NetPerformer® System Reference

Quality of Service

COMTECH EF DATA

MEMOTEC
redefining network efficiency

COPYRIGHTS AND DISCLAIMERS

Published Date: January 27, 2020
Document # 1606

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).

NetPerformer, SDM-8400, and the SDM-9XXX series of products are either registered trademarks or trademarks of Memotec Inc.in Canada, the United States of America, and in other countries.

Any other trademarks are the property of their respective companies.

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

Contents

List of Tables

List of Figures

1

NetPerformer QoS Architecture

1.1 QoS Support

The NetPerformer supports 8 classes of service for prioritizing legacy types of serial traffic, such as SNA, as well as LAN bridging and IP/IPX routing. A priority weighting is assigned to each class of service. The NetPerformer QoS architecture allows traffic to be divided into a hierarchy of classes, each one with its own relative weight.

- In case of congestion, each class is limited to a certain percentage of the available bandwidth
- Borrowing bandwidth from under-utilized or idle classes is permitted.
- All large data packets or frames are segmented into cells of 96 bytes maximum and the relative weight is measured on this basis.
- The cell-based architecture of the NetPerformer (PowerCell) allows very fine granularity and prevents, for example, a large IP data frame of 1500 bytes being treated the same way as a 20 byte voice sample.

NOTE: *PowerCell* is the trademarked term for the PVCR protocol (Programmable Variable Cell Relay).

Many different traffic types may be active on the LAN (FTP, Telnet, UDP, etc.). The NetPerformer provides the ability to further prioritize this traffic according to protocol type, port number or address.

While supporting standard QoS at both Layer 2 and 3, the NetPerformer also does mapping and conversion to provide end-to-end QoS, filling the gaps. With the advent of VoIP and other real-time applications this is a particularly valuable feature.

In addition to its use as an access device, the NetPerformer can provide switching at intermediate nodes.

- A network designed with NetPerformer at all locations that uses PowerCell over all WAN links can maintain QoS even when going over multiple hops and through multiple nodes.
- QoS is achieved independently of the network infrastructure used: switched or leased lines including ISDN, Frame Relay, satellite, or wireless.
- When using PowerCell over IP, prioritization within the IP tunnel is used to deliver the highest priority traffic to the network first (see next section).

1.2 PowerCell Prioritization over IP

PowerCell provides data compression, which significantly reduces the amount of bandwidth required across the WAN. This is especially true for IP traffic, including VoIP. Latency and jitter are also reduced through more efficient use of bandwidth.

PowerCell can also be encapsulated over IP to take advantage of cost effective IP-based carrier offerings.

- PowerCell not only provides the ability to prioritize all the traffic effectively, but can also regulate the flow of information over IP.
- The rate of information transfer can be configured to match any potential bottleneck in the network, regulating the flow of information from all PowerCell users in the network.
- The NetPerformer can also set the IP Precedence bits in order to take advantage of prioritization on the Service Provider network.
- The user can define the relative weight of all classes, and benefit from the PowerCell QoS architecture by mapping IP Precedence to the internal NetPerformer classes of service.

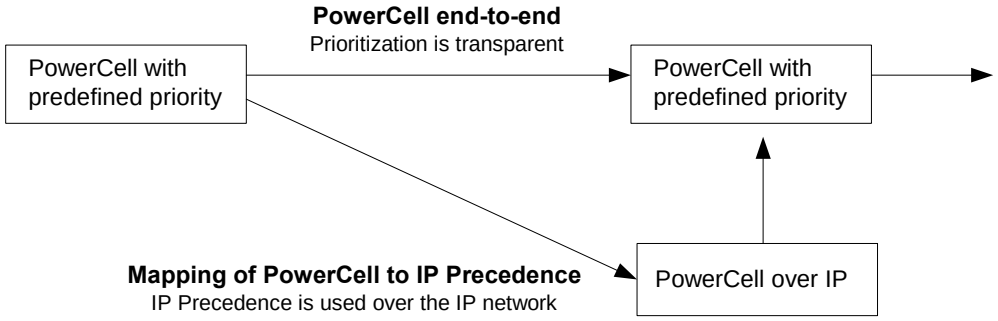

Figure 1-1: PowerCell Prioritization over IP

1.3 QoS Standards on the NetPerformer

Since IP is rapidly becoming the backbone protocol of choice, support of standards-based QoS is crucial to providing consistent levels of service across the network.

The NetPerformer supports the following QoS standards:

- **IEEE 802.1p/q:** Initially designed to support Virtual LAN (VLAN) interoperability, this standard was extended to support traffic priority.
  - The VLAN function was used to logically separate groups of users who share the same physical LAN segment. Refer to the Virtual LAN fascicle of this document series.
  - For VLAN traffic prioritization, the NetPerformer includes a special filter that adapts the 802.1p prioritization bits to PowerCell classes of service, allowing a relative priority weight to be used over the network.
  - When the remote destination does not use VLAN, the NetPerformer can be configured to map the Layer 2 prioritization bits to the Layer 3 IP Precedence (using the *VLAN Priority Conversion* parameter on the LAN port). This ensures that VLAN traffic is properly prioritized end-to-end over a fully routed network.
- **IETF Differentiated Services, or DiffServ:** A QoS mechanism related to IP Precedence, DiffServ offers up to 64 service classes and overcomes some Type of Service (ToS) limitations, especially when multiple networks are involved. See “DiffServ 802.1 p/q ToS” on page 3-1.

1.4 Voice/Fax Prioritization

Due to the delay-sensitive nature of voice and fax transmissions, it is essential that they be prioritized to avoid the generation of delays. All traffic entering a NetPerformer channel is automatically given high-priority status with respect to data from other sources. This ensures guaranteed bandwidth access in a shared network environment and an uninterrupted flow from source to destination.

