 1> IP multicast active (NO/YES,def:NO) ? YES
PORT ETH 1> IP multicast protocol (NONE/PIMDM,def:NONE) ? PIMDM
PORT ETH 1> IP multicast 1 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 2 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 3 (def:000.000.000.000) ?
PORT ETH 1> IP multicast 4 (def:000.000.000.000) ?
```

8.40 IP multicast protocol

Console	SNMP	Text-based Config
IP multicast protocol	iflanIpMulticastProtocol	[iflan#] IpMulticastProtocol

Selects the IP multicast protocol for communications between routers.

- **PIMDM:** Protocol Independent Multicast - Dense Mode. This routing algorithm was designed for multicast groups that are densely distributed across the network.
- **NONE:** No IP multicast protocol is used.

Values: NONE, PIMDM

Default: NONE

8.41 IP multicast 1,2,3,4

Console	SNMP	Text-based Config
IP multicast 1	iflanIpMulticastAddr1	[iflan#] IpMulticastAddr1
IP multicast 2	iflanIpMulticastAddr2	IpMulticastAddr2
IP multicast 3	iflanIpMulticastAddr3	IpMulticastAddr3
IP multicast 4	iflanIpMulticastAddr4	IpMulticastAddr4

Four distinct *IP multicast* addresses can be defined to determine which groups will be targeted for group-specific queries and membership reports. If all are left at the default value, **000.000.000.000**, no multicast groups will be recognized by the Ethernet port, and **IP Multicast will not work**.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

8.42 NAT enable

Console	SNMP	Text-based Config
NAT enable	iflanNatEnable	[iflan#] NatEnable

Enables (**YES**) or disables (**NO**) Network Address Translation (NAT) on this Ethernet port.

Values:	NO, YES
Default:	NO

NOTE: When *NAT enable* is set to **YES**, additional NAT parameters are also available, as shown in this example:

```
PORT ETH 1> NAT enable (def:NO) ? YES
PORT ETH 1> NAT rule (1-10) (def:) ?
PORT ETH 1> NAT side (def:INTERNAL) ?
```

8.43 NAT rule

Console	SNMP	Text-based Config
NAT rule	iflanNatRule	[iflan#] NatRule

Selects the NAT rule or rules to be used to translate address information for traffic to and from this LAN link. A rule defines the correspondence between internal IP addresses and external, globally unique NAT IP addresses. Select multiple rules by entering a comma between the rule numbers, for example: **1,3,4**.

Define all NAT rules with the **SETUP/IP/NAT** submenu. For details, refer to the chapter Network Address Translation (NAT) on page 1.

Values:	1 - 10
Default:	none

8.44 NAT side

Console	SNMP	Text-based Config
NAT side	iflanNatSide	[iflan#] NatSide

Determines which address realm this LAN link is associated with, and where NAT is carried out.

- **INTERNAL:** NAT is carried out on the internal side of the network. Select this value if the link connects to equipment on the local side (the private network).
- **EXTERNAL:** NAT is carried out on the external side of the network. Select this value if the LAN link connects to equipment on the remote side.

Values: INTERNAL, EXTERNAL

Default: INTERNAL

8.45 VLAN enable

Console	SNMP	Text-based Config
VLAN enable	iflanVlanEnable	[iflan#] VlanEnable

Enables (**YES**) or disables (**NO**) VLAN communications on the Ethernet port. Leave *VLAN enable* at its default value, **NO**, if you do not want the LAN port to be VLAN-aware.

NOTE: To set up a VLAN, consult the *Virtual LAN (VLAN)* fascicle of this document series.

Values: NO, YES

Default: NO

NOTE: When *VLAN enable* is set to **YES**, additional VLAN parameters are also available, as shown in this example:

```
PORT ETH 1> VLAN enable (def:NO) ? YES
PORT ETH 1> VLAN number (1-4095,def:1) ?
PORT ETH 1> VLAN Priority Conversion (def:NO) ?
```

8.46 VLAN number

Console	SNMP	Text-based Config
VLAN number	iflanVlanNumber	[iflan#] VlanNumber

The VLAN number, required for access to the NetPerformer unit using Telnet, FTP or SNMP via a specific VLAN.

Values:	1 - 4095
Default:	1

8.47 VLAN Priority Conversion

Console	SNMP	Text-based Config
VLAN Priority Conversion	iflanPriorityConversion	[iflan#] PriorityConversion

Enables (**YES**) or disables (**NO**) traffic priority information in the VLAN Tag Header. Set *VLAN Priority Conversion* to **YES** if you want this information to be preserved when routing broadcast frames.

Values:	NO, YES
Default:	NO

8.48 BRG enable

Console	SNMP	Text-based Config
BRG enable	iflanBridgeEnable	[iflan#] BridgeEnable

Enables (**YES**) or disables (**NO**) bridging on this LAN port. To ensure that the bridged traffic travels exclusively within a single LAN, **bridge functions must enabled on that LAN port, and *disabled* on the other LAN port**. This is required to support bridged traffic over a PowerCell over IP connection. Refer to Bridging Traffic over PowerCell over IP on page 4.

Values:	NO, YES
Default:	YES

8.49 IPX RIP

Console	SNMP	Text-based Config
IPX RIP	iflanIpxRip	[iflan#] IpxRip

Enables or disables RIP for Internetwork Packet Exchange (IPX) frames. Configure the Ethernet port for IPX RIP routing by setting this parameter to **ENABLE**. When the *IPX RIP* parameter is enabled, the NetPerformer can generate IPX routing tables, and both receive and transmit IPX RIP frames on this port. When *IPX RIP* is disabled the NetPerformer cannot transmit an IPX RIP frame, and discards all IPX RIP frames received.

NOTE: If you set this parameter to **ENABLE**, you must also configure the *IPX Network Number* and *IPX Encapsulation* parameters, described below.

Values:	DISABLE, ENABLE
Default:	DISABLE

8.50 IPX SAP

Console	SNMP	Text-based Config
IPX SAP	iflanIpxSap	[iflan#] IpxSap

Enables or disables the Service Advertising Protocol (SAP) for IPX frames. IPX SAP frames are exchanged between routers to indicate the nature and location of services available on a Novell network. Configure the Ethernet port for IPX SAP routing by setting this parameter to **ENABLE**. When the *IPX SAP* parameter is disabled the NetPerformer cannot transmit an IPX SAP frame, and discards all IPX SAP frames received.

NOTE: If you set this parameter to **ENABLE**, you must also configure the *IPX Network Number* and *IPX Encapsulation* parameters, described below.

Values:	DISABLE, ENABLE
Default:	DISABLE

8.51 IPX network number

Console	SNMP	Text-based Config
IPX network number	iflanIpxNetNum	[iflan#] IpxNetNum

The network number of the IPX node that is connected to the Ethernet port. This number is a 4-byte value in hexadecimal representation. The NetPerformer uses IPX network numbers to forward frames to their final destination.

When the IPX network number is set to **00000000**, the local node is unknown. To allow the NetPerformer to forward IPX frames to their final destination, an internal IPX network number must be defined, using the Setup IPX menu. For details, refer to the *Digital Data* fascicle of this document series.

Values: 00000000 - FFFFFFFF

Default: 00000000

8.52 IPX encapsulation

Console	SNMP	Text-based Config
IPX encapsulation	iflanIpxLanType	[iflan#] IpxLanType

The frame type used on the Ethernet LAN to encapsulate IPX frames. The following parameter values are available:

- **ETH 802.2**: Specifies Ethernet (802.3) frames using an 802.2 envelope. The DSAP and SSAP values indicate that the packets contain IPX frames.
- **ETH SNAP**: Specifies Ethernet (802.3) frames using an 802.2 envelope with SNAP (protocol ID). The DSAP and SSAP values indicate SNAP encapsulation, and the protocol ID indicates the presence of an IPX frame.
- **ETH 802.3**: Specifies raw IPX encapsulation of Ethernet (802.3) frames. This frame format cannot be used with IPX checksums.
- **ETH II**: Specifies Ethernet (802.3) frames using a DEC Ethernet II envelope.

Values: ETH 802.2, ETH SNAP, ETH 802.3, ETH II

Default: ETH 802.2

8.53 Physical connectivity detection

Console	SNMP	Text-based Config
Physical connectivity detection	iflanPhyConnectEnable	[iflan#] PhyConnectEnable

Indicates whether or not the physical connection to the Ethernet port will be monitored and detected.

Values: DISABLE, ENABLE

Default: DISABLE

9

SE/IP Configuration Parameters

NOTE: The **TELNET**, **FTP** and **RADIUS** submenus of the Setup IP command are dealt with in the chapter *Controlling Access to the NetPerformer* in the *Quick Configuration* fascicle of this document series. The NAT parameters are detailed in the chapter Network Address Translation (NAT) on page 1.

9.1 SE/IP/GLOBAL Submenu

9.1.1 Router

Console	SNMP	Text-based Config
Router	npipRouterEnable	[npip] RouterEnable

Enables or disables all IP functions and commands on the NetPerformer. When the *Router* parameter is enabled, the NetPerformer behaves like an IP router, and will respond to IP commands. When this parameter is disabled, the NetPerformer ignores all IP commands received.

Values:	DISABLE, ENABLE
Default:	ENABLE

9.1.2 Route broadcast to end station

Console	SNMP	Text-based Config
Route broadcast to end station	npipRouteBrdcastToEnd-Station	[npip] RouteBrdcastTo-EndStation

Used in IP Subnet Broadcasting and IP All Subnet Broadcasting. Set this parameter to **YES** to route all broadcast IP frames to the end station. The default value, **NO**, should be chosen for standard processing of broadcast IP frames.

Values:	NO, YES
Default:	NO

9.1.3 OSPF AS boundary router

Console	SNMP	Text-based Config
OSPF AS boundary router	npipOspfAsBoundary-Router	[npip] OspfAsBoundary-Router

Permits a redistribution of RIP entries through OSPF frames in an integrated IP RIP/OSPF network. Set this parameter to **YES** to allow RIP to OSPF redistribution. The default value, **NO**, should be chosen for standard processing of RIP entries.

Values:	NO, YES
Default:	NO

When the *OSPF AS boundary router* parameter is set to **YES**, three additional parameters are displayed at the console (described below):

. . .

```
IP> RIP to OSPF metric conversion cost (1-65534,def:2000) ?
IP> OSPF AS forwards RIP entries (def:YES) ?
IP> OSPF AS forwards STATIC entries (def:YES) ?
```

9.1.4 RIP to OSPF metric conversion cost

Console	SNMP	Text-based Config
RIP to OSPF metric conversion cost	npipRipOspfMetricCost	[npip] RipOspfMetric-Cost

For OSPF AS boundary router only
Determines the multiplicative factor to be used when converting RIP entries to OSPF in the routing table. The metric cost is the relative distance to another router in the network. A central point router must be able to convert the metric cost under RIP to the equivalent cost under OSPF.

Set this parameter to the multiplicative factor required to convert the metric cost under RIP to the equivalent cost under OSPF.

Values:	1 - 65534
Default:	2000

9.1.5 OSPF AS forwards RIP entries

Console	SNMP	Text-based Config
OSPF AS forwards RIP entries	npipOspfAsFwdRip	[npip] OspfAsFwdRip

For OSPF AS boundary router only
Determines whether the AS boundary router forwards RIP IP address entries in its routing table to other routers. RIP entries are forwarded by default. Set *OSPF AS forwards RIP entries* to **NO** to prevent this.

Values:	NO, YES
Default:	YES

9.1.6 OSPF AS forwards STATIC entries

Console	SNMP	Text-based Config
OSPF AS forwards STATIC entries	npipOspfAsFwdStatic	[npip] OspfAsFwdStatic

For OSPF AS boundary router only
Determines whether the AS boundary router forwards static IP address entries in its routing table to other routers. STATIC entries are forwarded by default. Set *OSPF AS forwards RIP entries* to **NO** to prevent this.

Values: NO, YES

Default: YES

9.1.7 RIP AS boundary router

Console	SNMP	Text-based Config
RIP AS boundary router	npipRipAsBoundary-Router	[npip] RipAsBoundary-Router

Permits a redistribution of OSPF entries through RIP frames in an integrated IP RIP/OSPF network.

Set this parameter to **YES** to allow OSPF to RIP redistribution. The default value, **NO**, should be chosen for standard processing of OSPF entries.

Values: NO, YES

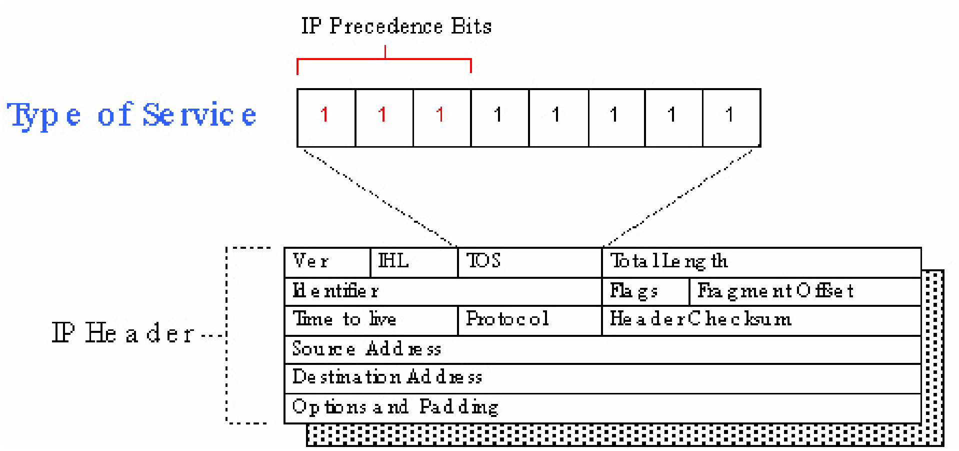
Default: NO

9.1.8 IP Precedence for FR over IP

Console	SNMP	Text-based Config
IP Precedence for FR over IP	npipFroIpPrecedence	[npip] FroIpPrecedence

Specifies which IP precedence bit settings should be applied to the Frame over IP traffic. This can be used to prioritize Frame over IP traffic.

