

QoS Customer Edge Devices

Research Report

April 4, 2001

Executive Summary

Over the last couple of years, the level and complexity of Internet traffic has increased as more people have gone online. With this increase, it has become necessary for MOREnet to consider methods whereby we can ensure that priority data is consistently delivered. Ensuring this consistent delivery is known as Quality of Service (QoS). Because some traffic requires “end-to-end” reliability, the overall effort to investigate QoS methods involves looking at our core network as well as devices or methods to be implemented at the customer site.

This project evaluates four devices for possible implementation at the customer site. Distributors graciously provided the devices to MOREnet for the evaluation period. The four devices evaluated are listed below:

Company	Product
Allot	NetEnforcer 301
NetScreen	NetScreen 1000
Packeteer	PacketShaper 4500
TopLayer	AppSwitch 3500

The devices were evaluated on their ability to detect, shape, and label traffic as required by MOREnet QoS initiatives. Specifically, MOREnet and its sponsors are interested in providing consistent delivery of video or other high priority traffic. MOREnet also recognizes that low priority traffic may have to be contained by rate-limiting or low-priority marking. For testing purposes H.323 video was chosen as a type of traffic to be protected and Napster and Gnutella were chosen as types of traffic to be contained. Some products tested have capabilities beyond what was tested such as firewalling, address translation (NAT), etc.

All products performed well in shaping traffic by IP network or port number. However, video and Napster/Gnutella cannot be identified by port due to the nature of the protocols. If sources of video and Napster-like traffic are always associated with specific network addresses, then this method may be adequate. This would require administrative overhead in maintaining tables of networks in our hub routers for specific bandwidth management.

Priority marking of packets by application allows for less administration in our core network. The impact of this approach on router performance has yet to be determined. This impact will be studied in an upcoming research project. In this study we specifically looked for methods that identify/shape and mark video and Napster-like applications by something beyond simple port number. The TopLayer AppSwitch and the Packeteer

PacketShaper can identify/shape/mark traffic by these methods with some limitations. We found the AppSwitch to be the best at identifying/shaping/marketing video traffic and the Packeteer at identifying/shaping/marketing Napster-like traffic.

Technical Background

Quality of Service

Currently the default service on the Internet is “best-effort” which treats all network traffic equally. When network congestion occurs, the quality of certain applications may decline to unacceptable levels. The term Quality of Service (QoS) describes a goal for network performance to ensure the appropriate delivery of high priority traffic vs. low priority traffic. MOREnet has been investigating options for providing the Quality of Service required by our customers/sponsors. This investigation evaluates QoS devices that could be used by our customers to prioritize traffic before it gets to the MOREnet backbone.

There are a variety of traffic type-based solutions to provide QoS, so it is necessary to provide a brief summary of service responses to traffic demands. One response is to guarantee a level of service for consistent delivery. Unfortunately this approach is not efficient because unused bandwidth is not available to other traffic. Another response provides relative, preferential treatment to packets with no guarantee of service (this is also referred to as Class of Service). Prioritized bandwidth methods typically allow low priority traffic to burst when capacity is available. Traffic can be assigned a priority based on variables such as protocols, ports, applications, network addresses, users, etc. Prioritized traffic is then typically shaped by different queuing strategies. This approach takes the pressure off of the core backbone to control traffic on a case-by-case basis and is the preferred method for large networks.

The primary QoS needs are congestion and latency control. When a network is congested (i.e. at a router or switch port), packets have to be queued for transmission and may be dropped if its queues overflow. For some traffic flows (like IP based video) a consistent latency must be maintained. This requires priority-based queues even if the queues never fill. Although QoS needs can be honored on many different devices, the actual method used will vary by device and device configuration. For example, one response may control traffic through priority queuing while another may control the size of the TCP window.

TOS/DiffServ

IP version 4 design provides a Type of Service (TOS) byte which has been used to mark prioritization or special handling. Specific Class of Service (CoS) methods utilize the bits from this byte to determine the service level of the packet. In the traditional model, the first three bits were set for precedence (priority), the next four bits determine how the traffic should be moved through the network, and the remaining bit is unused. The IETF has developed a “differentiated services” (DiffServ) model that would be scalable and provide consistent service classes independent of application. DiffServ uses the first six bits of the TOS byte to define how specific packets should be moved through the network. A third Class of Service method, called 802.p, is typically used with VLANs. 802.p marking was not tested in this project.

TOS

The type-of-service (TOS) byte in an IP header specifies precedence (priority) and type of service (RFC791, RFC1349). The precedence field is defined by the first three bits and supports eight levels of priority. The lowest priority is assigned to 0 and the highest priority is 7. The values 6 and 7 are reserved for network control packets so the values 0 through 5 can be set for priority based on IP networks or applications.