Through *protocol sorting*, the NetPerformer routing unit differentiates delay-sensitive cells from non-sensitive cells, and directs these cells to their corresponding transmit queues. The architecture of the NetPerformer receivers, combined with the differentiation of data upon entrance to the receivers, ensures accelerated processing of voice and fax without jeopardizing the processing of non-sensitive data.

Each NetPerformer along the virtual path selects a virtual channel according to the priority class assigned to the virtual connection. In this way, the high priority of voice/fax traffic is maintained uniformly along the entire virtual path.

Each NetPerformer routing unit monitors its transmit queue for the priority level of the cells that are received. When high-priority cells arrive, they are expedited to the next NetPerformer on the virtual path before processing the lower priority cells. Lower-priority cells are buffered until the higher-priority voice and fax cells are sent. This guarantees voice/fax prioritization throughout the network.

2

Traffic Priority

2.1 About Traffic Priority

An important aspect of the NetPerformer is its ability to prioritize mission-critical data.

- Delay sensitive protocols such as LLC2, SDLC or any other legacy protocol can be assigned a high priority so that they reach their destination before other less critical data cells, eliminating response time problems and session time-outs.
- In a shared network environment, mission critical applications can have guaranteed access to their part of the bandwidth.

To ensure prioritization of mission-critical data, the partitioning of the bandwidth must be planned carefully.

2.1.1 Prioritization

You can set the priority class separately for each traffic type or user destination (transparent port, PU, PVC or LAN traffic). Refer to “Configuring the Traffic Classes” on page 2-8.

- Each priority class is assigned a weight value, which determines how bandwidth should be allocated in case of congestion.
- The higher the weight, the greater the bandwidth portion used. The traffic in a high weight class will receive higher priority.
- Once the priority class is set for a particular traffic type, it is maintained uniformly from one unit to the next along the entire virtual path. This guarantees protocol prioritization throughout the network.

Traffic can also be prioritized with respect to other traffic types by taking advantage of protocol sorting and filters.

- You decide what priority class a particular traffic type should belong to, and set its priority by writing a filter that includes the name of the traffic type and its priority class.

  For example, **TELNET>3** places any TELNET traffic in priority class 3.

Filters are discussed further in the next section. Examples of all filter types are provided in the appendix “Traffic Filters” on page 4-1.

2.2 Traffic Filtering

The NetPerformer includes a filtering capability which can be used to prioritize a particular traffic type over the link or reduce broadcast traffic across the network. Each filter contains a set of criteria that:

- Send specific frames over the composite link at a certain priority level
- Prevent extraneous frames from being forwarded.

This enables the NetPerformer to keep up with the traffic flow and permits prioritization of specific traffic types.

An excessively large number of broadcast frames crossing the LAN segments can reduce the performance of remote bridge applications, where bandwidth is at a premium. For example, the use of all-routes, all-stations broadcasts for route determination under TCP/IP can place an unnecessary load on the composite links. If your network is large with multiple LANs, you should understand how broadcast techniques are used, and design filters (and your LAN topology) accordingly.

2.2.1 Configuring the Traffic Filters

The **FILTER** submenu of the **SETUP** command includes all parameters that are required to define a traffic filter.

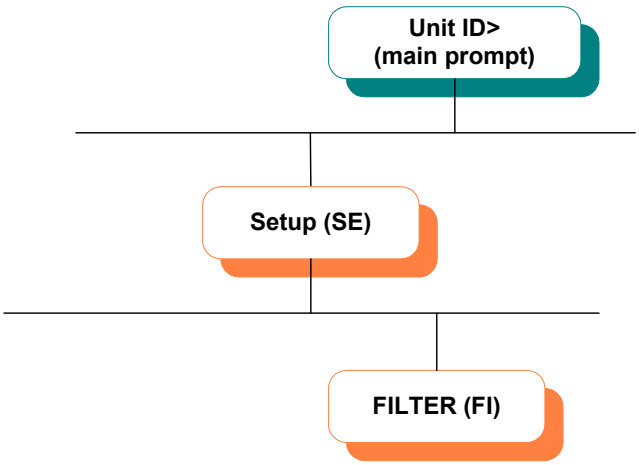

Figure 2-1: SETUP/FILTER Path in the CLI Tree

► To configure a traffic filter:

1. At the NetPerformer command line prompt, enter the menu sequence: **SE** ↵ **FILTER**
2. Select the *FILTER number*
3. Enter the filter *Definition*. Refer to "Filter Syntax" on page 2-4, and the appendix "Traffic Filters" on page 4-1
4. Set *Active* to **YES**.