Three IP Precedence bits can be set from the Type of Service (TOS) field in the IP header. Eight classes of traffic can be defined, as shown below.



IP Precedence Bits

IP Precedence for FR over IP determines the relative priority of Frame Relay over IP traffic, and usually represents classes of service from **0** to **7**. Typically, the **0** value is used when no QoS is implemented in the IP backbone network.

NOTE: The specific priority level of each class will vary depending on the IP service provided (private or public). However, **0** (zero) is commonly associated with the lowest priority level, and **7** the highest priority level.

Values: 0 - 7

Default: 0

9.1.9 Allow LAN-to-LAN routing

Console	SNMP	Text-based Config
Allow LAN-to-LAN routing	npipAllowLanToLanIp-Routing	[npip] AllowLanToLanIp-Routing

Permits (**YES**) or prevents (**NO**) routing between the two LAN ports on a single unit.

NOTE: This parameter does **not** affect routing between the two IP networks that can be configured on a single LAN port. This is controlled with the Ethernet port *Allow routing between IP network #1 and #2* parameter (see page 9).

Values: YES, NO

Default: YES

9.2 SE/IP/STATIC Submenu

9.2.1 IP static entry number

Console	SNMP	Text-based Config
IP static entry number	ipstaticEntry, ipstaticIndex	[ipstatic] Entry [ipstatic] Index

Enter the number of the IP Static entry you want to configure on the console command line. For SNMP, select the *ipstaticEntry* table and look under the *ipstaticIndex* for the desired IP Static entry.

Once you select an IP Static entry, the entry number is displayed thereafter at the beginning of each line from the console.

Values:	1 - 200
Default:	1

9.2.2 Valid

Console	SNMP	Text-based Config
Valid	ipstaticValid	[ipstatic] Valid

Sets the activation status of the IP Static entry. If you activate the entry (**YES**), the NetPerformer will add the entry information to its IP RIP routing table. If you select **NO**, the entry is ignored.

Values:	NO, YES
Default:	NO

9.2.3 Destination address

Console	SNMP	Text-based Config
Destination address	ipstaticIpDest	[ipstatic] IpDest

Provides the network address of the destination device. It is a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte, for example 128.128.0.0. When this parameter is set to **000.000.000.000**, no IP address is defined for this IP Static entry.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.2.4 Subnet mask (number of bits)

Console	SNMP	Text-based Config
Subnet mask (number of bits)	ipstaticMask	[ipstatic] Mask

The subnet mask associated with the Destination Address. It is configured like the Destination Address: a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The subnet mask identifies which bits of the Destination Address correspond to the physical network, and which bits correspond to host identifiers. For example, in address 255.255.000.000 all network bits are set to 1 and all host bits are set to **0**.

To change the value of the subnet mask using the console, enter the number of bits of that mask.

- For example, select **17** bit to define the mask 255.255.128.000; select **23** bits to define 255.255.254.000
- When you enter the number of bits at the console, the NetPerformer provides the resulting mask in dotted decimal notation to the right of the bits value
- If the value of the subnet mask is not valid for the *Destination address* configured, it will be rejected by the NetPerformer and the *Destination address* will be invalid.

NOTE: As of V10.2 the NetPerformer supports supernetting as well as subnetting. To accommodate this the subnet mask of any IP address can now be set to any whole integer value from **0** to **32**.

Values:	0 - 32 (equivalent to 000.000.000.000 - 255.255.255.255)
Default:	8 (equivalent to 255.000.000.000)

9.2.5 Next hop

Console	SNMP	Text-based Config
Next hop	ipstaticNextHop	[ipstatic] NextHop

The next unit to be reached on the path to the final destination. For the IP Static entry, this is the IP address of the remote NetPerformer that will be used to send the IP frame to the Destination Address.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.3 SE/IP/BOOTP Submenu

9.3.1 BOOTP

Console	SNMP	Text-based Config
BOOTP	bootpEnable	[bootp] Enable

Enables or disables the BOOTP protocol on the NetPerformer. When BOOTP is enabled the NetPerformer will act as a BOOTP/DHCP relay agent, and can forward BOOTREQUEST frames to their proper destination (as long as the Router parameter of the Setup IP Global menu is also enabled). When this parameter is disabled, the NetPerformer will flush any BOOTREQUEST frames received.

Values:	DISABLE, ENABLE
Default:	DISABLE

9.3.2 Max hops

Console	SNMP	Text-based Config
Max hops	bootpMaxHops	[bootp] MaxHops

The maximum number of BOOTP relay agents that a frame can cross on the way to its destination. The NetPerformer discards any frame with a hop count that exceeds this limit. This avoids the possibility of having a frame that loops and never dies.

The hop count is increased each time the frame crosses a relay agent. It is not increased when the frame crosses an intervening router on its way to the destination IP address on a different LAN. In fact, intervening routers do not require the relay agent capability in order to forward BOOTP frames, as the BOOTP frames are treated like regular IP frames on these routers.

When setting the Maximum Hop value, consider the impact of this parameter on normal traffic flow. The connectivity requirements of your network should not be limited by a Maximum Hop value that is too low.

NOTE: If two NetPerformers with the BOOTP relay agent are connected via a WAN link, we suggest that you set the Maximum Hop parameter to the same value on both units to ensure conformity across the network.

Values:	0 - 16
Default:	4

9.3.3 Destination IP address 1,2,3,4

Console	SNMP	Text-based Config
Destination IP address 1 Destination IP address 2 Destination IP address 3 Destination IP address 4	bootpIpDestAddr1 bootpIpDestAddr2 bootpIpDestAddr3 bootpIpDestAddr4	[bootp] IpDestAddr1 IpDestAddr2 IpDestAddr3 IpDestAddr4

Provides the IP address of the next BOOTP relay agent or server. It is a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte, for example 128.128.0.0. Up to 4 Destination IP Addresses can be defined. When the NetPerformer receives a BOOTREQUEST frame, it forwards the frame to the required address.

NOTE: A BOOTREQUEST frame can be forwarded to the next relay agent or directly to the server, but cannot be forwarded along the port on which the frame was received. Forwarding directly to the server is preferred because of reduced processing cost.

When all 4 Destination IP Addresses are set to **000.000.000.000**, BOOTP enters Broadcast Mode. In this mode the NetPerformer forwards the BOOTREQUEST frame on each open port (except the initiating port) using the IP broadcast method. This mode should be avoided if possible, as it increases processing time and generates a greater amount of traffic.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.4 SE/IP/OSPF Submenu

NOTE: The OSPF parameters that are defined on the link interface (Ethernet port or WAN connection) are detailed with the Ethernet parameters, starting from page 12.

9.4.1 SE/IP/OSPF/GLOBAL Parameters

Router ID

Console	SNMP	Text-based Config
Router ID	ospfGlobalRouterId	[ospf] GlobalRouterID

A 32-bit number that uniquely identifies this NetPerformer from all other routers in the Autonomous System. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The Router ID may be used to determine the designated router if the Router Priority value of the LAN port on the NetPerformer is equal to that of another router being considered. The router with the highest Router ID value will become the designated router. Refer to the description of the *OSPF Router priority* parameter, given on page 13.

To ensure a unique value for each router, we suggest that you set the Router ID to one of the NetPerformer's IP interface addresses. For example, you could choose the largest or smallest IP address that has been assigned to the NetPerformer.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

Auto virtual link

Console	SNMP	Text-based Config
Auto virtual link	ospfGlobalAutoVLink	[ospf] GlobalAutoVLink

Indicates whether the NetPerformer will automatically configure and enable OSPF virtual links as they are required. An OSPF virtual link must be configured to restore backbone connectivity when the AS is divided into non-contiguous areas. When a virtual link is defined it becomes part of the backbone.

If this parameter is set to **YES**, the NetPerformer will define an OSPF virtual link wherever it finds that it must route OSPF data to another backbone router via a common non-backbone area (the virtual link's transit area). The NetPerformer is one endpoint of the virtual link. A particular virtual link is enabled or disabled depending on the current shortest-path tree for the transit area, with the NetPerformer itself as the root. To be able to configure virtual links the NetPerformer must be an area border router.

If this parameter is set to **NO**, you must define all required virtual links manually. Refer to Defining OSPF Virtual Links on page 19 for details.

NOTE: If you configure the NetPerformer to automatically control all OSPF virtual links, you cannot configure the virtual link password. If any virtual link requires a unique password, you must set the Auto Virtual Link parameter to **NO** and configure all virtual links manually.

Values:	NO, YES
Default:	NO

Rack area ID

Console	SNMP	Text-based Config
Rack area ID	ospfGlobalRackAreaId	[ospf GlobalRackAreaID

For Central Site Units (rack models) only.
This parameter identifies the area to which the backplane belongs. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte. All routing protocol packets originating from the backplane of the NetPerformer are labelled with the Rack Area ID.

The value of this parameter must be the same as the Area ID of the area to which the attached network belongs (see Defining OSPF Area Characteristics on page 17). If you are defining subnetted networks as separate areas, you can use the appropriate IP network number as the Rack Area ID.

NOTE: The default Rack Area ID, 000.000.000.000, indicates that the backplane of the NetPerformer is included in the OSPF backbone.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

Global area ID

Console	SNMP	Text-based Config
Global area ID	ospfGlobalGlobalAreaId	[ospf] GlobalGlobalAreaId

This parameter identifies the area to which the NetPerformer’s global Default IP Address belongs. This Global Area ID requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The value of this parameter must be the same as the Area ID of the area to which the attached network belongs (see Defining OSPF Area Characteristics on page 17). If you are defining subnetted networks as separate areas, you can use the appropriate IP network number as the Global Area ID.

NOTE: The default Global Area ID, 000.000.000.000, indicates that the NetPerformer's global Default IP Address is included in the OSPF backbone.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

9.4.2 SE/IP/OSPF/AREA Parameters

OSPF Area entry number

Console	SNMP	Text-based Config
OSPF Area entry number	ospfAreaEntry, ospfAreaIndex	[ospf] Area Entry, [ospf] AreaIndex

Enter the OSPF Area Entry Number on the console command line. For SNMP, select the *ospfAreaEntry* table and look under the *ospfAreaIndex* for the desired area.

Once you select an area, its Entry Number is displayed thereafter at the beginning of each line from the console.

Values: 1 - 10

Default: 1

Area ID

Console	SNMP	Text-based Config
Area ID	ospfAreaAreaId	[ospf} AreaAreaID

A 32-bit number that uniquely identifies this area in the AS. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

If the area represents a subnetted network, its Area ID can be the same as the IP network number of the subnetted network.

NOTE: Area ID 000.000.000.000 is automatically reserved for the backbone, and does not need to be configured as a separate area.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

Enable

Console	SNMP	Text-based Config
Enable	ospfAreaEnable	[ospf] AreaEnable

This parameter allows you to enable or disable this area. By default, all areas are enabled.

Set this parameter to **DISABLE** to disable the area. This may be useful if you want to temporarily remove an area from the AS without deleting it. Each area that has been defined on the NetPerformer must be disabled separately.

To re-enable a previously disabled area, set this parameter to **ENABLE**. When an area is enabled, its version of the OSPF routing algorithm will restart.

Values:	DISABLE, ENABLE
Default:	ENABLE

Authentication type

Console	SNMP	Text-based Config
Authentication type	ospfAreaAuthType	[ospf] AreaAuthType

This parameter sets the type of password authentication that will be used for this area. All OSPF packet exchanges are authenticated. Each area in the AS can be configured with a separate Authentication Type.

Set the Authentication Type to **NONE** to “disable” password authentication for this area. The authentication field in the OSPF header can contain anything, and is not examined on packet reception. This is the default value.

Set this parameter to **SIMPLE** to “enable” password authentication. The actual password used must be defined with the OSPF Password parameter for the port or PVC. All packets sent on the network must carry this value in the OSPF header.

NOTE: If you select **SIMPLE**, only those routers that use the correct password will be able to participate in the routing domain of this area.

Values:	NONE, SIMPLE
Default:	NONE

Import AS extern

Console	SNMP	Text-based Config
Import AS extern	ospfAreaImportASExt	[ospf] AreaImportASExt

This parameter determines whether AS external advertisements will be flooded throughout the area.

Set Import AS External to **YES** to allow for AS external advertisements. This is the default value, and the only appropriate value for the backbone (Area ID 000.000.000.000) and for areas connected to the backbone by a virtual link.

Set this parameter to **NO** to exclude AS external advertisements from this area. This defines the area as a stub area. Within a stub area, routing to external destinations can be achieved through a default summary route only. The backbone cannot be configured as a stub area, and virtual links cannot be configured through stub areas.

NOTE: Inter-area and intra-area communications are not excluded from this area if you set the Import AS External parameter to **NO**. That is, routing tables will maintain information for the entire area, regardless of the value of this parameter. The only data that is blocked is EGP data, which is separate from OSPF link state data.