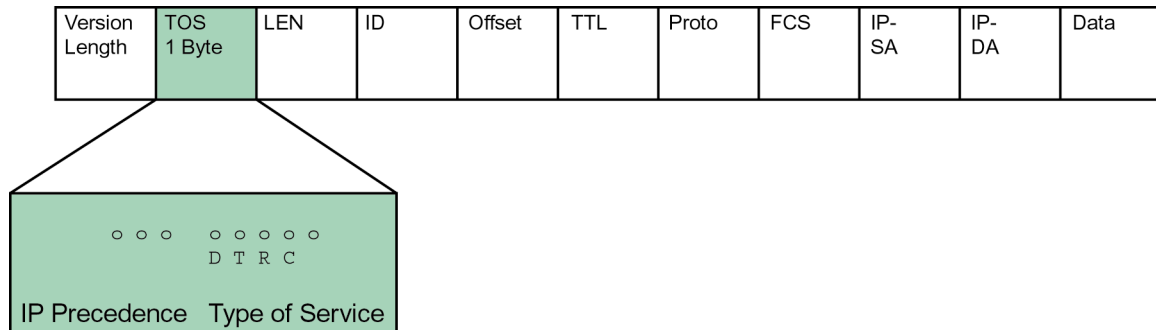


Figure 1: Standard IP header information

The next four bits determine the type of service. Only one of these bits can be turned on. Each bit determines a specific method for the router to select a path.

- D** The delay bit tells the router to choose high speed to minimize delay. This bit would typically be set for voice, video, telnet and rlogin.
- T** The throughput bit specifies high capacity links used for bulk transfers.
- R** Routing protocols and network management applications use this for fault tolerant paths.
- C** The cost bit is for low priority applications such as NNTP and signifies the lowest cost path.

The last bit of the byte is unused and always set to 0. Best effort delivery is used if all bits of the TOS byte are set to 0.

Differentiated Services

This model was developed to provide simple differentiation of traffic such that the traffic's relative priority could be determined on a hop-by-hop basis as opposed to maintaining end-to-end flow states that consume network resources. Traffic is classified and assigned to a behavior aggregate by marking the DS field with a Differentiated Services Code Point (DSCP). The definition and format of the DS field is described in RFC2474. The DSCP then triggers a per-hop behavior (PHB) from the components of the network.

DiffServ reclaims the TOS byte and uses the first 6 bits to mark the DSCP, which is then mapped to the PHB. Service providers have control over how the codepoints are mapped to PHBs, and each time a packet enters a network domain it may be remarked.

Sample values for DiffServ mapping are shown below (from NEC, Europe Ltd.)

RTP with = 64 kbit/s audio	EF (premium service)
Other R/T streaming video/audio	AF1
Non R/T audio/video	AF2
Other TCP	AF3
Other	Best effort

Basic Issues with Traffic Identification

MOREnet customer needs are varied, but video/voice quality is of importance to everyone as well as controlling Napster-like applications. We tested the products on their capability to identify, mark, and shape traffic generated by these applications.

Internet applications vary considerably in the method of communication between hosts. Table 1 shows approximately how the protocols we investigated fit into the OSI/DOD models of networking.

Table 1: OSI and DOD Networking Models

Application										
Presentation	Process/ Application Host to Host									
Session		F T P	T e l n e t	S M T P	H . 3 2 3 S t a r t	N a p s t e r	D N S	S N M P	T F T P	R T P , R T C P
Transport	Internetwork	TCP				UDP				
Network	Network Access	IP & ICMP								
DataLink		IEEE802.2, Token Ring, FDDI								
Physical		Ethernet, leased lines, UTP								

All products tested could control traffic at the network or transport layers. Traffic can be shaped at the network layer if specific IP addresses or subnets are determined. Traffic can be shaped at the transport layer if it can be identified by well-known ports such as 80, 25, etc. The Cisco routers at our customer sites also have this capability. To identify and shape traffic that is not identifiable by a well-known port, a device must interpret information above the transport layer.

TCP is connection oriented and, by its nature, guarantees delivery of data. On the other hand, UDP is connectionless and, although faster, cannot guarantee delivery of data. Many applications use a combination of both protocols when transferring data. We tested video using H.323 which includes sub protocols that are TCP or UDP. Once a video session is established via TCP, a series of ports are negotiated dynamically and finally the data is sent as RTP over UDP. For more information on the H.323 protocol go to: http://support.intel.com/support/videophone/trial21/h323_wpr.htm#a5.