SE/FILTER 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) ? FILTER
FILTER number (1-32,def:1) ?
FILTER 1> Definition (def:) ? TELNET>3
FILTER 1> Active (def:NO) ? YES
```

These filter parameters and their SNMP and configuration text equivalents are detailed in "SE/FILTER Configuration Parameters" on page 4-2.

2.2.2 Filter Syntax

Filters can be written in two ways, depending on the traffic type:

- The filter assigns a particular traffic type to a particular class (from **1** to **8**).

  Example: FTP>4, which assigns all FTP traffic to class 4. The actual priority of FTP traffic is determined by the way class 4 is defined, as explained in "Configuring the Traffic Classes" on page 2-8.

  - This simple method of writing a filter can be used for the following traffic types: APPLETALK, FRoIP, FTP, IPX, IPXSAP, NETBIOS, SNA, SNMP, TELNET
  - The traffic type name *must* precede the **>** symbol. Enter the name exactly as you see it listed in the appendix "Traffic Filters" on page 4-1.

- The filter assigns a specific address, port or frame value to a particular class.

  Example: IPSRC(198.168.43.0,255.255.255.0)>3, which instructs the NetPerformer to mask the source IP address of all IP frames received, and compare them with address 198.168.43.0. If a match is found the frame is sent under the priority determined by class 3.

  - Examples of this method of writing a filter include those for the following traffic types: IP, MAC, TCP, UDP
  - IP and MAC filters specify the source or destination address as well as the mask. Each address received is masked and compared with the specified address. The filter syntax is:

IPSRC(address,mask)>class or *IPDST(address,mask)>class*

Example: IPDST(128.128.128.0,255.255.255.0)>3

MACSRC(address)>class or *MACDST(address)>class*

Example: MACSRC(002083XXXXXX)>3, which generates the source MAC address 002083000000 and the MAC mask FFFFFF000000.

- TCP and UDP filters specify the minimum and maximum destination ports. All frames whose destinations lie within this range (including the minimum and maximum ports themselves) are sent under the priority determined by the specified class. The filter syntax is:

TCP(min_port,max_port)>class

Example: TCP(2065,2067)>3

UDP(min_port,max_port)>class

Example: UDP(160,163)>3

- Filters can also be combined with the logical operators **!** (not), **&** (and) and **|** (or). For example, to assign a subset of IP and MAC data to class 3, you could write the filter:

  IPSRC(128.128.128.0,255.255.255.0)|MACDST(123456789ABC)>3

NOTE: Examples of all filter types are given in the appendix "Traffic Filters" on page 4-1. See also "Configuring the Traffic Filters" on page 2-3.

2.2.3 Discarding Frames

You can also use the filters to discard frames. To do this:

- Enter **X** instead of the class number after the > symbol in the filter definition.

  Example: FTP>X, which instructs the NetPerformer to discard all FTP traffic.

  Use this notation whenever possible to reduce broadcast traffic across the network.

NOTE: FTP and TELNET traffic can also be restricted to a list of authorized IP addresses. For details, refer to the chapter *Controlling Access to the NetPerformer* in the *Quick Configuration*.

2.2.4 How the Filters are Applied

Default Order of Application

The NetPerformer supports a maximum of 32 filters, numbered from **1** to **32**. When more than one filter is defined:

- The NetPerformer applies all configured filters to each frame, starting with the lowest numbered filter.

  This is the order the filters are applied on power up or reset. If you add a new filter during a session, it will be applied after the other filters, even if it is a lower numbered filter.

- When a match is found, the NetPerformer performs the action specified by the class number (or **X**), and then analyzes the next frame.

  Since the filters are applied in order, you should define the filters that affect a large number of frames before the filters that affect fewer frames, to minimize the number of filters the NetPerformer has to examine.

Example:

- If your network has mostly MAC traffic, use Filter 1 to define the MAC filter. This way, a higher amount of frames will be handled by the first filter.
- If you also have SNMP and UDP traffic, use Filter 2 for SNMP and Filter 3 for UDP. This will avoid sending the SNMP traffic under the priority class intended for UDP, since SNMP runs under UDP.
- If you want to discard specific IP frames, use a higher numbered filter. The NetPerformer will have to examine this filter only if the frame does not match any of the lower numbered filters.

Fine-tuning the Application

By default, all filters are applied to all connections on the NetPerformer unit in the order described in the preceding section. WAN traffic filters, however, can be fine tuned on an individual connection (port or PVC) using the *Filter* parameter. Each port or PVC can be configured with a different filter application scenario.

NOTE: This includes traffic going over the FireWire links of an SDM-9500 chassis.

- Enter **ALL** to apply all traffic filters that were defined with the **SETUP/FILTER** command. This is the default value for all WAN connections and FireWire links.

  Use the **DP/FILTER** console command to view the current list of filters.
- Enter **NONE** to disable filtering on an individual WAN connection.
- Enter a specific filter number (from **1** to **32**) or a set of filter numbers to select a subset of the filters that have been defined for this NetPerformer unit.

  When selecting more than one filter, separate each filter number with commas:

  ```
  PORT 1> Filter (def:ALL) ? 2,21,10
  ```

  The filters you select are applied to the traffic that passes over this WAN connection **in the order you specify** for this parameter. In the above example, Filter **21** would be applied before Filter **10**.

NOTE: WAN configuration details are provided in the *WAN/Leased Lines* and *WAN/Frame Relay* fascicles of this document series.

2.3 Traffic Distribution and Bandwidth Allocation

Traffic enters the NetPerformer from several sources and, depending on how the ports are used, the output traveling over the link can be sorted into several classes.

The link can accommodate 8 possible classes, each with its own user-defined relative weight. This determines how much of the available bandwidth each class can use.

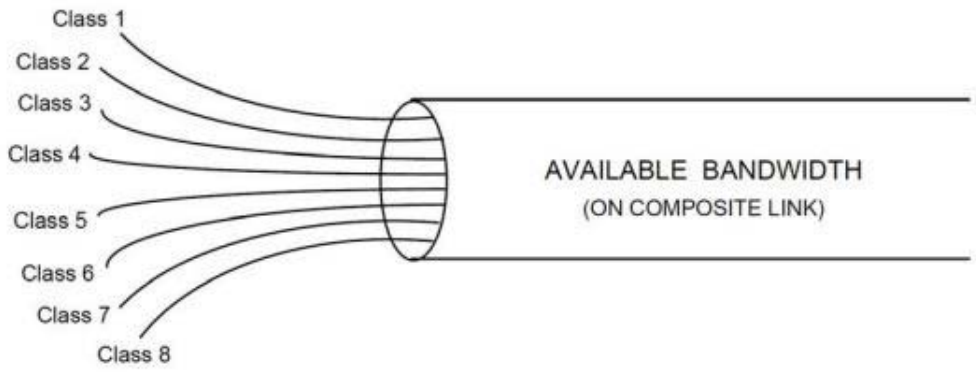

Figure 2-2: Classes Sharing the Bandwidth

Due to the multiple demands made on the bandwidth, the portion of the bandwidth allocated to each class must be carefully determined in terms of its relative weight. The weight determines how many cells or packets are allowed to be transmitted for a particular class versus the other classes within a given period of time.

Each user port, PU, PVC and LAN connection is assigned to a particular class by default. You can change both the class assignment and its weight for any individual connection. Refer to “Configuring the Traffic Classes” on page 2-8.

Type of Connection	Default Class	Default Weight
LAN (IP routing, bridge)	1	1
PU (LINKS mode)	2	1
PVC (Frame Relay)	3	1
User port (HDLC, PASSTHRU, T-ASYNC, R-ASYNC, BSC, COP, DDCMP protocol)	3	1

Table 2-1: Default class and weight of NetPerformer connections

Example:

- You could configure class 1 with weight 2 for 50% of the bandwidth, and leave classes 2 and 3 at the default weight 1 for 25% of the bandwidth each.
- In this way, all user destinations that are set to priority class 1 are guaranteed 50% of the bandwidth, regardless of the frame size used or the number of frames generated.

- If you did not change the default class assignments, this would mean that all LAN traffic would receive higher priority than the other traffic types.

NOTE: If you leave the weight at its default value of 1 for all classes, the NetPerformer will not differentiate between traffic types. That is, all traffic types defined on your network will receive an equal portion of the bandwidth. **To prioritize data you must increase the weight of the class to which the data type belongs.**

The class assignments can be supplemented by filters, which make specific correlations between data types and priority classes.

- This can be used to effectively handle data that may not originate from the direct NetPerformer connection.

  For example, bridged SNA traffic will be assigned to class 1 by default unless you define the filter **SNA>2**.
- Filters can also be used to handle exceptions to how the classes are assigned for a particular data type, as is often required for IP routing.

For details on writing filters, see “Configuring the Traffic Filters” on page 2-3 and “Filter Syntax” on page 2-4.

2.3.1 Configuring the Traffic Classes

The **CLASS** submenu of the **SETUP** command includes all parameters that are required to define all traffic classes, and to select which traffic class shall be used as the default class for traffic types that have not been configured with a specific class.

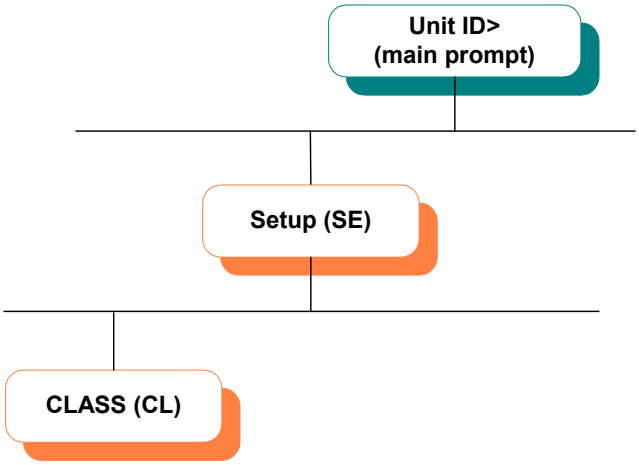

Figure 2-3: SETUP/CLASS Path in the CLI Tree

► To configure a traffic class:

1. At the NetPerformer command line prompt, enter the menu sequence: **SE** ↵ **CLASS**
2. Select an integer value for the *CLASS number*
3. Enter the relative bandwidth *Weight* that can be used by this class on the WAN link (SNMP equivalent: *classWeight*). Refer to "Bandwidth Allocation Example" on page 2-10

   The higher the weight, the greater the amount of bandwidth that will be allocated to traffic in this class. Thus the class with the highest relative weight will have the highest priority, which can be used to guarantee the response time for mission-critical traffic.
4. Define the *Preferred route* with the *Unit name* of a remote NetPerformer unit (SNMP equivalent: *classPrefRoute*).

   When multiple WAN links are required to reach the traffic destination, this unit will be used as the next hop for this class of traffic if:

   - More than one route is available, *and*
   - All available routes are equal in cost.

   Tip: Assigning a different *Preferred route* to different classes helps ensure static load balancing of transparent traffic over all routes.

SE/CLASS example: with integer CLASS number

```
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) ? CLASS
CLASS number (DEFAULT/1/2/3/4/5/6/7/8,def:1) ?
CLASS 1> Weight (1-16,def:1) ? 2
CLASS 1> Preferred route (def:) ? CHICAG0-9230
```

The *default traffic class* represents the class number used for all LAN traffic that is not redirected to a specific traffic class using filters.

► To select the default traffic class:

1. At the NetPerformer command line prompt, enter the menu sequence: **SE** ↵ **CLASS**
2. Enter **DEFAULT** as the *CLASS number*
3. Select the *Default class*.

SE/CLASS example: with default CLASS