Values:	NO, YES
Default:	YES

Stub metric

Console	SNMP	Text-based Config
Stub metric	ospfAreaStubMetric	[ospf] AreaStubMetric

If the area has been defined as a stub area (Import AS External parameter set to **NO**), the Stub Metric parameter indicates the cost of the default summary link that the NetPerformer should advertise into the area. The Stub Metric value will actually be added to the total cost of the default summary route only if the NetPerformer is an area border router.

Values:	1 - 255
Default:	1

9.4.3 SE/IP/OSPF/RANGE Parameters

OSPF Range entry number

Console	SNMP	Text-based Config
OSPF Range entry number	ospfRangeEntry, ospfRangeIndex	[ospf] RangeEntry, [ospf] RangeIndex

Enter the OSPF Range Entry Number on the console command line. For SNMP, select the *ospfRangeEntry* table and look under the *ospfRangeIndex* for the desired range entry.

Once you select a range, its Entry Number is displayed thereafter at the beginning of each line from the console.

Values:	1 - 10
Default:	1

Range net

Console	SNMP	Text-based Config
Range net	ospfRangeNet	[ospf] RangeNet

A 32-bit number that identifies a group of subnets in this address range. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The Range Net, together with the Range Mask (see next parameter), describes the collection of IP addresses contained in the address range. Only one link summary advertisement is required for all subnets defined by the range, rather than one for each subnet included in the network.

Each address range should be configured carefully. Networks and hosts are assigned to an area depending on whether their addresses fall into one of the area’s defining address ranges. Routers may belong to multiple areas, depending on their attached networks’ area membership. When configuring the Range Net values for all address ranges, keep all subnetted networks in the same area.

NOTE: Make sure that the configured address ranges do not overlap.

Values:	000.000.000.000 - 255.255.255.255
Default:	1

Range mask

Console	SNMP	Text-based Config
Range mask	ospfRangeMask	[ospf] RangeMask

A 32-bit number associated with the Range Net to define a group of subnets in this address range. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

Any address mask can be used for this parameter. In other words, the Range Mask is not restricted to the natural address class mask for the Range Net address.

NOTE: Unlike the Subnet Mask parameter that defines an IP address, you cannot define the Range Mask by entering a number of bits at the console. You must enter the actual value of the Range Mask directly.

Values:	000.000.000.000 - 255.255.255.255
Default:	1

Enable

Console	SNMP	Text-based Config
Enable	ospfRangeEnable	[ospf] RangeEnable

This parameter enables or disables this range for the associated area. By default, all address ranges are enabled.

Set this parameter to **DISABLE** to disable this address range. This may be useful if you want to temporarily remove an address range from an area without deleting it. Each range that has been defined on the NetPerformer must be disabled separately. To re-enable a previously disabled range, set this parameter to **ENABLE**.

Values:	DISABLE, ENABLE
Default:	ENABLE

Status

Console	SNMP	Text-based Config
Status	ospfRangeStatus	[ospf] RangeStatus

A status indication is associated with each address range. The Status parameter can be set to either **ADVERTISE** or **DON'T ADV** (Don't Advertise). When set to **ADVERTISE**, a single route will be advertised for this address range on a summary link advertisement that is external to the area. This is the default Status value.

If you set the Status parameter to **DON'T ADV**, no route will be advertised for this address range. With an unadvertised range, you can intentionally hide certain networks from other areas.

Values: DON'T ADV, ADVERTISE

Default: ADVERTISE

Add to area

Console	SNMP	Text-based Config
Add to area	ospfRangeAddToArea	[ospf] RangeAddToArea

The Area ID that identifies the area to which this range belongs. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte. Select one of the areas that has already been defined with an Area ID (see page 12).

Values: 000.000.000.000 - 255.255.255.255

Default: 1

9.4.4 SE/IP/OSPF/VLINK Parameters

OSPF Virtual link entry number

Console	SNMP	Text-based Config
OSPF Virtual link entry number	ospfVLinkEntry, ospfVLinkIndex	[ospf] VLinkEntry, [ospf] VLinkIndex

Enter the OSPF Virtual Link Entry Number on the console command line. For SNMP, select the *ospfVLinkEntry* table and look under the *ospfVLinkIndex* for the desired virtual link.

Once you select a virtual link, its Entry Number is displayed thereafter at the beginning of each line from the console.

Values: 1 - 10

Default: 1

Transit area ID

Console	SNMP	Text-based Config
Transit area ID	ospfVLinkTransitAreaId	[ospf] VLinkTransitAreaId

The Area ID that identifies the virtual link's transit area. This parameter requires a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte.

The transit area is the common non-backbone area to which both endpoint routers are attached. Therefore, the transit area must contain the router defined by the Neighbor's Router ID (see next parameter). Its value is the Area ID of the area through which the virtual link passes. It cannot be a stub area (*Import AS External* parameter set to **NO**).

⚠ **Caution**: Since by definition the Transit Area ID is a non-backbone area, it must be changed from its default value, **000.000.000.000**, which specifies the backbone. Otherwise, the transit area will be considered part of the backbone, and the virtual link will not work.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

Neighbor’s router ID

Console	SNMP	Text-based Config
Neighbor’s router ID	ospfVLinkNeighborRtrId	[ospf] VLinkNeighborRtrId

The Router ID of the other endpoint of the virtual link. This parameter specifies a neighboring area border router that is attached to the virtual link’s transit area. It requires a 4-byte value in dotted decimal representation, with a maximum value of **255** for each byte.

Values: 000.000.000.000 - 255.255.255.255

Default: 000.000.000.000

Enable

Console	SNMP	Text-based Config
Enable	ospfVLinkEnable	[ospf] VLinkEnable

This parameter enables or disables this virtual link. By default, all virtual links are enabled.

Set this parameter to **DISABLE** to disable (turn off) the virtual link. This may be useful if you want to temporarily remove a virtual link without deleting it. Each virtual link that has been defined on the NetPerformer must be disabled separately. To re-enable a previously disabled virtual link, set this parameter to **ENABLE**.

Values: DISABLE, ENABLE

Default: ENABLE

Transit delay

Console	SNMP	Text-based Config
Transit delay	ospfVLinkTransitDelay	[ospf] VLinkTransitDelay

The estimated number of seconds required to transmit a Link State Update packet over the virtual link interface. Link state advertisements contained in the Link State Update packet will have their age incremented by this amount before transmission.

Values: 1- 360

Default: 1

Retransmit interval

Console	SNMP	Text-based Config
Retransmit interval	ospfVLinkRetransmitInt	[ospf] VLinkRetransmitInt

The number of seconds that elapse between retransmissions of link state advertisements. This parameter is used for adjacencies that belong to the virtual link interface, and for retransmissions of OSPF Database Description and Link State Request packets.

To avoid unnecessary retransmissions, set this parameter to a value that is higher than the expected round-trip delay.

Values: 1 - 360

Default: 5

Hello interval

Console	SNMP	Text-based Config
Hello interval	ospfVLinkHelloInt	[ospf] VLinkHelloInt

The length of time, in seconds, between Hello Packets sent on the virtual link interface. The value of the Hello Interval parameter is advertised in the Hello Packets sent out from this interface. This value must be the same on the neighboring router and all other routers attached to the same network.

If you set the Hello Interval to a short length of time, changes to the OSPF topological database will be detected more quickly. However, a short Hello Interval creates more OSPF routing protocol traffic.

Values: 1 - 360

Default: 10

Dead interval

Console	SNMP	Text-based Config
Dead interval	ospfVLinkDeadInt	[ospf] VLinkDeadInt

The length of time, in seconds, before neighboring routers declare a router down when they stop hearing its Hello Packets. The value of the Dead Interval parameter is advertised in Hello Packets sent out from the virtual link interface, and must be the same on all other routers having a connection with the network attached to this interface.

Set the Dead Interval to a multiple of the Hello Interval (described above).

Values: 1 - 2000

Default: 60

Authentication type

Console	SNMP	Text-based Config
Authentication type	ospfVLinkAuthType	[ospf] VLinkAuthType

Determines the type of authentication that will be performed to generate and/or verify OSPF protocol packets sent over this virtual link.

- **SIMPLE:** The *Password* (see page 21) is used for all OSPF packets sent on the network. All OSPF packets sent on this network must contain this value in the *Authentication* field of the OSPF header. The remaining contents of each OSPF packet are also verified with a checksum operation.

  ⚠ **Caution:** The **SIMPLE** authentication type does not protect the OSPF routing domain from passive attacks via the Internet, as anyone with physical access to the network can learn the *Password* that is used.

- **CRYPTOGRAPHIC:** A shared secret key, configured with the *Cryptographic auth. key* parameter (see page 21), is used to generate and/or verify a one-way *message digest* that is appended to each OSPF packet sent over this virtual link. **CRYPTOGRAPHIC** authentication also includes the use of:
  - The MD5 algorithm, to generate the message digest
  - A sequence number in the *Authentication* field of each OSPF packet, to protect against replay attacks.

NOTE: This is a more secure authentication method, as the *Cryptographic auth. key* itself is never sent over the virtual link.

- **NONE:** No authentication is performed on OSPF protocol packets.

Values: NONE, SIMPLE, CRYPTOGRAPHIC

Default: NONE

When the *Authentication type* parameter is set to **CRYPTOGRAHPHIC**, three additional parameters are displayed at the console (described below):

```
...
OSPF VLINK 1> Authentication type (def:NONE) ? CRYPTOGRAPHIC
OSPF VLINK 1> Cryptographic auth. ID (0-255,def:1) ?
OSPF VLINK 1> Cryptographic auth. key (def:) ?
```

When the *Authentication type* parameter is set to **SIMPLE**, one additional parameter is displayed at the console (described on page 21):

```
...
OSPF VLINK 1> Authentication type (def:NONE) ? SIMPLE
OSPF VLINK 1> Password (def:) ?
```

Cryptographic auth. ID

Console	SNMP	Text-based Config
Cryptographic auth. ID	ospfVLinkCryptoAuthId	[ospf] VLinkCryptoAuthId

For CRYPTOGRAPHIC Authentication type only
Identifies which algorithm and shared secret key will be used to create the message digest appended to an OSPF packet. This ID is particular to this virtual link only.

Values:	0 - 255
Default:	1

Cryptographic auth. key

Console	SNMP	Text-based Config
Cryptographic auth. key	ospfVLinkCryptoAuthKey	[ospf] VLinkCryptoAuthKey

For CRYPTOGRAPHIC Authentication type only
Defines the value of the shared secret key that will be used to create the message digest for OSPF packets sent over this virtual link.

Values:	maximum 16-character string
Default:	no value

Password

Console	SNMP	Text-based Config
Password	ospfVLinkPassword	[ospf] VLinkPassword

For SIMPLE Authentication type only
Defines the 64-bit value that will appear in the authentication field of all OSPF packets sent or received on this virtual link. The *Password* allows the authentication procedure to generate and/or verify the *Authentication* field in the OSPF header.

⚠ **Caution:** The value of the *Password* parameter must be the same as that configured on all other routers having a connection with the network attached to this virtual link. In other words, all routers in the same area must have the same password (or no authentication at all).

NOTE: Since the *Password* is configured separately for each interface, there can be a separate password for each network in the AS.

Values:	Maximum 8-character string
Default:	none

9.5 SE/IP/TIMEP Submenu

9.5.1 Negative time zone

Console	SNMP	Text-based Config
Negative time zone	timepTimeZoneSign	[timep] TimeZoneSign

Sets the time zone to a positive or negative zone with respect to Greenwich Mean Time (GMT). Set this parameter to **YES** (the default value) if the NetPerformer unit is located in a negative time zone. Otherwise, set it to **NO**.

Values:	NO, YES
Default:	YES

9.5.2 Time zone offset from GMT (min)

Console	SNMP	Text-based Config
Time zone offset from GMT (min)	timepTimeZone	[timep] TimeZone

Sets the number of minutes offset from GMT to the time zone where the NetPerformer unit is located. The value of this parameter must be positive, and it must be entered in minutes. Use Table 2 for assistance in setting the value for this parameter.

Table 2Time Zones and Offsets from Greenwich Mean Time

Time zone	Offset From GMT (min)	Locations
GMT -12:00	720	Eniwetok, Kwajalein
GMT -11:00	660	Midway Island, Samoa
GMT -10:00	600	Hawaii
GMT -09:00	540	Alaska
GMT -08:00	480	Pacific Time (US & Canada), Tijuana
GMT -07:00	420	Arizona, Mountain Time (US & Canada)
GMT -06:00	360	Central Time (US & Canada), Mexico City, Tegucigalpa, Saskatchewan
GMT -05:00	300	Bogota, Lima, Eastern Time (US & Canada), Indiana (East)
GMT -04:00	240	Atlantic Time (Canada), Caracas, La Paz
GMT -03:30	210	Newfoundland
GMT -03:00	180	Brasilia, Buenos Aires, Georgetown
GMT -02:00	120	Mid-Atlantic

Table 2Time Zones and Offsets from Greenwich Mean Time

Time zone	Offset From GMT (min)	Locations
GMT -01:00	60	Azores, Cape Verde Is.