Napster and similar programs such as Gnutella do not conform to well-known ports. Napster traffic starts as a request from a client to Napster.com. The request is then redirected to one of several servers with a database of registered Napster clients that are sharing files. Typically, servers listen on ports 5555, 6666, 8888, etc. but not always. A

program called Napigator will give a list of redirect servers and the ports they are currently listening on. Napigator also provides the option to connect directly to these servers without going through Napster.com. Each workstation running Napster becomes a file server and the user can configure it to listen on any port. This makes the file transfers difficult to track. Napster uses only TCP and the class of service for these exchanges is typically categorized as bulk file transfer. For more detailed information on Napster please refer to:

<http://david.weekly.org/code/Napster.php3> or <http://www.Napster.com>

Gnutella is one of many applications that facilitates peer-to-peer file sharing of multimedia files. To register, the user must know the IP number of any Gnutella “servant” (other Gnutella user) on the Internet. The new user then becomes part of the Gnutella network and may download or share files. At this time only TCP is used for file transfers. For more about Gnutella refer to:

<http://www.rixsoft.com/Knowbuddy/Gnutellafaq.html>. Because of Gnutella and other publicly available programs, the current litigation against Napster.com will have no effect on the growth in peer-to-peer sharing of large multimedia files.

Project Objectives

For this project, we assessed four products for their ability to shape network traffic between a local network and the Internet. We were particularly interested in identifying and maintaining QoS for H.323 video and limiting Napster-type traffic. We used both Napster and Gnutella clients to generate traffic and set up an H.323 video connection between two Polycom units. In addition to this traffic, we periodically sent large amounts of typical Internet traffic generated by Chariot software across the network.

We tested each product's ability to identify, mark, and shape traffic as well as limit bandwidth. We also noted additional features such as LDAP compatibility, firewall features, VLAN capability, etc. The features of each product are shown in Appendix A. All products performed well within the scope of their capabilities. The next section gives a brief overview of each product and its specific features.

Top Layer AppSwitch

The Top Layer AppSwitch device is designed to be a multi-zone firewall and shapes traffic based on policies defined between zones. Physically, the box is a 12 to 14 port switch. Top Layer's method of identifying traffic is called "7-Layer Application Control" which actually uses information from all layers of the OSI model to identify and shape traffic. Through stateful inspection of control ports and control protocols, the product monitors the flow of traffic per session. When a session (such as H.323) starts on a well-known port, the AppSwitch applies the policy for that class of service to the whole flow during the session, including RTP/UDP data transfers. Once the connection is made the device follows the session as dynamic ports are assigned. For more information on TopLayer's "7-layer Application Control," refer to:
http://www.toplayer.com/research/whitepapers/the_need_8_25.html#_Toc457104485

The AppSwitch manages bandwidth through several methods of QoS as determined by the user. Policy Sets are applied to "zones" as described below.

Zones are logical groupings of resources with similar requirements that are not restricted to physical location on the network. Zones are defined by networks, hosts, ports, or users, and a "Policy Set" is then determined between two zones. Although this device has several firewall features, these were disabled for testing and the policies defined for our test zones determined QoS for specific applications. Each policy set is comprised of one or more policies defining the Quality of Service for specific applications. The Application Definition Library (ADL) contains over 400 predefined applications that are identified by layer 4 through 7 "characteristics."

Another application, "TopFlow," can be run on a workstation and provide usage data to an MS Access database. This application could be run for network usage analysis, troubleshooting, etc.

The user may add new applications and may define policies for specific applications. A policy set is then defined to include all policies relating to the transfer of data between two zones.

When defining a policy for an application a “service class” must be chosen. Service Class attributes include:

- Name** — Description; e.g., Denial of Service, Best Effort, etc.
- QoS Type** — Weighted Priority, Guaranteed Bandwidth, Graduated Priority, Limited Bandwidth.
- Parameters** — Depends on the type of QoS selected.

Figure 2 shows an example of a service class configuration screen.