```
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) ? CLASS
CLASS number (DEFAULT/1/2/3/4/5/6/7/8,def:1) ? DEFAULT
CLASS> Default class (0-8,def:1) ?
```

2.3.2 Bandwidth Allocation Example

The following scenario is for a network that uses IP routing and handles both SNA and BSC data. The same procedure can be used for any network that interconnects devices with different traffic types.

► To allocate the available bandwidth:

1. Draw a table with your *Traffic Type* list under one column, and add columns for the *Class*, *Weight* and *Bandwidth*. Enter the default weight (always 1) and class for each traffic type, referring to Table 2-1.

Traffic Type	Class	Weight	Bandwidth
SNA	2	1	__ %
IP	1	1	__ %
BSC	3	1	__ %

Table 2-2: Example of Bandwidth Allocation Characteristics (Part 1)

2. Decide which traffic type requires the highest priority. Raise its weight to 2.

   If more than two levels of priority are required, raise the weight of the highest priority traffic to 3 and the weight of the medium priority traffic to 2.

Traffic Type	Class	Weight	Bandwidth
SNA	2	2	__ %
IP	1	3	__ %
BSC	3	1	__ %

Table 2-3: Example of Bandwidth Allocation Characteristics (Part 2)

3. Add up all values in the *Weight* column. Fill in the *Bandwidth* column for each traffic type by calculating the percentage of the weight divided by the total weight:

$$\left(Bandwidth = \frac{Weight}{TotalWeight} \times 100\right)$$

Traffic Type	Class	Weight	Bandwidth
SNA	2	2	33%
IP Routing	1	3	50%
BSC	3	1	17%

Table 2-4: Example of Bandwidth Allocation Characteristics (Part 3)

4. Verify that the proposed bandwidth usage for each traffic type matches the needs of your network. If it does not, adjust the weight values accordingly.
5. Configure the classes using the **SETUP/CLASS** menu, as described in "Configuring the Traffic Classes" on page 2-8. For each class you require, specify its weight.

6. If you want to configure an individual port, PU or PVC as belonging to a different class than the default class, adjust the *Class* parameter for that port, PU or PVC.
7. Add any filters you may need, using the **SETUP/FILTER** menu, described in “Configuring the Traffic Filters” on page 2-3.

In this example, the filter **SNA>2** will ensure that bridged SNA data is given the same priority treatment as data from the PUs in spoofing mode. With this filter the bridged SNA data will be sent under class 2, rather than the default LAN traffic class 1.

3

DiffServ 802.1 p/q ToS

3.1 NetPerformer DiffServ Support

Another QoS capability of the NetPerformer is its support of IETF Differentiated Services, or DiffServ. This feature allows specific protocols with low-latency or high-bandwidth demands to be routed more efficiently than other protocols. In particular, DiffServ allows prioritization of IP frames received from user devices such as IP phones.

NOTE: The NetPerformer is also capable of setting the IP Precedence bits when transmitting PowerCell over IP, to preserve the prioritization scheme over IP-based backbones. See “QoS Standards on the NetPerformer” on page 1-4.

DiffServ is an extension of prioritization models that use IP precedence, where the Type of Service (ToS) field in the IP header is used to assign a particular level of privilege to the packet. In the case of DiffServ, six bits of the ToS field are used (rather than only three in the IP precedence model), providing greater flexibility in managing traffic prioritization.

Prior to version 9.2.0, the NetPerformer was able to prioritize internal traffic (that is, internally generated voice packets) as well as external traffic received using a known protocol, for example, Telnet. With DiffServ, the NetPerformer can also support standard prioritization schemes coming from a connected IP-based device. Furthermore, the priority level assigned to each class of traffic is customizable, using the **IPQOS** filter.

- The **IPQOS** filter maps the 6-bit differentiated service (DS) field in the IP header to the NetPerformer’s 8 classes of service plus the **HIGH** priority setting, which is usually used for voice.
- This mapping provides a total of 32 filter definitions, providing a good level of flexibility in various applications.
- The DS field can be set according to the priority given to a specific kind of traffic or protocol. This allows the network to process those packets more efficiently, and accommodate any real-time high bandwidth requirements this traffic may have.

3.2 Setting Up the DiffServ Values

On the NetPerformer, DiffServ is implemented using the already existing **FILTER** submenu of the **SETUP** (**SE**) command. A new filter, **IPQOS**, can be defined to differentiate between packets containing different values in their DS fields. These values are mapped to a configured traffic priority class, which the NetPerformer uses internally to prioritize the packets.

The **IPQOS** filter itself consists of a range of DS field values, assigned to a priority level from 1 to 8. A **HIGH** priority level can also be configured, which is typically used for voice applications running on the NetPerformer.

► To define the IPQOS filter on the NetPerformer unit using the console:

- Enter the menu sequence: **SE** ↵ **FILTER**.
- Select the *FILTER number*.
- Set *Active* to **YES** to activate this filter.
- Enter the **IPQOS** filter definition at the *Definition* prompt.
- The *Definition* must have the following syntax:

  IPQOS(*min_value_of_range*,*max_value_of_range*)>*class_of_service*

  where:*min_value_of_range* is a 6-bit binary value representing the lowest DS field value

  max_value_of_range is a 6-bit binary value representing the highest DS field value

  class_of_service represents the NetPerformer class of service, and ranges from 1 to 8 plus the **HIGH** priority level usually used for voice.

NOTE: If two IPQOS filters are defined with an overlapping range of values, the overlapped portion is assigned to the filter with the lowest *FILTER number* (1 to 32).

SE/FILTER example: for DiffServ