GMT	GMT	Greenwich Mean Time, Dublin, Edinburgh, London, Monrovia, Casablanca
GMT +01:00	60	Berlin, Stockholm, Rome, Bern, Brussels, Vienna, Amsterdam, Lisbon, Warsaw, Paris, Madrid, Prague
GMT +02:00	120	Athens, Helsinki, Istanbul, Cairo, Eastern Europe, Harare, Pretoria, Israel
GMT +03:00	180	Baghdad, Kuwait, Nairobi, Riyadh, Moscow, St. Petersburg
GMT +03:30	210	Tehran
GMT +04:00	240	Abu Dhabi, Muscat, Tbilisi, Kazan, Volgograd
GMT +04:30	270	Kabul
GMT +05:00	300	Islamabad, Karachi, Ekaterinburg, Tashkent
GMT +05:30	330	Bombay, Calcutta, Madras, New Delhi, Colombo
GMT +06:00	360	Almaty, Dhaka
GMT +07:00	420	Bangkok, Jakarta, Hanoi
GMT +08:00	480	Beijing, Chongqing, Urumqi, Hong Kong, Perth, Singapore, Taipei
GMT +09:00	540	Tokyo, Osaka, Sapporo, Seoul, Yakutsk
GMT +09:30	570	Adelaide
GMT +10:00	600	Brisbane, Melbourne, Sydney, Guam, Port Moresby, Vladivostok, Hobart
GMT +11:00	660	Magadan, Solomon Is., New Caledonia
GMT +12:00	720	Fiji, Kamchatka, Marshall Is., Wellington, Auckland

Values:	0 - 720
Default:	300

9.5.3 Time server protocol

Console	SNMP	Text-based Config
Time server protocol	timepServerProtocol	[timep] ServerProtocol

Sets the protocol supported by the time server. Select **BOTH** if both UDP and TCP are available to the time server. If you set this parameter to **NONE**, the time server will be disabled.

Values:	NONE, UDP, TCP, BOTH
Default:	NONE

9.5.4 Time client protocol

Console	SNMP	Text-based Config
Time client protocol	timepClientProtocol	[timep] ClientProtocol

Sets the protocol used by the time client to communicate with the server. To activate the time client select either UDP or TCP, depending on your installation. If you set this parameter to **NONE**, the time client will be disabled.

Values:	NONE, UDP, TCP
Default:	NONE

9.5.5 Time client server IP address

Console	SNMP	Text-based Config
Time client server IP address	timepServerIpAddress	[timep] ServerIpAddress

Sets the IP address of the client's time server. It is a 4-byte value in dotted decimal representation, with a maximum value of 255 for each byte, for example 128.128.128.122. When this parameter is set to **000.000.000.000**, no Time Client Server IP Address is defined.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.5.6 Time client update interval (min)

Console	SNMP	Text-based Config
Time client update interval (min)	timepClientUpdate-Interval	[timep] ClientUpdateInterval

Determines the interval, in minutes, at which the client will try to synchronize its real-time clock with the time server.

Values: 1 - 65534

Default: 1440

9.5.7 Time client UDP timeout (s)

Console	SNMP	Text-based Config
Time client UDP timeout (s)	timepClientUdpTimeout	[timep] ClientUdpTimeout

Defines the wait time, in seconds, before the UDP client will retransmit a clock synchronization request.

This parameter also plays a role in determining the length of the interval, in seconds, at which the NetPerformer will interrogate the time server at startup. This interval is defined as:

$$30 + (\mathit{UDP\_timeout} \text{ X } \mathit{UDP\_retransmissions})$$

The NetPerformer unit will try to interrogate the server up to 20 times. If after 20 attempts the client has not yet synchronized, the unit reverts to its normal interrogation interval. This avoids wasting bandwidth if the time server is not online.

Values: 1 - 255

Default: 20

9.5.8 Time client UDP retransmissions

Console	SNMP	Text-based Config
Time client UDP retransmissions	TimepClientUdpRetransmissions	[timep] ClientUdpRetransmissions

Defines the number of retransmissions that are allowed before a fail is declared.

This parameter also plays a role in determining the length of the interval, in seconds, at which the NetPerformer will interrogate the time server at startup. This interval is explained in the description of the Time Client UDP Timeout parameter, above.

Values: 0 - 255

Default: 3

9.6 SE/IP/SNMP Submenu

9.6.1 Get community

Console	SNMP
Get community	(not available)

Community for Get Requests. This parameter determines the community that the SNMP agent accepts for all Get Requests. It can be configured using a direct or dialup console terminal only.

Values:	Maximum 16-character string: **A-Z**, **0-9**, (space)
Default:	PUBLIC

9.6.2 Set community

Console	SNMP
Set community	(not available)

Community for Set Request. This parameter determines the community that the SNMP agent accepts for the Set Request. It can be configured using a direct or dialup console terminal only.

Values:	Maximum 16-character string: **A-Z**, **0-9**, (space)
Default:	PUBLIC

9.6.3 Trap community

Console	SNMP
Trap community	(not available)

Community for Traps. This parameter determines the community that the SNMP agent uses when sending Traps to the manager. It can be configured using a direct or dialup console terminal only.

NOTE: Traps are sent only to those SNMP managers that are specified in the global SNMP trap IP addresses. You can define up to four IP addresses as the destinations for SNMP traps, using the Setup Global menu. Refer to the chapter *Global Functions* in the *Quick Configuration* fascicle of this document series.

Values:	Maximum 16-character string: **A-Z**, **0-9**, (space)
Default:	PUBLIC

9.7 SE/IP/DNS Submenu

9.7.1 Primary server address

Console	SNMP	Text-based Config
Primary server address	ipDnsIpAddress1	[Dns] IpAddress1

IP address of the primary DNS server. If you leave the *Primary server address* at its default value (**000.000.000.000**) it is considered **not defined**.

NOTE: At least the *Primary server address* must be defined for DNS address resolution to work.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.7.2 Secondary server address

Console	SNMP	Text-based Config
Secondary server address	ipDnsIpAddress2	[Dns] IpAddress2

IP address of the secondary DNS server. If you leave the *Secondary server address* at its default value (**000.000.000.000**) it is considered **not defined**.

Values:	000.000.000.000 - 255.255.255.255
Default:	000.000.000.000

9.7.3 Ignore DNS time to live

Console	SNMP	Text-based Config
Ignore DNS time to live	ipDnsIgnoreTtl	[Dns] IgnoreTtl

Set the *Ignore DNS time to live* parameter to **YES** if you want the NetPerformer to disregard any Time To Live (TTL) information provided by the DNS server. In this case, the DNS entries will not age or expire.

Values:	NO, YES
Default:	NO

10

SE/BRIDGE Configuration Parameters

10.1 Bridge Parameters on NetPerformer Products Running V10.X

10.1.1 Enabled

Console	SNMP	Text-based Config
Enabled	bridgeEnable	[bridge] Enable

Enable or disable bridge functions. This parameter determines whether the configured bridge will be used to transmit data across the network. If you enable the bridge (**YES**), the NetPerformer transfers information between two LAN segments. If you select **NO**, the bridge connection is disabled.

Values:	NO, YES
Default:	NO

10.1.2 Spanning Tree Protocol active

Console	SNMP	Text-based Config
Spanning Tree Protocol active	bridgeStpEnable	[bridge] StpEnable

This parameter defines whether the Spanning Tree protocol will be enabled. The NetPerformer uses the spanning tree algorithm to decide how to forward frames and how to propagate broadcast packets so that only one copy of a broadcast frame is delivered to each LAN. STP ensures that only one active route is used at one time during transparent bridging.

⚠ **Caution**: **STP must be enabled on an Ethernet network**. Without it, broadcast packets will be forwarded in both directions at the same time, which will produce disastrous results.

Values:	NO, YES
Default:	NO

10.1.3 Aging time (s)

Console	SNMP	Text-based Config
Aging time (s)	bridgeAgingTime-s	[bridge] AgingTime-s

The allotted time, in seconds, before inactive addresses will be deleted from the routing table. In transparent bridging, the routing table contains a combination of fixed and learned addresses. For the learned addresses, if a station has not been heard from for the Aging Time period, its entry can be deleted.

This process keeps the size of the routing table to a manageable level, and minimizes the amount of memory and processing required to search it.

Values:	10 - 1000000
Default:	300

10.1.4 Hello messages interval (s)

Console	SNMP	Text-based Config
Hello messages interval (s)	bridgeHelloTime-s	[bridge] HelloTime-s

The interval, in seconds, between Hello messages that the NetPerformer transmits to all LAN segments to which it is connected.

Hello messages carry timestamp information as well as information concerning the current root bridge. From this information the NetPerformer is able to determine the current network topology and compute the shortest delay paths to destination devices. Automatic single-route broadcast uses the Hello message to detect when bridges enter and leave the network.

Values:	1 - 10
Default:	2

10.1.5 Hello frames maximum age (s)

Console	SNMP	Text-based Config
Hello frames maximum age (s)	bridgeMaxAge-s	[bridge] MaxAge-s

The maximum time that the bridge can wait for reception of a Hello frame. If this bridge is the network root bridge, the Hello Frames Maximum Age value will be coded in all Hello messages that the NetPerformer transmits.

Values:	6 - 40
Default:	20

10.1.6 Forward delay (s)

Console	SNMP	Text-based Config
Forward delay (s)	bridgeForwardDelay-s	[bridge] ForwardDelay-s

The wait time, in seconds, that the NetPerformer spends in the Learning state before moving to the Forwarding state. In the Learning state, the NetPerformer is building address tables and participating in the spanning tree algorithm, but is not forwarding frames. In the Forwarding state, the NetPerformer forwards frames in addition to its other bridge activities.

Values:	4 - 30
Default:	15

10.1.7 Bridge priority

Console	SNMP	Text-based Config
Bridge priority	bridgePriority	[bridge] Priority

Priority assigned to this bridge for the Spanning Tree algorithm. The NetPerformer uses this parameter to determine the network topology.

Values:	0 - 65535
Default:	32767

10.2 Bridge Parameters on Legacy NetPerformer Products

Some legacy NetPerformer products support Token-Ring LAN traffic, and can bridge this traffic from a remote legacy NetPerformer. The following parameters are found on these products in addition to the parameters described above:

10.2.1 LAN type

Console	SNMP
LAN type	bridgeLanType

For a NetPerformer that has no LAN interface installed.

The type of LAN data that must be bridged via the NetPerformer. Use this parameter if the remote NetPerformer supports a LAN whose traffic must be bridged through the local NetPerformer. This may include legacy NetPerformer models supporting a Token-Ring LAN. For a description of the various Ethernet protocol types available, refer to the Ethernet port *Protocol* parameter on page 2.

NOTE: The *LAN type* parameter is required only when the bridge *Enabled* parameter is set to **YES**.

Values: ETH AUTO, ETH 802.3, ETH V2, TKN-RING

Default: ETH AUTO

10.2.2 Bridge number

Console	SNMP
Bridge number	bridgeNumber

The bridge number, required for source routing of Token-Ring data via the NetPerformer. This number represents the bridge between two links on a Token-Ring network. You must set the bridge number if the NetPerformer acts as an intermediary bridge to a remote legacy NetPerformer that supports a Token-Ring LAN.

NOTE: The value of the *Bridge number* parameter on the local unit must be different from the value of the Bridge Number parameter on the remote unit.

Values: 00 - 0F

Default: 01

10.2.3 Span mode

Console	SNMP
Span mode	bridgeSteSpan

Span mode for STE (Spanning Tree Explorer) frames, required when Token-Ring data is bridged via the NetPerformer. This parameter determines how the port will behave when presented with STE frames (Single Route Broadcast only). You must set the STE Span Mode if the NetPerformer acts as an intermediary bridge to reach a remote legacy NetPerformer that supports a Token-Ring LAN.

The possible STE SPAN settings are:

- **AUTO**: The NetPerformer uses the Spanning Tree network topology to decide whether to forward STE frames. To operate properly, the Spanning Tree protocol must be enabled with the STP Enable parameter.
- **DISABLE**: All STE frames received are discarded by the NetPerformer.
- **FORCED**: All STE frames received are forwarded by the NetPerformer.

Values:	AUTO, DISABLE, FORCED
Default:	AUTO

10.2.4 Maximum hop

Console	SNMP
Maximum hop	bridgeMaxHop

The maximum number of bridges that a broadcast frame can cross on the way to its destination. The NetPerformer discards any frame with a routing information field that exceeds this limit. This parameter is used for source routing of Token-Ring data via the NetPerformer. You must specify the Maximum Hop if the NetPerformer acts as an intermediary bridge to reach a remote legacy NetPerformer that supports a Token-Ring LAN.

Before using a value different from the default value, consider the impact of this parameter on normal traffic flow. Also take into account the requirements of a backup situation if a bridge failure should occur. The connectivity requirements of your network should not be limited by a Maximum Hop value that is too low.

NOTE: If two NetPerformers are connected via a WAN link, the value of the *Maximum Hop* parameter on the local unit must be the same as the value of the *Maximum Hop* parameter on the remote unit.

Values:	0 - 7
Default:	7

11

DISPLAY Command Statistics

11.1 Ethernet Port Statistics

11.1.1 DC/PORT/ETH

Transmitter rate

Console	SNMP
Transmitter rate	statIflanMeanTx-kbps

The average (M) or highest (P) throughput level for transmissions sent to the Ethernet LAN, in Kbps.

Receiver rate

Console	SNMP
Receiver rate	statIflanMeanRx-kbps

The average (M) or highest (P) throughput level for data received from the Ethernet LAN, in Kbps.