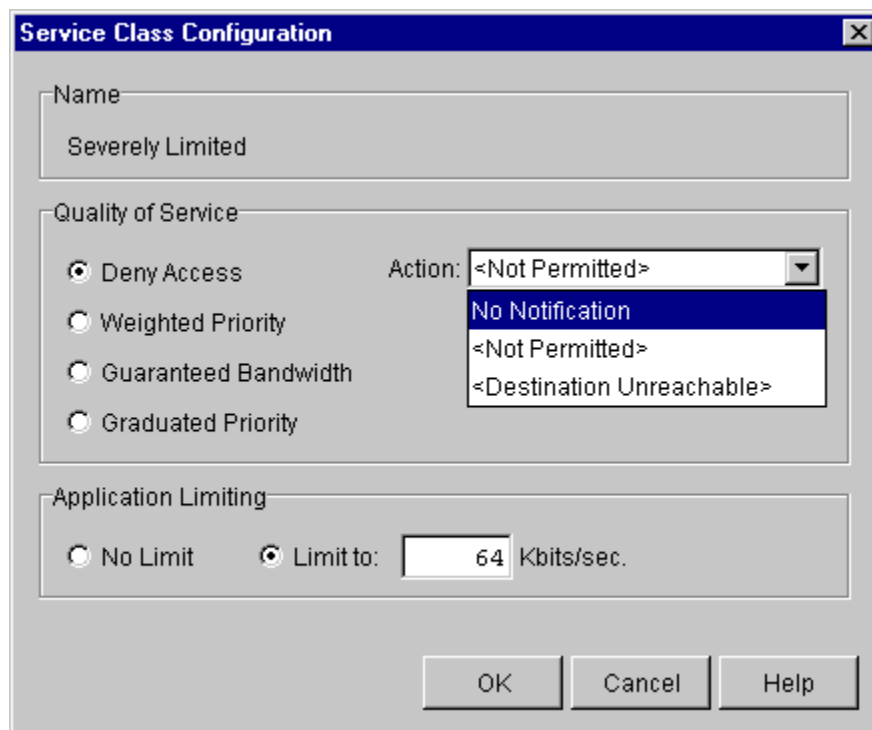


Figure 2: Top Layer Service Class Configuration

Graduated priority defines specific conditions under which queuing will occur. Weighted priority can use TOS, DSCP or 802.1p to queue traffic by marked priority. Guaranteed bandwidth specifies a minimum amount of bandwidth that will always be available for a specific application. Application limiting controls individual flows within the QoS group.

For testing purposes, we defined the zone “dormitories” and Napster and H323 policies were created. A Policy Set was then applied to the intersection of the dormitories and world zones. Once this was configured, packets were redirected to a workstation for real time analysis and logging.

For additional information on this product go to: <http://www.toplayer.com>

NetScreen

The NetScreen Device has a unique internal architecture that is comprised of up to 6 Processor modules, a Switch module, and an Auxiliary module. Each processor module has a RISC processor and NetScreen’s GigaScreen ASIC. The switch module provides gigabit access to the networks and the primary processor module passes traffic to the other processor modules as necessary, based on defined rules. The auxiliary module is for management and backup.

This device is primarily designed to be a firewall and supports VPNs as well. The traffic shaping feature prioritizes packets by network or port and supports guaranteed bandwidth as well as priority queuing. The NetScreen does not fully meet MOREnet’s requirements; it cannot change the TOS byte of the IP header, so we did not test this product as extensively. However, it may be a great product for certain Enterprise networks that need a firewall that can shape traffic. Figure 3 shows some of the features available for controlling traffic with NetScreen. For more information about the product go to: <http://www.netscreen.com>.

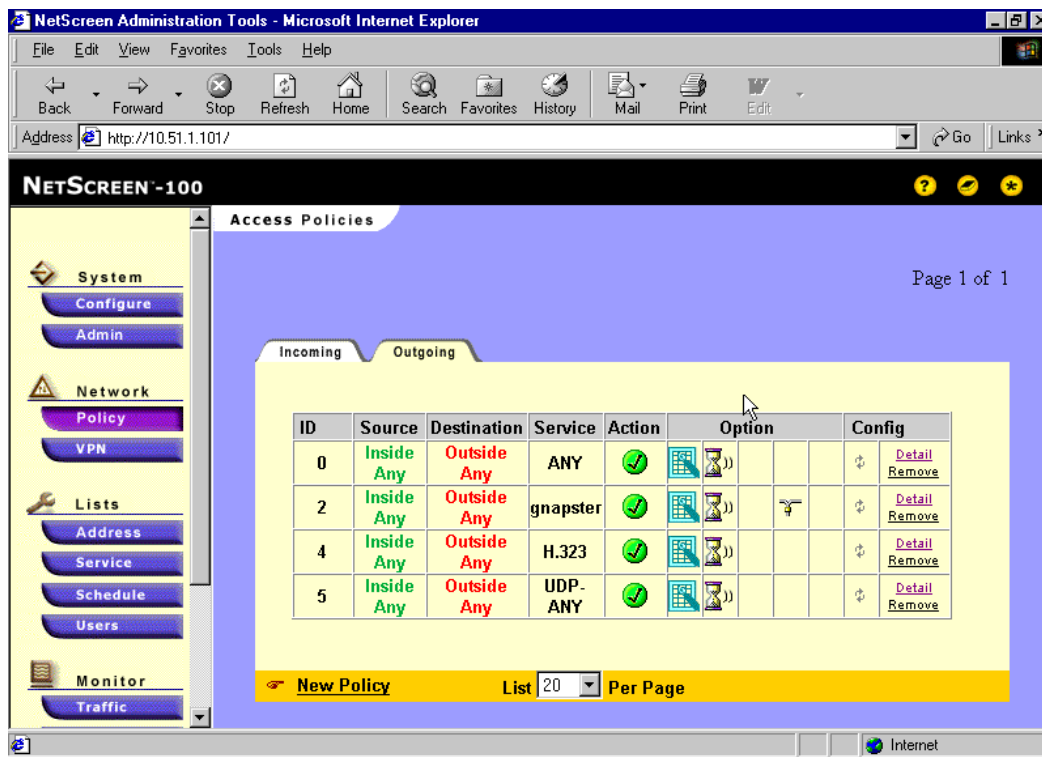


Figure 3: NetScreen Access Policies

Allot NetEnforcer

Allot’s NetEnforcer product is primarily a traffic management device. It is logically based on “Virtual Channels (VC)” which are defined bandwidth pipes for aggregate types

of data. The device can mark and shape traffic based on network, ports and users. The product has a “Per-flow” queuing method that allows all high priority flows to be maintained. For more information about this product please go to: <http://www.allot.com>