```
CHICAGO>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) ? FILTER
FILTER number (1-32,def:1) ?
FILTER #1> Active (def:NO) ? YES
FILTER #1> Definition (def:) ? IPQOS(000000,010111)>3
```

To define the **IPQOS** filter on the NetPerformer unit via SNMP, use the equivalent SNMP variables: *filterEntry*, *filterActive* and *filterDefinition*. The *filterDefinition* requires the same syntax as for the console *Definition* parameter.

3.3 DiffServ Application Scenario

The most common DiffServ application is the prioritization of voice packets over data packets. Since voice applications have a very low latency requirement, these packets must be given greater priority than other packets to ensure the quality of the output.

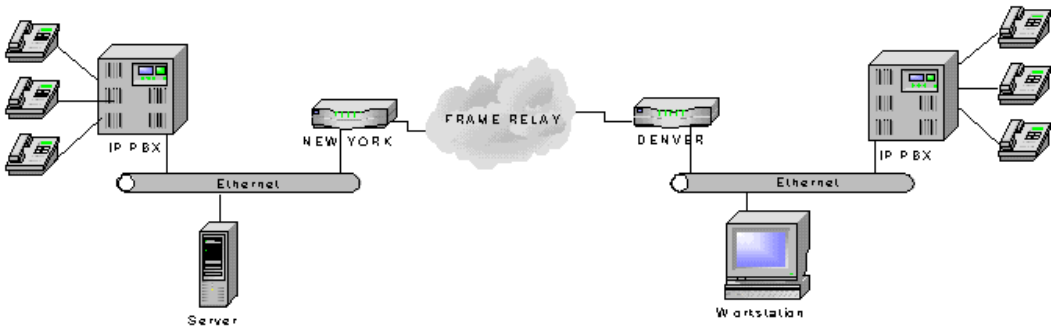

Figure 3-1: DiffServ Application

In this example, a Frame Relay link is used to transmit voice and data between one NetPerformer unit in New York and another in Denver.

- An IP PBX is connected to each NetPerformer via a local Ethernet LAN.
- A LAN connection at each site carries data traffic between workstations in Denver and the server in New York
- Each IP PBX uses voice packets containing a DS field set to high priority (**111011**)
- Both NetPerformer units have been configured with an **IPQOS** filter that assigns all voice packets carrying this DS field value to **HIGH** priority:
- The two applications (voice and data) coexist harmoniously on the same link, with priority always being given to the voice application. Data traffic is routed with a relatively larger latency.

SE/FILTER example: for DiffServ voice prioritization

```
CHICAGO>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) ? FILTER
FILTER number (1-32,def:1) ? 1
FILTER #1> Definition (def:) ? IPQOS(111011,111011)>HIGH
FILTER #1> Active (def:NO) ? YES
```

3.4 DiffServ Statistics Counters

New statistics counters have been added to the NetPerformer console to provide information on how much traffic has been routed at each priority level. These counters are grouped under the *Item* named **QOS**.

► To display the QOS counters:

- Enter **DC** at the NetPerformer console command prompt.
- Enter **QOS** at the *Item* prompt.

DC/QOS example