11.1.2 DS/PORT/ETH

Protocol

Console	SNMP
Protocol	statIflanProtocol

The protocol used on the Ethernet port. The displayed protocol may be **ETHERNET** or **OFF**.

Interface

Console	SNMP
Interface	statIflanEth-Interface

The connection interface used on the Ethernet port. The physical connection is either **10BASET** for the 10Base-T connection, or **10BASE5** for the 10Base-5 connection.

Speed

Console	SNMP
Speed	statIflanSpeed

The speed of the Ethernet connection. The speed displayed is the fixed Ethernet speed, **10M**.

Duplex mode

Console	SNMP
Duplex mode	statIflanDuplexMode

The duplex mode of the Ethernet connection, either **HALF** or **FULL**.

Operating mode

Console	SNMP
Operating Mode	statIflanOperatingMode

This includes Link Integrity and the receive wire status of the Ethernet connection. These are indicated by:

- **L**: Link Integrity detected
- **P**: 10BASE-T receive wires are reversed

When Link Integrity is not detected, the NetPerformer displays a dash [**-**] in the first (**L**) position. When the 10BASE-T receive wires are not reversed, the NetPerformer displays a dash in the second (**P**) position. Both the Link Integrity and receive wire status are combined in the SNMP variable *statIflanOperatingMode*.

State

Console	SNMP
State	statIflanConnectionStatus

The logical connection status of the Ethernet port. This may be any of the following:

- **RESET:** The Ethernet controller is currently in reset state for loading the software and initializing the interface chip.
- **CLING:** Closing; the NetPerformer is currently trying to get out of the LAN.
- **CLOSE:** Closed; the NetPerformer is no longer on the LAN.
- **OPING:** Opening; the NetPerformer is currently trying to get on the LAN. The NetPerformer will stay in this state if it is unable to get on the LAN.
- **OPEN:** Open; the NetPerformer is now on the LAN.

Network address

Console	SNMP
Network address	statSystemNa

The address that the NetPerformer uses to send data on the LAN. This may be the burned-in address (**BIA**) or the address configured using the MAC Address parameter in the Ethernet port configuration (see Configuring the Ethernet LAN Port on page 2).

Burned in address

Console	SNMP
Burned in address	statSystemBia

The burned-in Ethernet address for the NetPerformer.

Number of deferred transmission

Console	SNMP
Number of deferred transmission	statIflanEth-DeferredTrans

The number of frames deferred before transmission on the Ethernet port. These frames were delayed on the first transmission attempt because the LAN was busy. Frames that are involved in collisions or deferred during a later transmission attempt are not included.

Number of collision frames

Console	SNMP
Number of collision frames	statIflanEth-AllCollision

The number of all types of collision frames on the Ethernet port. These include the number of single collision frames, two collision frames and three and more collision frames.

11.1.3 DE/PORT/ETH

Number of excessive collisions

Console	SNMP
Number of excessive collisions	statIflanEth-ExcessiveCollision

The number of frames aborted during transmission due to an excessive number of collisions. These are frames that have not been transmitted successfully. The maximum value for this counter is **9999**.

Number of late TX collision errors

Console	SNMP
Number of late TX collision errors	statIflanEth-TxLateColl

The number of frames transmitted with a late collision, that is, the number of times that a collision was detected later than 512 bits into the transmitted packet.

Number of underruns

Console	SNMP
Number of underruns	statIflanUnderruns

An underrun indicates a transmission error for an incomplete frame. The Underruns counter is incremented when a frame currently in transmission has been aborted because the end of the frame was not received on time.

Number of late RX collision errors

Console	SNMP
Number of late RX collision errors	statIflanEth-RxLateColl

The number of frames received with a late collision, that is, the number of times that a collision was detected later than 512 bits into the received packet.

Number of overruns

Console	SNMP
Number of overruns	statIflanOverruns

This counter displays the number of overruns, indicating the number of times that an overflow occurred on reception of a frame.

Number of busy conditions

Console	SNMP
Number of busy conditions	statIflanEth-BusyConditions

This counter displays the number of frames received and discarded due to a lack of buffers.

Number of FCS errors

Console	SNMP
Number of FCS errors	statIflanEth-FrameCheckSeq

This counter displays the number of frames received on the Ethernet port with frame check sequence (FCS) errors.

Number of alignment errors

Console	SNMP
Number of alignment errors	statIflanEth-Align

This counter displays the number of incomplete frames received on the Ethernet port that did not pass the CRC check.

Number of carrier sense errors

Console	SNMP
Number of carrier sense errors	statIflanEth-CarrierSense

This counter indicates the number of frames transmitted with carrier sense errors: either the carrier sense signal from the physical layer interface was not asserted, or it was deasserted during transmission of the frame without collision.

Number of bad frames

Console	SNMP
Number of bad frames	statIflanBadFrames, statIflanBadFlags

The number of bad frames that have been detected on the Ethernet port. The Bad Frames counter (*statIflanBadFrames* in SNMP) indicates the number of bad frames received and rejected, and the flags indicate the types of errors that have occurred. The Bad flags (*statIflanBadFlags* in SNMP) are displayed with a six-character field, of which only the **S** and **B** flags are used for an Ethernet port:

- **S**: Overrun; overflow on reception
- **B**: Bad CRC; frame contains a bad CRC

Number of retries

Console	SNMP
Number of retries	statIflanRetries

The number of retries that have occurred on the Ethernet port. This error counter is incremented when there is a retransmission between the NetPerformer and the LAN.

Number of restarts

Console	SNMP
Number of restarts	statIflanRestart

This counter is incremented every time an error on a PVC caused the Ethernet port to restart. It indicates how many times the LAN interface has had to be resynchronized due to errors occurring on the PVC connections.

11.2 Bridge Port Statistics (DB)

11.2.1 Global Bridge Statistics

Address discard

Console	SNMP
Address discard	statBridgeBridgeAddressDiscard

This counter indicates the number of times that an address entry in the filtering database has been removed to make room for a new address. If this counter increases rapidly, the filtering database is too small for the number of addresses (stations) in the network.

Transparent frame discard

Console	SNMP
Transparent frame discard	statBridgeBridgeFrameDiscard

This counter indicates the number of times that a frame has not been bridged because its destination is local.

Designated root

Console	SNMP
Designated root	statBridgeBridgeDesignatedRoot

Identifier of the designated root bridge. This is an 8-byte hexadecimal label composed of the bridge priority level (the first 2 bytes) and the address of the adapter connecting the bridge to the LAN segment with the lowest LAN segment number. The root bridge has the lowest bridge identifier of all bridges in the network, and is at the top of the spanning tree. It is also the bridge that sends the “Hello” message to detect when other bridges enter and leave the network. The root bridge usually carries the greatest amount of traffic, since it connects the two halves of the network together.

Root cost

Console	SNMP
Root cost	statBridgeBridgeRootCost

From each NetPerformer there are potentially many different paths to the root bridge. The root cost is the lowest path cost, that is, the shortest relative path length to the root bridge. When the root cost is displayed as zero (0), either this NetPerformer is the root bridge or there is no direct path from this unit to the root bridge.

Root port

Console	SNMP
Root port	statBridgeBridgeRootPort

This is the port with the root cost. In other words, it is the port in the direction of the least path cost to the root bridge. The root port is identified by the index of the bridge port. **NONE** indicates that this NetPerformer is the root bridge.

NOTE: The actual destination of each bridge port can be identified by examining the Destination statistic, described in Bridge Port Statistics on page 8.

Frame filtered

Console	SNMP
Frame filtered	statBridgeBridgeFrameFiltered

This counter indicates the number of frames that have not been forwarded because of a filter configured on the NetPerformer.

Frame timeout discard

Console	SNMP
Frame timeout discard	statBridgeBridgeFrameTimeout

This counter indicates the number of frames that have been discarded because the elapsed time exceeded the transit delay. The transit delay is configured using the global *Transit Delay* parameter. Refer to the chapter *Global Functions* in the *Quick Configuration* fascicle of this document series.

11.2.2 Bridge Port Statistics

Destination

Console	SNMP
Destination	statBridgePortDestination

Identifier of the destination of this bridge port. When the bridge port is the LAN, the displayed destination is given as **LOCAL LAN**. For the other bridge ports the displayed destination is the name of the attached remote unit.

Port index

Console	SNMP
Port index	statBridgePortIndex

A unique identifier of the bridge port. This value is used by the Spanning Tree Protocol, and is provided as a statistic for troubleshooting purposes by Memotec personnel.

State

Console	SNMP
State	statBridgePortState

The current state of the port in the bridge topology. This state can be one of the following:

- **DISABLED**: Note participating in the bridge topology
- **BLOCKING**: Participation limited to ensuring that another bridge forwards frames onto the network segment
- **LISTENING**: Participates in the spanning tree algorithm
- **LEARNING**: Participates in the spanning tree algorithm and builds address tables
- **FORWARD**: Participates in the spanning tree algorithm, builds address tables and forwards frames

Designated root

Console	SNMP
Designated root	statBridgePortDesignatedRoot

Identifier of the designated root for this port, that is, the bridge that this port considers to be the root bridge of the network. The bridge ID is an 8-byte hexadecimal label composed of the bridge priority level (the first 2 bytes) and the address of the adapter connecting the bridge to the LAN segment (or group) with the lowest number.

Designated cost

Console	SNMP
Designated cost	statBridgePortDesignatedCost

The cost of the path to the root bridge provided by the designated port (see Designated Port parameter, below). The designated port connects to the same LAN as the port being examined.

Designated bridge

Console	SNMP
Designated bridge	statBridgePortDesignatedBridge

This is an 8-byte hexadecimal label composed of the bridge priority level (the first 2 bytes) and the address of the adapter connecting the bridge to the LAN segment (or group) with the lowest number. For each LAN segment (or group), only one bridge is in the forwarding state at any one time. This is the designated bridge for that LAN (or group). All other bridges in the network are in the blocking state, and do not forward frames or build address tables.

Designated port

Console	SNMP
Designated port	statBridgePortDesignatedPort

The port that is considered the designated port for this LAN (the LAN to which the port being examined is connected). All LAN data is sent via this port to the root bridge. The designated port ID is a 2-byte hexadecimal label, where the high byte indicates the port priority and the low byte indicates the port number.

Transparent frame in

Console	SNMP
Transparent frame in	statBridgePortTrspFrameIn

The number of frames received on the port for a transparent bridge.

Transparent frame out

Console	SNMP
Transparent frame out	statBridgePortTrspFrameOut

The number of frames forwarded from the port for a transparent bridge.

11.2.3 Bridge Statistics on Legacy NetPerformer Products

The following bridge port statistics are displayed on a legacy NetPerformer unit when one of the following conditions is met:

- The bridge *LAN Type* parameter has been set to **TKN-RING**
- The legacy NetPerformer is equipped with a Token-Ring LAN interface.

These statistics are displayed for a source routing bridge only.

Specifically routed frame in

Console	SNMP
Specifically routed frame in	statBridgePortTr-SpecRteFrameIn

The number of Token-Ring frames received on the port that contain a Routing Information field (other than broadcast frames).

Specifically routed frame out

Console	SNMP
Specifically routed frame out	statBridgePortTr-SpecRteFrameOut

The number of Token-Ring frames forwarded from the port that contain a Routing Information field (other than broadcast frames).

All-route broadcast frame in

Console	SNMP
All-route broadcast frame in	statBridgePortTr-AllRteFrameIn

The number of all-route broadcast Token-Ring frames received on the port.

All-route broadcast frame out

Console	SNMP
All-route broadcast frame out	statBridgePortTr-AllRteFrameOut

The number of all-route broadcast Token-Ring frames forwarded from the port.

Single-route broadcast frame in

Console	SNMP
Single-route broadcast frame in	statBridgePortTr-SingleRteFrameIn

The number of single-route broadcast Token-Ring frames received on the port.

Single-route broadcast frame out

Console	SNMP
Single-route broadcast frame out	statBridgePortTr-SingleRteFrameOut

The number of single-route broadcast Token-Ring frames forwarded from the port.

Segment mismatch discards

Console	SNMP
Segment mismatch discards	statBridgePortTr-SegmentMismatch

The number of single-route or all-route broadcast Token-Ring frames that have been discarded by this port because the Routing Information field contained an invalid adjacent segment value.

Duplicate segment discards

Console	SNMP
Duplicate segment discards	statBridgePortTr-SegmentDuplicate

The number of non-broadcast Token-Ring frames that have been discarded by this port because the Routing Information field contained the same segment identifier more than once.

Hop count exceed discard

Console	SNMP
Hop count exceed discard	statBridgePortTr-HopCntExceeded

The number of single-route or all-route broadcast Token-Ring frames that have been discarded by this port because the Routing Information field reached the maximum number of hops permitted.

Frame length exceeded

Console	SNMP
Frame length exceeded	statBridgePortTr-FrmLngExceeded

The number of non-broadcast Token-Ring frames that have been discarded by this port because the frame length is greater than the maximum length permitted.

11.3 IP Connection Statistics

11.3.1 DR/IP/UNICAST/RIP

DESTINATION

Console	SNMP
DESTINATION	MIB II - *mgmt* - ipRouteTable

The routing destination, identified by its IP address. All destinations are listed in numeric order.

VAL

Console	SNMP
VAL	MIB II - *mgmt* - ipRouteTable

indicates whether the connection to this destination is currently active (**Y**) or inactive (**N**).