This device works at layer 4 and can mark packets by either DiffServ or standard TOS. The channels were easy to define and the real time monitoring feature was good but requires a specific Java runtime version. The Java version caused a few problems. In general the monitor and VC configuration modules took an extremely long time to load. The service catalog contained several well-known protocols but not H.323 or Napster. Users can define Napster in the catalog by known Napster ports but not by more consistent markers.

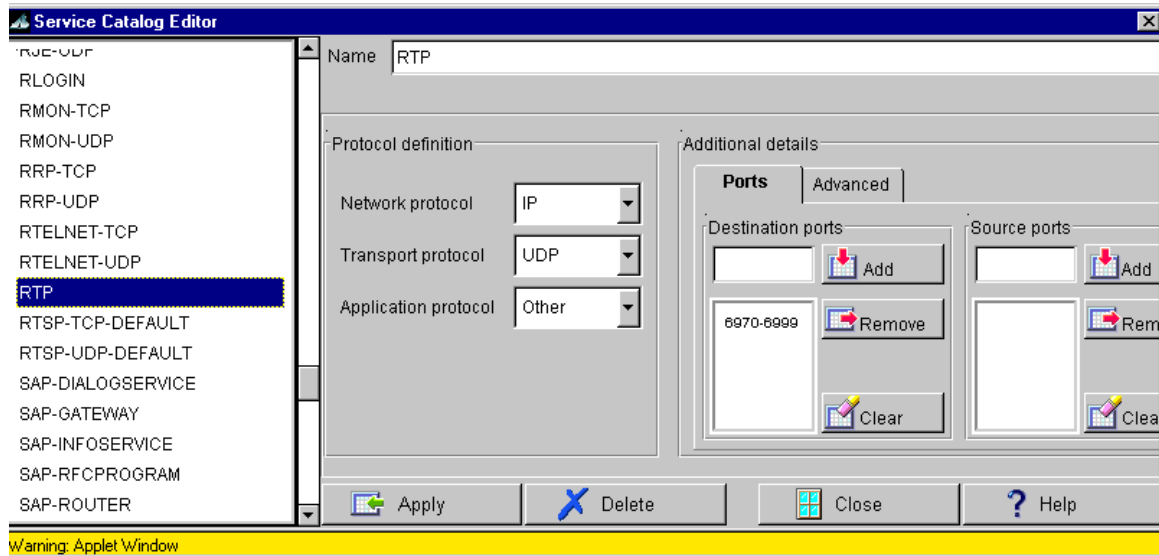


Figure 4: NetEnforcer Service Catalog

Packeteer PacketShaper

This device shapes TCP traffic by controlling the size of the TCP window and handshaking as needed based on policies defined for service classes. UDP traffic is managed by queuing. The product can recognize over 200 types of traffic, including Napster and Gnutella. When necessary, this product has the capability to examine the data portion of the packet for a “signature” from Napster and Gnutella that identifies the applications. A set of service classes are predefined for applications, but users may create their own based on application or network information. The product can rate-limit, deny, prioritize by TOS or DiffServ, and provide guaranteed bandwidth. Policies are read in specific orders which the user may change based on individual needs.

Service classes are arranged hierarchically with some classes having subsets of protocols or rules. With H.323 there is a subset of rules that apply for individual protocols within the H.323 protocol. This is particularly important, as Packeteer handles TCP and UDP differently. Figure 5 shows the Packeteer monitor screen with no policies defined. The

RTP traffic is H.323 video going across the network. Once the H.323 session has been established, the traffic is displayed as RTP (Real Time Protocol).