```
CHICAGO>DC
DISPLAY COUNTERS
Item (CONFIG/PORT/PVC/IP/BOOTP/TIMEP/SLOT/Q922/Q933/NAT/SVC/QOS,
def:CONFIG) ? QOS

Priority Level 1
Routed frames.....................................987
Routed bytes...................................654235
Priority Level 2
.
.
.
Priority Level 8
Routed frames.....................................456
Routed bytes...................................245545
Priority Level High
Routed frames.....................................786
Routed bytes...................................495434
```

4

Traffic Filters

4.1 SE/FILTER Configuration Parameters

FILTER number

Console	SNMP	Text-based Config
FILTER number	filterIndex	[filter#]

Table 4-1: Filter Number Parameters

The filter entry that you want to define.

Values: range is from *1* to the number of filters present on the system

Default: *1*

Definition

Console	SNMP	Text-based Config
Definition	filterDefinition	[filter#] Definition

Table 4-2: Definition Parameters

The filter definition. The required syntax depends on the traffic type being filtered. Refer to the list of filters on the following pages.

Values: Maximum of 128 alphanumeric characters

Default: undefined

Active

Console	SNMP	Text-based Config
Active	filterActive	[filter#] Active

Table 4-3: Active Parameters

Enables (**YES**) or disables (**NO**) the filter.

Values: *NO, YES*

Default: *NO*

The following is an alphabetical list of the filters that are currently supported by the Memotec NetPerformer, including a brief explanation and example of each. At the end of this list is an example of how filters can be combined using logical operators (see “Combinations” on page 4-25).

4.2 APPLETALK

Permits filtering Appletalk frames defined by:

- SNAP = 00000080F3
- SNAP = 080007809B
- Ethertype = 809B

Syntax: APPLETALK>n

- If $1 \leq n \leq 8$, all Appletalk frames are transmitted under class n.
- If n = X, all Appletalk frames are discarded.

Example: To assign Appletalk data to class 3, use the filter: **APPLETALK>3**.

4.3 DSAP

Permits filtering frames that have a DSAP (Token-Ring and Ethernet 802.3 only).

Syntax: **DSAP(HH)>n**

- *HH* is a hexadecimal value indicating the frame type. For example, HH = 04 for SNA, E0 for IPX, F0 for NetBIOS, and so on.
- If $1 \leq n \leq 8$, all frames with the specified DSAP are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To remove all NetBIOS frames, use the filter: **DSAP(F0)>X**.

4.4 ETHERTYPE

Permits filtering frames with Ethertype (valid for Ethernet frames only).

Syntax: **ETHERTYPE(HHHH)>n**

- *HHHH* represents the Ethernet hexadecimal frame type. For example, HHHH = 0800 for IP frames, 0806 for ARP frames, 809B for Appletalk frames, and so on.
- If $1 \leq n \leq 8$, all specified Ethernet frames are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To remove all IPX frames from Ethernet transmissions, use the filter: **ETHERTYPE(8137)>X.**

4.5 FRP

Permits filtering all Frame Relay over IP packets.

Syntax: **FRP>n**

- If $1 \leq n \leq 8$, all FRP frames are transmitted under class n.
- If n = X, all FRP frames are discarded.

Example: To prioritize Frame Relay traffic, use the filter: **FTP>1.**

4.6 FTP

Permits filtering frames that carry the FTP protocol. These are TCP frames that have a source or destination port equal to 0x14 (FTP Data) or 0x15 (FTP Control).

Syntax: **FTP>n**

- If $1 \leq n \leq 8$, all FTP frames are transmitted under class n.
- If n = X, all FTP frames are discarded.

Example: To assign FTP to class 2, use the filter: **FTP>2**.

4.7 IPDST

Permits filtering frames with a specific destination IP address.

Syntax: **IPDST(ip address, ip mask)>n**

- *ip address* is a valid IP address, for example, 201.168.48.0.
- *ip mask* is used to mask the IP address of the frame to be transmitted over the link. That is, the destination IP address of the frame is masked and then compared to the address defined in the filter.
- If $1 \leq n \leq 8$, all IP frames specified by the filter are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To assign all frames destined for network 201.168.43.0 to class 3, use the filter: **IPDST(201.168.43.0,255.255.255.0)>3**.

4.8 IPQOS

Permits differentiating between packets with different values in their DS fields. These values are then mapped to a configured traffic priority class, which the NetPerformer uses internally to prioritize the packets. Refer to “DiffServ 802.1 p/q ToS” on page 3-1.

Syntax: **IPQOS(min,max)>n**

- *min* is a 6-bit binary value representing the lowest DS field value
- *max* is a 6-bit binary value representing the highest DS field value
- If $1 \leq n \leq 8$, all packets specified by the filter are mapped to class *n*
- If n = **HIGH**, all packets specified by the filter are assigned the highest traffic priority level, usually used for voice traffic.

NOTE: If two IPQOS filters are defined with an overlapping range of values, the overlapped portion is assigned to the filter with the lowest *FILTER number* (1 to 32).

Example: To assign all packets with DS field values from 0 to 23, use the filter: **IPQOS(000000,010111)>3**.

4.9 IPSRC

Permits filtering frames with a specific source IP address.

Syntax: **IPSRC(ip address, ip mask)>n**

- *ip address* is a valid IP address, for example, 201.168.48.0.
- *ip mask* is used to mask the IP address of the frame to be transmitted over the link. That is, the source IP address of the frame is masked and then compared to the address defined in the filter.
- If $1 \leq n \leq 8$, all IP frames specified by the filter are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To remove all frames originating from network 200.168.56.0, use the filter: **IPSRC(200.168.56.0,255.255.255.0)>X**.

4.10 IPX

Permits filtering IPX frames, which are defined by:

- SNAP = 0000008137 (for SNAP 802.3 or 802.5)
- Ethertype = 8137 (for Ethernet II)
- SAP = DSAP = E0 (for 802.2)
- SAP = DSAP = FFFF (for 802.3 only)

Syntax: **IPX>n**

- If $1 \leq n \leq 8$, all IPX frames are transmitted under class n.
- If n = X, all IPX frames are discarded.

Example: To send all IPX frames under class 1, use the filter: **IPX>1**.

4.11 IPXSAP

Permits filtering IPX SAP routing table entries.

Syntax: **IPXSAP>n**

- If $1 \leq n \leq 8$, all IPX SAP frames are transmitted under class n.
- If n = X, all IPX SAP frames are discarded.

Example: To send all IPX SAP entries, use the filter: **IPXSAP>X**.

4.12 MACDST

Permits filtering frames with a specific destination MAC address.

Syntax: **MACDST(yyyyyyyyyyyy)>n**

- *yyyyyyyyyyyy* represents the MAC address.
- A mask is generated from this address.
- The destination MAC address of the frame is masked and compared to the MAC address defined in the filter.
- If $1 \leq n \leq 8$, all frames with the destination MAC address specified by the filter are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To discard all frames with a destination MAC address beginning with 0002083, use the filter: **MACDST(00002083XXXX)>X**.