COST

Console	SNMP
COST	MIB II - *mgmt* - ipRouteTable

The hop count, or number of NetPerformers that must be passed over to reach the destination. When at 0, it indicates a direct connection to the destination. Its value is usually between 1 and 15 for an indirect connection. When at 16, it indicates that the destination is unreachable.

INTF

Console	SNMP
INTF	MIB II - *mgmt* - ipRouteTable

The interface used to reach the destination. On the console display, the values in this column correspond to physical connections on the NetPerformer:

- **WAN*x*:** PVCR port. The value *x* is equivalent to the serial port number
- **LAN*x*:** LAN port (Ethernet). The value *x* is equivalent to the LAN port number
- **PVC*x*:** Frame Relay or ATM PVC. The value *x* is equivalent to the PVC number
- **FW*xy*:** Firewire (backplane) connection on a SDM-9500 rackmount model. The value *x* is the Rack ID and *y* is the Slot ID.

NEXT HOP

Console	SNMP
NEXT HOP	MIB II - *mgmt* - ipRouteTable

The next unit to be reached on the path to the final destination. On the IP routing table, the next hop is identified by the IP address of the router that will be used to send the IP frame.

AGE

Console	SNMP
AGE	MIB II - *mgmt* - ipRouteTable

The aging time since this destination was entered on the routing table. For a valid connection (**VALID = YES**), the aging time is incremented until an update is received for this entry. It will be displayed in seconds (s), minutes (m), hours (h), days (d) or years (y), depending on how long the entry has been in the table.

If a destination is reached through a direct connection, its AGE value will remain at 0 seconds, and will increase only when the port is closed. For an invalid connection (**VALID = NO**), the AGE value will increment for 120 seconds, at which time it is removed from the routing table.

MASK

Console	SNMP
MASK	MIB II - *mgmt* - ipRouteTable

The subnet mask associated with the IP address for this entry.

TYPE

Console	SNMP
TYPE	MIB II - *mgmt* - ipRouteTable

The destination type. The following types may be displayed:

- **NET**: a network,
- **SUB**: a subnet,
- **HOST**: a host,
- **DGTW**: a default gateway,
- **RNGE**: a range, that is, an entry that can be recognized by OSPF.

PROT

Console	SNMP
PROT	MIB II - *mgmt* - ipRouteTable

The type of routing used to reach the destination. The following protocols may be displayed:

- **LOCAL**: when the destination is the IP address of a NetPerformer interface,
- **OSPF**: when the destination is reached through OSPF routing,
- **RIP**: when the destination is reached through IP RIP routing,
- **STATIC**: when the destination is reached via a static route,
- **UNK**: unknown protocol, for example, an invalid static route.

11.3.2 DR/IP/UNICAST/MULTIHOMED

NOTE: Many of the statistics on this routing table are explained in the preceding section (see page 13).

TTL

Console	SNMP
TTL	MIB II - *mgmt* - ipRouteTable

On multihomed routing table only.
The Time To Live (TTL) for this entry in the routing table. Unlike the Age statistic on the IP routing table, which is incremented in seconds, the TTL statistic is decremented in minutes from an initial value of 10 minutes. When the TTL value reaches 0, the entry is removed from the routing table.

11.3.3 DR/IP/UNICAST/SOURCE-STATIC

11.3.4 DR/IP/MULTICAST

ADDRESS

Console	SNMP
ADDRESS	MIB II - *mgmt* - ipRouteTable

The IP address of the IP multicast group.

INTRF

Console	SNMP
INTRF	MIB II - *mgmt* - ipRouteTable

The physical interface on the NetPerformer that is used to reach this group.

AGE

Console	SNMP
AGE	MIB II - *mgmt* - ipRouteTable

The aging time, in seconds, since the entry for this group was entered on the routing table.

TTL

Console	SNMP
TTL	MIB II - *mgmt* - ipRouteTable

The Time To Live (TTL) for this entry. The TTL may be between 0 and 260 seconds.

LAST REPORTER

Console	SNMP
LAST REPORTER	MIB II - *mgmt* - ipRouteTable

The IP address of the client which most recently declared itself a member of the group.

11.3.5 DC/IP

In received

Console	SNMP
In received	ipInReceives (MIB II)

The total number of input datagrams received from the lower level. This includes datagrams that were discarded due to errors or a lack of buffer space.

In header errors

Console	SNMP
In header errors	ipInHdrErrors (MIB II)

The number of input datagrams that were discarded because of an error in the IP header. IP header errors include bad checksum, time-to-live expired, formatting and processing errors.

In address errors

Console	SNMP
In address errors	ipInAddrErrors (MIB II)

The number of input datagrams that were discarded because of an error in the destination IP address. The address may be invalid because it is undefined, belongs to an unsupported class or cannot be forwarded by the NetPerformer.

In unknown protocols

Console	SNMP
In unknown protocols	ipInUnknownProtos (MIB II)

The number of input datagrams that were discarded because the protocol is unknown or not supported.

In discarded

Console	SNMP
In discarded	ipInDiscards (MIB II)

The number of input datagrams that were received without error but were discarded during processing. The most common reason for these discards is a lack of buffer space. This counter does not include the number of input datagrams that were discarded during reassembly.

In delivered

Console	SNMP
In delivered	ipInDelivers (MIB II)

The total number of input datagrams that were successfully delivered to the higher levels (UDP, TCP, OSPF).

Reasm timeout

Console	SNMP
Reasm timeout	ipReasmTimeout (MIB II)

The number of times that the Reassembly Timer expired. This timer specifies the maximum time, in seconds, during which the NetPerformer can hold received IP fragments for reassembly.

Reasm requested

Console	SNMP
Reasm requested	ipReasmReqds (MIB II)

The number of times that IP fragments were received for reassembly.

Reasm ok

Console	SNMP
Reasm ok	ipReasmOKs (MIB II)

The number of times that datagrams were successfully reassembled from IP fragments.

Reasm failed

Console	SNMP
Reasm failed	ipReasmFails (MIB II)

The number of times the reassembly of IP fragments failed. Failures may be due to timeouts or errors detected. Since this counter is incremented once for each failure, it does not necessarily reflect the actual number of IP fragments that were discarded.

Forwarded datagrams

Console	SNMP
Forwarded datagrams	ipForwDatagrams (MIB II)

The number of input datagrams that were forwarded to the next hop on the route to their final destination.

Out requested

Console	SNMP
Out requested	ipOutRequests (MIB II)

The number of IP datagrams received from local channels for transmission to the next hop. This counter does not include forwarded datagrams.

Out discarded

Console	SNMP
Out discarded	ipOutDiscards (MIB II)

The number of output datagrams (including forwarded datagrams) that were discarded during processing. The most common reason for these discards is a lack of buffer space.

Out no routes

Console	SNMP
Out no routes	ipOutNoRoutes (MIB II)

The number of output datagrams (including forwarded datagrams) that were discarded because a route to the requested destination could not be found or was unavailable.

Fragmentation ok

Console	SNMP
Fragmentation ok	ipFragOKs (MIB II)

The number of times that IP datagrams were successfully fragmented by the NetPerformer.

Fragmentation failed

Console	SNMP
Fragmentation failed	ipFragFails (MIB II)

The number of times the fragmentation of IP datagrams failed. This provides the number of datagrams that were discarded because they could not be fragmented by the NetPerformer.

Fragments created

Console	SNMP
Fragments created	ipFragCreates (MIB II)

The number of times that the NetPerformer created IP fragments during the fragmentation process.

Out DF discarded

Console	SNMP
Out DF discarded	(not available)

The number of discarded output datagrams that the NetPerformer could not fragment because the *Don’t Fragment* bit was raised.

RIP frames discarded

Console	SNMP
RIP frames discarded	(not available)

The number of discarded RIP frames for all interfaces on the NetPerformer. RIP frames may be discarded when IP RIP is disabled in one or both directions, or when RIP Version 2 frames with authentication are received on an interface running RIP Version 1.

11.3.6 PING

Time

Console	SNMP
Time	(not available)

The elapsed time since the beginning of the test.

Transmit

Console	SNMP
Transmit	(not available)

The number of ping requests transmitted.

Receive

Console	SNMP
Receive	(not available)

The number of ping responses received.

Timeout

Console	SNMP
Timeout	(not available)

The number of times a ping request was sent and no response was received within the ping timeout.

Error

Console	SNMP
Error	(not available)

The number of times a response was invalid due to bad frames, line failures or other error situations.

MinResp

Console	SNMP
MinResp	(not available)

The fastest response time, in milliseconds.

MaxResp

Console	SNMP
MaxResp	(not available)

The slowest response time, in milliseconds.

MeanResp

Console	SNMP
MeanResp	(not available)

The average response time, in milliseconds.

11.4 BOOTP Statistics

11.4.1 DC/BOOTP

Number of BOOTREQUEST frames received

Console	SNMP
Number of BOOTREQUEST frames received	statBootpNbRequestReceived

The total number of BOOTREQUEST frames received by the NetPerformer.

Number of BOOTREQUEST frames sent

Console	SNMP
Number of BOOTREQUEST frames sent	statBootpNbRequestSend

The total number of BOOTREQUEST frames transmitted by the NetPerformer.

Number of BOOTREPLY frames received

Console	SNMP
Number of BOOTREPLY frames received	statBootpNbReplyReceived

The total number of BOOTREPLY frames received by the NetPerformer.

Number of BOOTREPLY frames sent

Console	SNMP
Number of BOOTREPLY frames sent	statBootpNbReplySend

The total number of BOOTREPLY frames transmitted by the NetPerformer.

11.4.2 DE/BOOTP

Reply with invalid giaddr

Console	SNMP
Reply with invalid giaddr	statBootpReplyWithInvalidGiaddr

The number of BOOTREPLY frames received with an invalid Giaddr (gateway IP address) field. The Giaddr field is invalid when its value does not correspond to any IP address configured on the NetPerformer ports. All frames with an invalid Giaddr are flushed.

Hops limit exceeded

Console	SNMP
Hops limit exceeded	statBootpHopsLimitExceed

The number of BOOTREQUEST frames received with a hops count greater than the value of the BOOTP Maximum Hops parameter. All frames that exceed the hops limit are flushed.

Request received on port bootpc

Console	SNMP
Request received on port bootpc	statBootpRequestReceivedOn-PortBootpc

The number of BOOTREQUEST frames received on the BOOTPC (BOOTP client) UDP port. All such frames received on this port are flushed.

NOTE: This error type should never occur. If the value of this counter is greater than zero, verify that IP addressing has been properly configured on the NetPerformer.

Invalid op code field

Console	SNMP
Invalid op code field	statBootpInvalidOpCodeField

The number of frames received on the BOOTPC (BOOTP client) or BOOTPS (BOOTP server) port with an invalid operation code field. These frames are neither BOOTREQUEST nor BOOTREPLY frames. All frames with an invalid op code are flushed.

Cannot route frame

Console	SNMP
Cannot route frame	statBootpCannotRouteFrame

The number of frames that could not be routed because none of the Destination IP Addresses configured for BOOTP are valid. The NetPerformer cannot determine over which port it should transmit these frames. All frames that cannot be routed are flushed.

Frame too small to be a BOOTP frame

Console	SNMP
Frame too small to be a BOOTP frame	staBootpFrameTooSmallToBeA-BootpFrame

The number of frames that are smaller than the minimum length of a BOOTP frame. Some fields are missing from these frames. All such frames are flushed.

Reply received on port bootpc

Console	SNMP
Reply received on port bootpc	statBootpReplyReceivedOnPort-Bootpc

The number of BOOTREPLY frames received on the BOOTPC (BOOTP client) UDP port. All such frames received on this port are flushed.

NOTE: Although this error type is unusual, it can occur when a BOOTP/DHCP server is on the local LAN and a client on this LAN sends a broadcasted BOOTREPLY.

Cannot receive and forward on the same port

Console	SNMP
Cannot receive and forward on the same port	statBootpCannotReceiveAnd-ForwardOnTheSamePort

The number of frames that were received over the same port that tries to send them. All such frames are flushed. This error type can occur when the relay agents are in Broadcast Mode (all Destination IP Addresses set to **000.000.000.000**). It should not occur if the NetPerformer is properly configured.

11.5 TIMEP Statistics

11.5.1 DC/TIMEP

Number of frames received

Console	SNMP
Number of frames received	statTimeNbFrameReceived

The total number of frames received by the NetPerformer Time Protocol function.

Number of frames sent

Console	SNMP
Number of frames sent	statTimeNbFrameSent

The total number of frames sent by the NetPerformer Time Protocol function.

Server's number of requests received

Console	SNMP
Server's number of requests received	statTimeNbRequestReceived

The total number of requests received by the time server. A request is in the form of a frame in UDP and a connection in TCP.

Server's number of replies sent

Console	SNMP
Server's number of replies sent	statTimeNbReplySent

The total number of replies sent by the time server.

Client's number of requests sent

Console	SNMP
Client's number of requests sent	statTimeNbRequestSent

The total number of requests sent by the time client. A request is in the form of a frame in UDP and a connection in TCP.

Client's number of replies received

Console	SNMP
Client's number of replies received	statTimeNbReplyReceived

The total number of replies received by the time client.