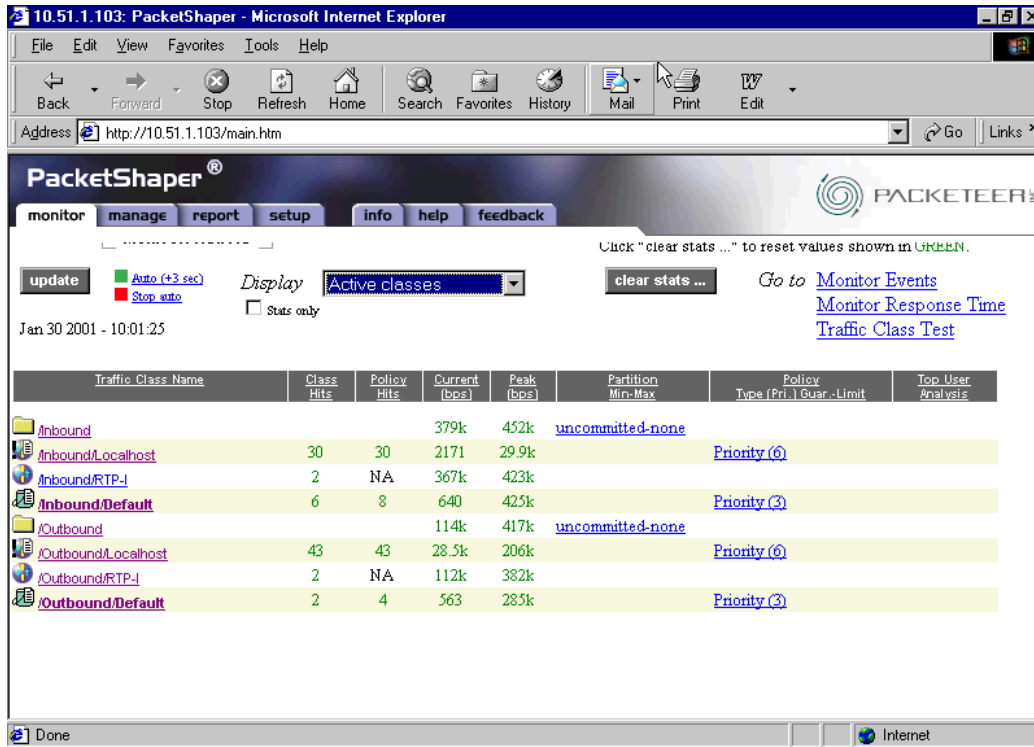


Figure 5: PacketShaper monitor

Testing and results

Although each of these products has outstanding features, we were only interested in testing specific capabilities due to the scope of our research. The objectives of this testing were:

- To determine if the device can identify traffic by application, specifically H.323 and Napster.
- To determine if the device can limit the aggregate traffic to the router.
- To determine if the device can mark packets using the TOS field for TOS or DiffServ.
- To determine if the device can shape the traffic according to policy.

To generate video traffic for the various devices, we established an H.323 session using two Polycom Viewstation v.35 units. For Napster and Gnutella, we downloaded the clients and tested by downloading files from the Internet and from workstation to workstation on different ports. Other flows were generated by Ganymede Chariot software to simulate heavy Internet traffic (shown in Figure 6 below).

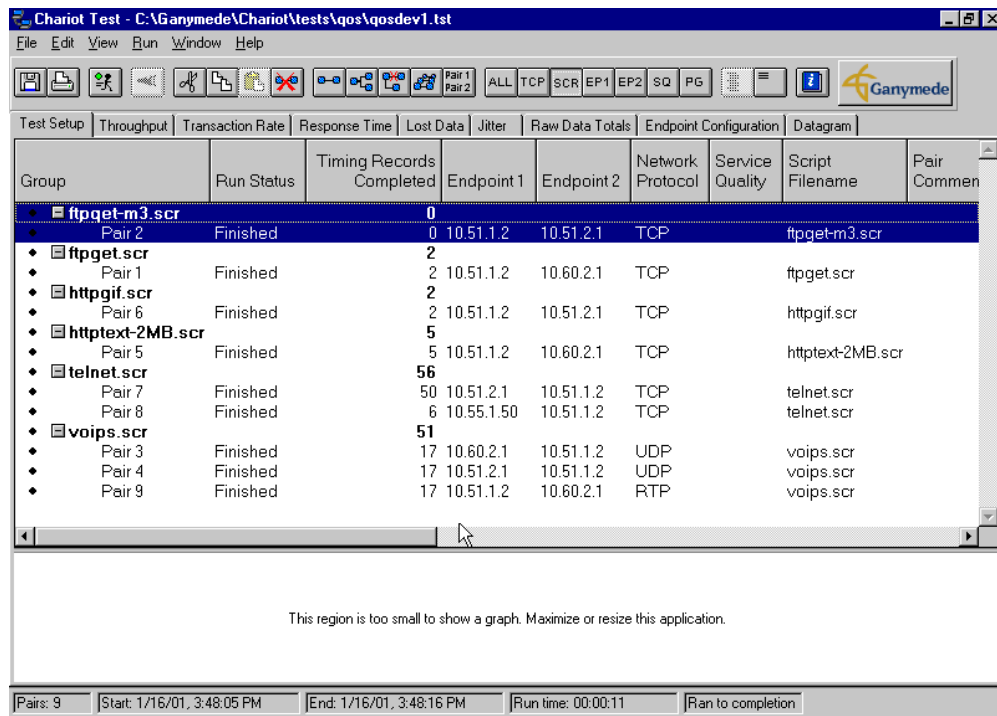


Figure 6: Chariot tests

Controlling rates on the physical ports:

We tested each device for aggregate bandwidth control on a physical port to see how closely they kept the network traffic to the desired level. We again used the Chariot suite

of tests to evaluate control at 4 and 8 MB. All devices performed fairly well and were easy to configure. The Top Layer device was the most consistent and its throughput is shown in Figure 7.