- The address generated by this filter is 000020830000, and the generated mask is FFFFFFFF0000.
- If the destination address of a frame is 0000208300D4, then 0000208300D4&FFFFFFFF0000 is compared with 000020830000, and the frame is filtered.

4.13 MACQOS

Permits mapping the 802.1P priority level to a NetPerformer traffic class. This permits prioritization of VLAN traffic when bridging via PowerCell (PVCR).

Syntax: **MACQOS(min,max)>n**

- *min* is a 3-bit binary value representing the lowest VLAN priority value (taken from the *User Priority* field in the VLAN *Tag Header*)
- *max* is a 3-bit binary value representing the highest VLAN priority value
- If $1 \leq n \leq 8$, all packets specified by the filter are mapped to class *n*
- If n = **HIGH**, all packets specified by the filter are assigned the highest traffic priority level, usually used for voice traffic.

Example: To assign highest priority to all packets tagged with a *User Priority* field of 111, use the filter: **MACQOS(111,111)>HIGH**.

4.14 MACSRC

Permits filtering frames with a specific source MAC address.

Syntax: **MACSRC(yyyyyyyyyyyy)>n**

- *yyyyyyyyyyyy* represents the MAC address.
- A mask is generated from this address.
- The source MAC address of the frame is masked and compared to the MAC address defined in the filter.
- If $1 \leq n \leq 8$, all frames with the source MAC address specified by the filter are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To discard all frames with a source MAC address beginning with 0002096, use the filter: **MACSRC(00002096XXXX)>X**.

- The address generated by this filter is 000020960000, and the generated mask is FFFFFFFF0000.
- If the source address of a frame is 0000209600A3, then 0000209600A3&FFFFFFFF0000 is compared with 000020960000, and the frame is filtered.

4.15 NETBIOS

Permits filtering NetBIOS frames, which are defined by:

- SSAP = DSAP = F0
- NetBIOS can also be encapsulated under TCP with the source and destination port equal to 137,138 or 139. In this case, use a TCP filter, described later.

Syntax: **NETBIOS>n**

- If $1 \leq n \leq 8$, all NetBIOS frames are transmitted under class n.
- If n = X, all NetBIOS frames are discarded.

Example: To assign all NetBIOS frames to class 4, use the filter: **NETBIOS>4**.

4.16 SNA

Permits filtering all SNA frames transmitted over the link. SNA frames are defined by:

- SSAP is a multiple of 0x04
- DSAP is a multiple of 0x04

Syntax: **SNA>n**

- If $1 \leq n \leq 8$, all SNA frames are transmitted under class n.
- If n = X, all SNA frames are discarded.

Example: To send all SNA frames under class 1, use the filter: **SNA>1**.

4.17 SNAP

Permits filtering frames that have SNAP (Token-Ring and Ethernet 802.3 only).

Syntax: **SNAP(yyyyyyyyyy)>n**

- *yyyyyyyyyy* is a hexadecimal value. A SNAP (*sssssssss*) and a mask (*mmmmmmmmmm*) are generated from this value.
- If y = X, the corresponding m and s values = 0.
- If y ≠ X, then s = y and m = 0xF.
- Example values are 0000000800 for SNAP IP, 0000000806 for SNAP ARP, 0000008137 for SNAP IPX.
- If 1 ≤ n ≤ 8, all frames with the specified SNAP value are transmitted under class n.
- If n = X, all specified SNAP frames are discarded.

Example: To discard a specific subset of Appletalk frames, use the filter:
SNAP(080007809B)>X

4.18 SNMP

Permits filtering SNMP frames, that is, UDP frames with the source and destination port equal to 161 or 162.

- SNMP transported under TCP/IP is not supported by this filter. For these frames, use a TCP filter, described later.

Syntax: **SNMP>n**

- If $1 \leq n \leq 8$, all SNMP frames are transmitted under class n.
- If n = X, all SNMP frames are discarded.

Example: To assign SNMP to class 2, use the filter: **SNMP>2**.

4.19 SSAP

Permits filtering frames that have an SSAP (Token-Ring and Ethernet 802.3 only).

Syntax: **SSAP(HH)>n**

- *HH* is a hexadecimal value indicating the frame type. For example, HH = 04 for SNA, E0 for IPX, F0 for NetBIOS, and so on.
- If $1 \leq n \leq 8$, all frames with the specified SSAP are transmitted under class n.
- If n = X, all specified frames are discarded.

Example: To remove all SNA frames, use the filter: **SSAP(04)>X**.

4.20 TCP

Permits filtering TCP/IP frames with a destination port equal to or included in the range defined by the filter.

Syntax: **TCP(port1,port2)>n**

- *port1* and *port2* specify the range of destination ports in the TCP frame to be transmitted over the link. Up to 5 digits are allowed for the port fields, each of which can have a maximum value of 65535.
- If $1 \leq n \leq 8$, all TCP/IP frames within the specified range are transmitted under class n.
- If n = X, all specified TCP/IP frames are discarded.

Examples:

- To discard SNMP frames under TCP/IP, use the filter: **TCP(161,162)>X**
- To assign NetBIOS under TCP/IP to class 2, use the filter: **TCP(137,139)>2**

4.21 TELNET

Permits filtering all frames carrying the TELNET protocol. These are TCP frames with a source or destination port equal to 0x17.

Syntax: **TELNET>n**

- If $1 \leq n \leq 8$, all TELNET frames are transmitted under class n.
- If n = X, all TELNET frames are discarded.

Example: To transmit TELNET under class 3, use the filter: **TELNET>3**.

4.22 TN3270

Permits filtering all frames carrying the TELNET protocol used with IBM 3270 or similar equipment.

Syntax: **TN3270>n**

- If $1 \leq n \leq 8$, all TN3270 frames are transmitted under class n.
- If n = X, all TN3270 frames are discarded.

Example: To transmit TN3270 under class 3, use the filter: **TN3270>3**.

4.23 UDP

Permits filtering UDP frames with a destination port equal to or included in the range defined by the filter.

Syntax: **UDP(port1,port2)>n**

- *port1* and *port2* specify the range of destination ports in the UDP frame to be transmitted over the link.
- If $1 \leq n \leq 8$, all UDP frames within the specified range are transmitted under class n.
- If n = X, all specified UDP frames are discarded.

Example: To discard SNMP frames, use the filter: **UDP(161,162)>X**.

4.24 Combinations

You can combine two or more filters with the operators **|** (OR), **&** (AND) and **!** (NOT) to define more specific filters.

Example: **IPDST(201.168.43.0,255.255.255.0)&MACSRC(002083XXXX)>X**

- This filter will discard only those IP frames that are destined for network 201.168.43.0 and that have a source MAC address beginning with 00002083.
- A station with source MAC address 00222200001 in network 201.168.43.0 will continue to receive all frames specifying its address as the destination.

Index

For local offices and sales representatives, visit our website:

www.memotec.com

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

COMTECH EF DATA

MEMOTEC
redefining network efficiency