11.5.2 DE/TIMEP

Client's number of retransmissions

Console	SNMP
Client's number of retrans-missions	statTimeClientRetransmissions

The number of retransmissions that the UDP client has made.

Client's number of sync failures

Console	SNMP
Client's number of sync failures	statTimeClientSyncFailures

The number of times that the client has failed to synchronize the real-time clock. For UDP, this counter is incremented when the allowable number of retransmissions has been executed without success. For TCP, this counter is incremented when the connection between client and server cannot be opened.

Invalid local IP address

Console	SNMP
Invalid local IP address	statTimeInvalidLocalIpAddress

The number of frames that could not be sent because both the LAN port IP address and the global IP address are equal to zero (**000.000.000.000**). When this occurs, there is no valid source IP address to put in the frame.

Frames received with invalid port number

Console	SNMP
Frames received with invalid port number	statTimeInvalidPortNumbers

The number of frames received by the Time Protocol function that had an unknown UDP/TCP port number.

12

Transparent Protocol over IP multicast

12.1 Transparent Protocol over IP multicast

The NetPerformer software has been enhanced to support the transfer of synchronous and asynchronous multi-drop serial protocols (referred to as transparent protocols) over IP Multicast, using UDP/IP encapsulation of the transparent traffic. This enhancement and its configuration are described in this chapter.

12.1.1 Feature Overview

Multi-dropped Lines

Many serial transparent protocols, such as IBM BiSync, Burroughs Poll/Select and asynchronous protocols, are multi-dropped by design. These protocols are used to link one host port to multiple remote terminals using a multi-dropped line, with a concentration ratio ranging from 1:1 to 1:50. They usually run at low speeds (up to 19,200 bps). In the typical transparent protocol application, the host initiates a session with the terminal through a handshaking process, represented by Request/Answer A and B in Figure 12-1. Once a session is established, data transfer takes place in the form of a broadcast transmission from one side to the other. The end of session is accomplished through another Request/Answer procedure.

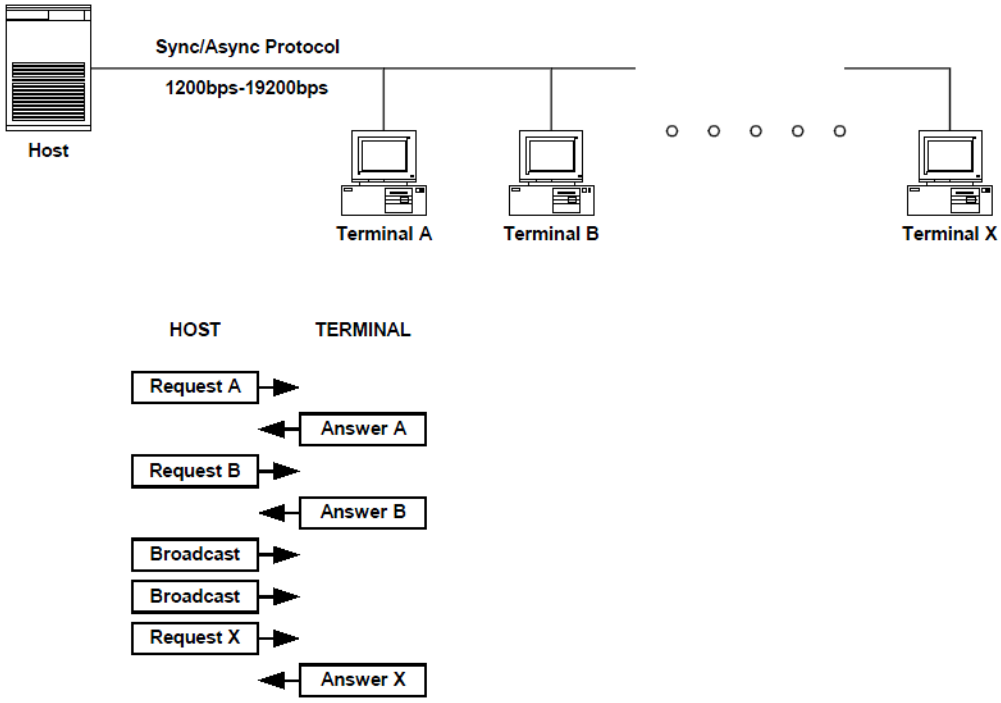


Figure 12-1: A Typical Transparent Protocol Application

12.1.2 Without Transparent Protocol over IP Multicast

The NetPerformer supports a broad range of legacy synchronous and asynchronous protocols in a point-to-point configuration. Without IP encapsulation, a large number of ports may be required on host side NetPerformer units, as shown in Figure 12-2. This increases both the complexity and cost of the network.

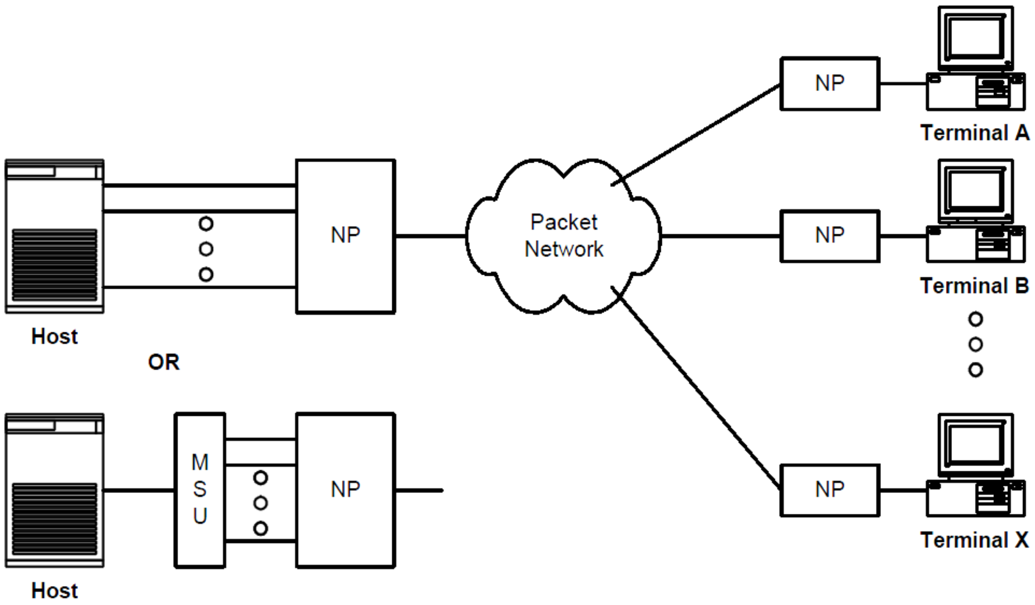


Figure 12-2: Network without IP Encapsulation

Fragmentation and Encapsulation

To simplify this network, the NetPerformer can encapsulate the transparent protocol inside UDP/IP frames before transmission, and send the frames using IP Multicast to each of the remote units. IP Unicast is used on the return route from the remote units to the host side.

- Data coming from the host is cut into fragments (48 bytes or smaller) to avoid excessive latency over the network.
- Each fragment is then encapsulated into an UDP/IP frame for transmission over a packet network.

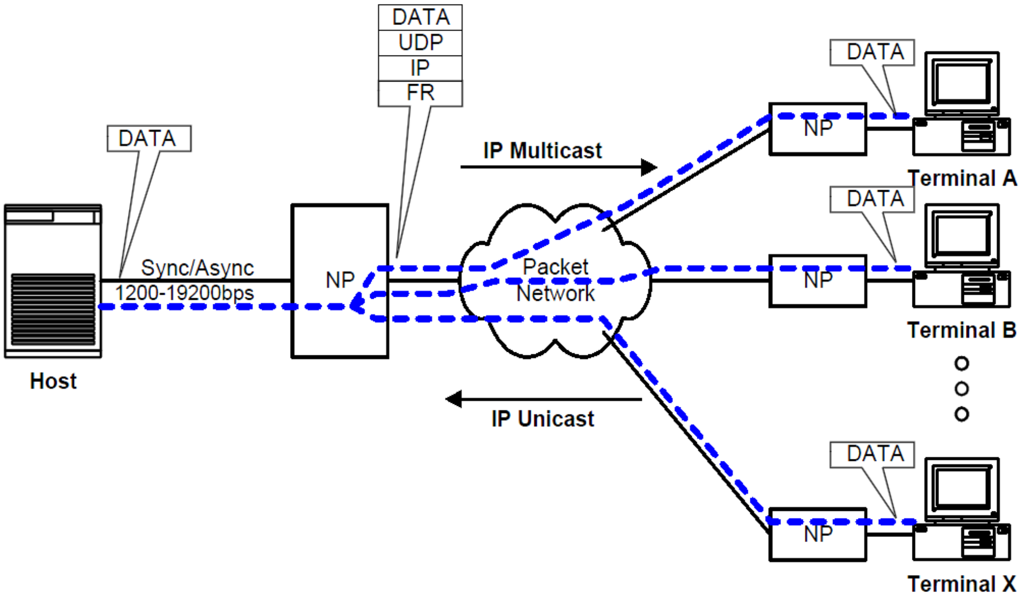


Figure 12-3: UDP/IP Encapsulation Reduces the Port Count on the Host Side

Routing and Extraction

- The IP frames are routed to the remote NetPerformer units using IP Multicast (see Figure 12-3).
- When a frame arrives at a remote NetPerformer unit, its UDP port routes the IPframe to the proper physical port.
- The remote NetPerformer then extracts the data fragment from the IP frame, reassembles the original transparent traffic and sends it to the remote terminal.

Configuration

- Each port configured for transparent operation can support Transparent Protocol over IP Multicast on an individual basis.
- Remote NetPerformer units must be defined with a Multicast Membership address in order to participate in the multicast group.
- At the host location, the destination is defined as the Multicast Membership Address. This allows the NetPerformer to broadcast the data to multiple locations.
- At all participating remote locations, the destination is defined as a Unicast address, equivalent to the IP address of the LAN port on the host side. This reduces the number of ports required on the host-side NetPerformer unit.

Defining the LAN Port

The LAN port on the host side must be configured with a valid IP address.

- On the NetPerformer console, use the SETUP/SLOT menu and enter the number of the slot containing the LAN card.
- Set the IP Address parameter to a valid IP address.
- The IP Multicast Active and IP Multicast Protocol parameters can be left at their default values.

NOTE: This LAN address is used as the value of the Remote Unit parameter for the transparent ports on the remote side. See Remote Unit Parameter on page 52.

Defining the Frame Relay Connection

The UDP/IP encapsulated IP traffic is sent over a WAN connection (a PVCR port or PVC). For transmission over Frame Relay, you need to define the following on both sides of the network:

- A Frame Relay port set to the FR-USER protocol, *and*
- For each remote unit in the multicast group, a separate PVC set to PVCR mode. ***You must activate IP Multicasting on these PVCs:***
  - Set the IP Multicast Active parameter to YES.
  - Set the IP Multicast Protocol parameter to PIMDM.

Defining a Remote Unit as a Multicast Member

The SETUP/IP/GLOBAL menu has a new parameter, Multicast Membership Address, for defining a NetPerformer unit as a member of a multicast group. This parameter is defined on the remote units only. To configure this parameter from the console:

- Enter SETUP at the console command line.
- Enter IP.
- Enter GLOBAL.
- Enter a non-null IP address as the value of the Multicast Membership Address parameter. ***This address must be the same as the value of the Remote Unit parameter you define on the host-side transparent port.*** See ***Remote Unit Parameter***, below.
  - This NetPerformer unit becomes a member of the multicast group, and will receive a copy of all multicast frames received.
  - If other members of the same group can be reached via a WAN or LAN port, a copy of each multicast frame is transmitted over that port.
  - The multicast frames are then processed internally on each member of the multicast group.

NOTE: *If you leave the Multicast Membership Address parameter at its default value,* 0.0.0.0, **the remote NetPerformer will not participate in the multicast group**.

Defining a Transparent Port

To configure a transparent port on the NetPerformer unit:

- Select a data port for configuration. On the NetPerformer console, use the SETUP/PORT menu, choose DATA (if required on your unit), and enter the number of the data port.
- Set the Protocol parameter to T-ASYNC, HDLC, BSC, COP or PASSTHRU.

NOTE: The R-ASYNC protocol does not support the Transparent Protocol over IP Multicast feature.

Remote Unit Parameter

The Remote Unit parameter of a transparent port determines how the NetPerformer will route traffic coming from this port to the unit on the other side of the network.

- **Host-side NetPerformer port:** For a Transparent Protocol over IP Multicast application, the Remote Unit parameter on the host-side transparent port determines whether IP Multicast will be used to send transparent traffic to multiple remote locations.
  - For IP multicasting, configure the Remote Unit parameter to a valid IP address. ***This address must be the same as the value of the Multicast Membership Address parameter you define on the remote NetPerformer units.***
  - The NetPerformer encapsulates the transparent traffic from this port into a UDP/IP frame for immediate transmission.
- **Remote NetPerformer port:** The Remote Unit parameter on a remote transparent port ***must be defined with the IP address of the LAN port on the host-side NetPerformer***.

Remote Port Parameter

The Remote Port parameter of a transparent port determines where the traffic will be sent on the remote unit. If you have defined the Remote Unit parameter for this port as an IP address, the NetPerformer automatically interprets the value of the Remote Port parameter as a logical UDP port.

- **On both sides of the network:** Select any port number from 1 to 255. ***All participating transparent ports must have the same Remote Port value.***

NOTE: Data will be transported correctly from one end to another ***only if the remote ports are set to the same logical UDP port***.