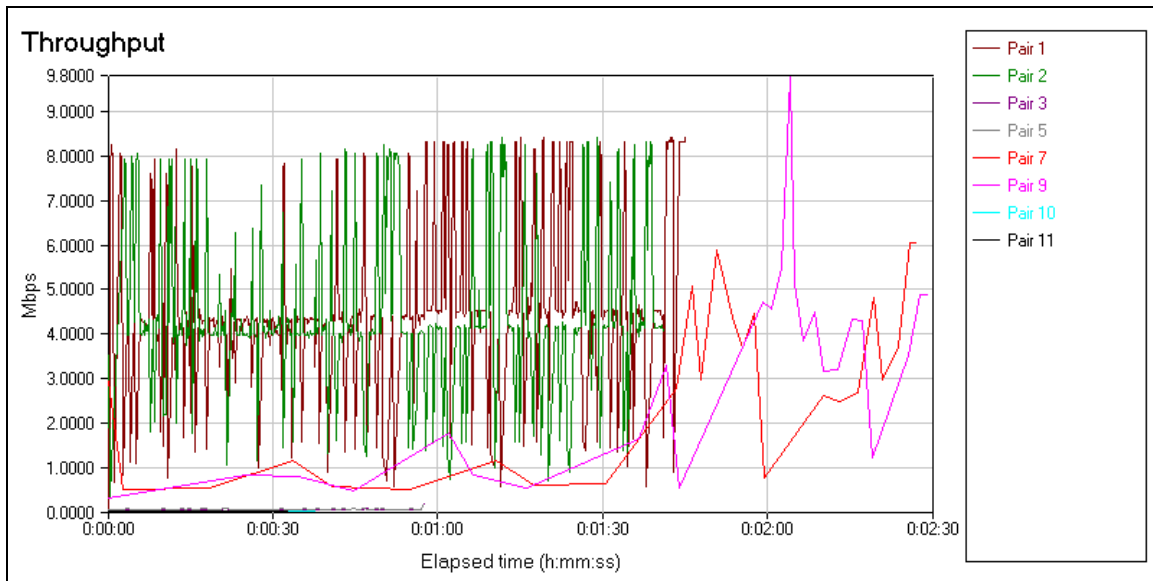


Figure 7: Top Layer Port at 8Mb

Application Recognition

We evaluated how each product recognized applications by observing their monitoring screens as traffic was flowing through the systems. PacketShaper, NetEnforcer, and AppSwitch all have real time displays using pie charts or bar graphs. NetScreen only shows one type of traffic at a time in a linear graph. AppSwitch also has an application called TopFlow that can be installed on a workstation and used to write detailed information to an MS Access database. All the tested products have the capability to provide at least a minimal level of logging.

This was perhaps the most interesting and enlightening part of the project. NetScreen was the most limited in that it only monitored and logged traffic as defined by specific rules (see Figure 3) using IP numbers or ports.

AppSwitch has a real-time monitor that can be turned on to show traffic between 2 zones where policies apply. Although the device can keep track of an entire H.323 session for shaping, the traffic is still identified by protocol. In Figure 8 the UDP traffic on ports 49xxx reflects the data portion of the H.323 traffic.