12.1.3 Application Example

The following example shows how to configure a simple Transparent Protocol over IP Multicast application, represented in Figure 12-4. In this example:

- The NetPerformer on the host side has the unit name CENTRAL.
- Two units are defined on the remote side: REMOTE-1 and REMOTE-2. As these units have an identical configuration, only the procedure for REMOTE-1 is shown below.
- The logical UDP port is defined as port 5 on both sides of the network.

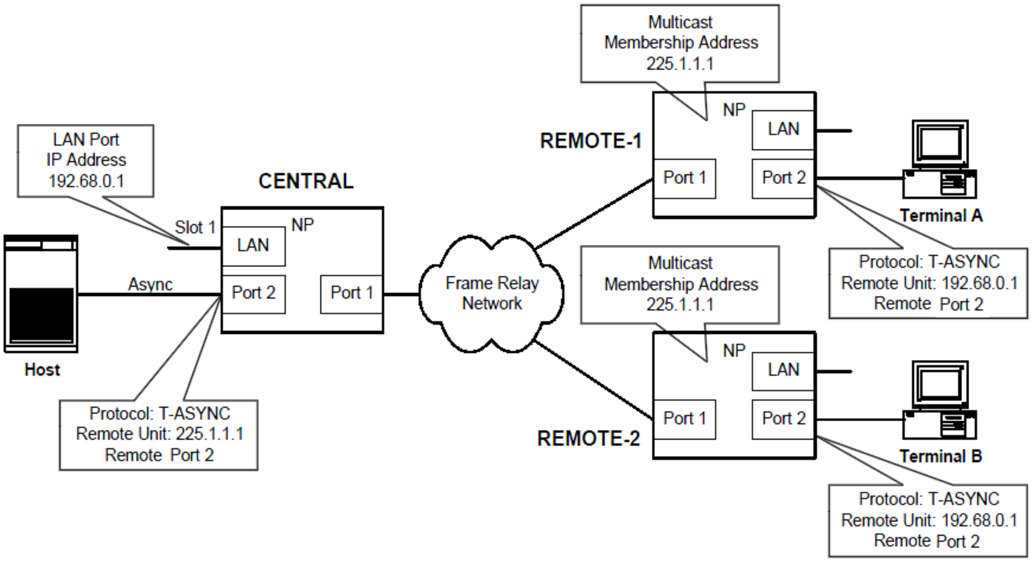


Figure 12-4: A Transparent Protocol over IP Multicast Application

12.1.4 Host Side Configuration (CENTRAL)

1. Define the LAN port:

```
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:GLOBAL) ? SLOT
SLOT> Slot number (1/2,def:1) ? 1
SLOT 1>
PORT ETH> Protocol (def:ETH AUTO) ?
PORT ETH> MAC address (def:000000000000) ?
PORT ETH> IP address (def:000.000.000.000) ? 198.68.0.1
PORT ETH> Subnet mask (number of bits) (0-24,def:0) ?
{255.255.255.000}
PORT ETH> Frame size (128-4096,def:1500) ?
PORT ETH> IP RIP (def:V1) ?
PORT ETH> IP RIP TX/RX (def:DUPLEX) ?
PORT ETH> OSPF (def:DISABLE) ?
PORT ETH> IGMP enable (def:NO) ?
PORT ETH> IP multicast active (def:NO) ?
PORT ETH> IP multicast protocol (def:NONE) ?
PORT ETH> NAT enable (def:NO) ?
PORT ETH> IPX RIP (def:DISABLE) ?
PORT ETH> IPX SAP (def:DISABLE) ?
PORT ETH> IPX network number (def:00000000) ?
```

2. Define the Frame Relay port:

```
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:SLOT) ? PORT
Port type (DATA/VOICE,def:DATA) ? DATA
Port number (CSL/1/2/3/4,def:1) ? 1
PORT #1> Protocol (def:FR-USER) ? FR-USER
PORT #1> Port speed (bps) (1200-2048000,def:56000) ?
PORT #1> Fallback speed (def:ENABLE) ?
PORT #1> Interface (def:DTE-V35) ?
PORT #1> Clocking mode (def:EXTERNAL) ?
PORT #1> Management interface (def:LMI) ?
PORT #1> Congestion flow control (def:ON) ?
PORT #1> Enquiry timer (sec) (2-30,def:10) ?
PORT #1> Report cycle (1-256,def:6) ?
PORT #1> CLLM function (def:OFF) ?
PORT #1> Cell Packetization (def:YES) ?
PORT #1> Maximum number of voice channels (0-10000,def:10000) ?
PORT #1> Maximum Voice Channels If High Priority Data (0-
10000,def:10000) ?
PORT #1> Drop signals on LMI down (def:NO) ?
PORT #1> SVC address type (def:NONE) ?
```

3. Define a Frame Relay PVC to each remote unit. *You must activate IP multicasting on these PVCs*:

```
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:PORT) ? PVC
PVC number (1-96,def:1) ? 1
PVC #1> Mode (def:PVCR) ? PVCR
PVC #1> Port (def:1) ? 1
PVC #1> DLCI address (0-1022,def:0) ? 100
PVC #1> Committed Information rate (1200-2048000,def:56000) ?
PVC #1> Burst Information rate (1200-2048000,def:56000) ?
PVC #1> Remote unit name (def:) ? REMOTE-1
PVC #1> Type (def:DEDICATED) ?
PVC #1> Timeout (msec) (1000-30000,def:1000) ?
PVC #1> Number of retransmission retries (1-1000,def:100) ?
PVC #1> Compression (def:YES) ?
PVC #1> IP address (def:000.000.000.000) ?
PVC #1> Subnet mask (number of bits) (0-24,def:0) ? {000.000.000.000}
PVC #1> IP RIP (def:V1) ?
PVC #1> IP RIP TX/RX (def:DUPLEX) ?
PVC #1> OSPF (def:DISABLE) ?
PVC #1> IP multicast active (def:NO) ? YES
PVC #1> IP multicast protocol (def:NONE) ? PIMDM
PVC #1> IPX RIP (def:DISABLE) ?
PVC #1> IPX SAP (def:DISABLE) ?
PVC #1> IPX NETWORK NUMBER (def:00000000) ?
PVC #1> NAT enable (def:NO) ?
PVC #1> Broadcast group (def:NO) ?
PVC #1> Maximum number of voice channels (0-10000,def:10000) ?
PVC #1> Maximum Voice Channels If High Priority Data (0-
10000,def:10000) ?
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:PORT) ? PVC
PVC number (1-96,def:1) ? 2
PVC #1> Mode (def:PVCR) ? PVCR
PVC #1> Port (def:1) ? 1
PVC #1> DLCI address (0-1022,def:0) ? 101
PVC #1> Committed Information rate (1200-2048000,def:56000) ?
PVC #1> Burst Information rate (1200-2048000,def:56000) ?
PVC #1> Remote unit name (def:) ? REMOTE-2
PVC #1> Type (def:DEDICATED) ?
PVC #1> Timeout (msec) (1000-30000,def:1000) ?
PVC #1> Number of retransmission retries (1-1000,def:100) ?
PVC #1> Compression (def:YES) ?
PVC #1> IP address (def:000.000.000.000) ?
PVC #1> Subnet mask (number of bits) (0-24,def:0) ? {000.000.000.000}
PVC #1> IP RIP (def:V1) ?
PVC #1> IP RIP TX/RX (def:DUPLEX) ?
PVC #1> OSPF (def:DISABLE) ?
PVC #1> IP multicast active (def:NO) ? YES
PVC #1> IP multicast protocol (def:NONE) ? PIMDM
PVC #1> IPX RIP (def:DISABLE) ?
PVC #1> IPX SAP (def:DISABLE) ?
PVC #1> IPX NETWORK NUMBER (def:00000000) ?
PVC #1> NAT enable (def:NO) ?
PVC #1> Broadcast group (def:NO) ?
PVC #1> Maximum number of voice channels (0-10000,def:10000) ?
PVC #1> Maximum Voice Channels If High Priority Data (0-
10000,def:10000) ?
```

4. Define the transparent port. This application uses the T-ASYNC protocol. Make sure you set the Remote Unit parameter to the Multicast Membership Address of the remote units.

```
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:PVC) ? PORT
Port type (DATA/VOICE,def:DATA) ? DATA
Port number (CSL/1/2/3/4,def:1) ? 2
PORT #2> Protocol (def:HDLC) ? T-ASYNC
PORT #2> Port speed (bps) (50-115200,def:56000) ? 19200
PORT #2> Interface (def:DTE-V35) ? DCE-RS232
PORT #2> Clocking mode (def:ISO-INT) ? ASYNC
PORT #2> Format (def:8-NONE) ?
PORT #2> Modem control signal (def:STATIC) ?
PORT #2> Remote unit (def:) ? 225.1.1.1
PORT #2> Class (def:3) ?
PORT #2> Remote port number (1-255,def:2) ? 2
```

12.1.5 Remote Side Configuration (REMOTE-1)

1. Define the remote unit as a member of the multicast group:

```
REMOTE-1>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:GLOBAL) ? IP
Item (GLOBAL/STATIC/BOOTP/OSPF/TIMEP/SNMP/NAT,def:GLOBAL) ? GLOBAL
IP> Router (def:ENABLE) ?
IP> Route broadcast to end station (def:NO) ?
IP> Multicast membership address (def:000.000.000.000) ? 225.1.1.1
```

2. Define the Frame Relay port:

```
REMOTE-1>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:IP) ? PORT
Port type (DATA/VOICE,def:DATA) ? DATA
Port number (CSL/1/2/3/4,def:1) ? 1
PORT #1> Protocol (def:FR-USER) ? FR-USER
PORT #1> Port speed (bps) (1200-2048000,def:56000) ?
PORT #1> Fallback speed (def:ENABLE) ?
PORT #1> Interface (def:DTE-V35) ?
PORT #1> Clocking mode (def:EXTERNAL) ?
PORT #1> Management interface (def:LMI) ?
PORT #1> Congestion flow control (def:ON) ?
PORT #1> Enquiry timer (sec) (2-30,def:10) ?
PORT #1> Report cycle (1-256,def:6) ?
PORT #1> CLLM function (def:OFF) ?
PORT #1> Cell Packetization (def:YES) ?
PORT #1> Maximum number of voice channels (0-10000,def:10000) ?
PORT #1> Maximum Voice Channels If High Priority Data (0-
10000,def:10000) ?
PORT #1> Drop signals on LMI down (def:NO) ?
PORT #1> SVC address type (def:NONE) ?
```

3. Define a Frame Relay PVC, and activate IP multicasting on it:

```
CENTRAL>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:PORT) ? PVC
PVC number (1-96,def:1) ? 1
PVC #1> Mode (def:PVCR) ? PVCR
PVC #1> Port (def:1) ? 1
PVC #1> DLCI address (0-1022,def:0) ? 100
PVC #1> Committed Information rate (1200-2048000,def:56000) ?
PVC #1> Burst Information rate (1200-2048000,def:56000) ?
PVC #1> Remote unit name (def:) ? CENTRAL
PVC #1> Type (def:DEDICATED) ?
PVC #1> Timeout (msec) (1000-30000,def:1000) ?
PVC #1> Number of retransmission retries (1-1000,def:100) ?
PVC #1> Compression (def:YES) ?
PVC #1> IP address (def:000.000.000.000) ?
PVC #1> Subnet mask (number of bits) (0-24,def:0) ? {000.000.000.000}
PVC #1> IP RIP (def:V1) ?
PVC #1> IP RIP TX/RX (def:DUPLEX) ?
PVC #1> OSPF (def:DISABLE) ?
PVC #1> IP multicast active (def:NO) ? YES
PVC #1> IP multicast protocol (def:NONE) ? PIMDM
PVC #1> IPX RIP (def:DISABLE) ?
PVC #1> IPX SAP (def:DISABLE) ?
PVC #1> IPX NETWORK NUMBER (def:00000000) ?
PVC #1> NAT enable (def:NO) ?
PVC #1> Broadcast group (def:NO) ?
PVC #1> Maximum number of voice channels (0-10000,def:10000) ?
PVC #1> Maximum Voice Channels If High Priority Data (0-
10000,def:10000) ?
```

4. Define the transparent port. Make sure you set the Remote Unit parameter to the LAN port address on the host-side unit.

```
REMOTE-1>SE
SETUP
Item (GLOBAL/PORT/SLOT/PU/SCHEDULE/IP/BRIDGE/PHONE/FILTER/CLASS/PVC/
IPX/MAP/HUNT,def:PVC) ? PORT
Port type (DATA/VOICE,def:DATA) ? DATA
Port number (CSL/1/2/3/4,def:1) ? 2
PORT #2> Protocol (def:HDLC) ? T-ASYNC
PORT #2> Port speed (bps) (50-115200,def:56000) ? 19200
PORT #2> Interface (def:DTE-V35) ? DCE-RS232
PORT #2> Clocking mode (def:ISO-INT) ? ASYNC
PORT #2> Format (def:8-NONE) ?
PORT #2> Modem control signal (def:STATIC) ?
PORT #2> Remote unit (def:) ? 198.68.0.1
PORT #2> Class (def:3) ?
PORT #2> Remote port number (1-255,def:2) ? 2
```

The unit REMOTE-2 shown in Figure 3-7 is configured in the same way as REMOTE-1.

Index

REACH FURTHER. OFFER MORE.

Contact Memotec:

tel.: +1-514-738-4781
e-mail: MemotecSupport@memotec.com

7755 Henri Bourassa Blvd. West
Montreal, Quebec | Canada H4S 1P7

www.memotec.com

MEMOTEC
redefining network efficiency