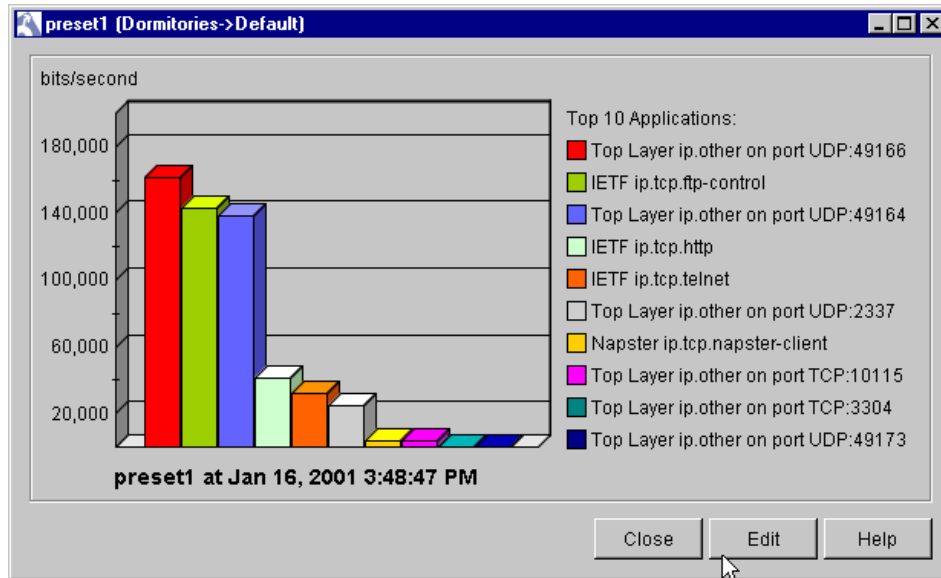


Figure 8: Top Layer Traffic recognition

In Figure 8, the Napster traffic was identified as classified below (see figure 9) in the Application Definition Library. The device could not identify Napster unless the specific ports were defined in the service class. This would appear to be a limitation of the “7 layer application control” approach, as Napster is not a standard application and the ports it uses vary. The device could not identify Gnutella or any similar type program that is not initiated on a port as defined in the ADL. The AppSwitch does have the ability to examine the packet beyond the header but this ability was not used for its Napster recognition. According to TopLayer, Gnutella and better Napster identification have been included in later releases.

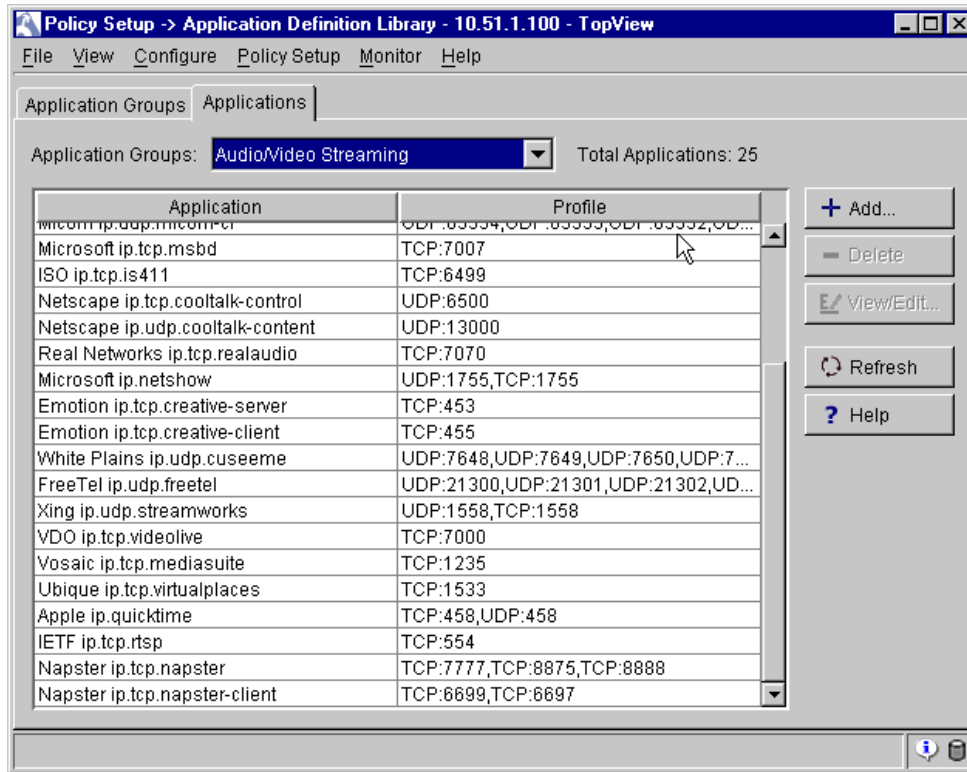


Figure 9: Napster Application Definition

Currently, Napster-generated traffic using ports other than those defined in Figure 9 will not be identified as Napster. Even though the H.323 is displaying as multiple protocols, in a session, this traffic will still be shaped as the AppSwitch tracks the session negotiation and applies the policy to this portion as well. It makes it a little more difficult to use the program to analyze data on the network but administrators typically have a handle on their video data.

The NetEnforcer represented traffic in a similar way. If the traffic was actually using the ports as defined in the catalog then it was correctly displayed. Unlike the AppSwitch however, the NetEnforcer was not designed to capture control port information and follow the session regardless of the port. Virtual channels must be defined by the user for specific services by IP network address and/or port. If the service is not identified correctly then the virtual channel monitoring won't be informative. The same thing is true for H.323. Because H.323 uses multiple protocols and dynamic ports, we could not set up a virtual channel for H.323 on this device. Figure 10 shows a NetEnforcer display of video traffic. The traffic shown is video but because the data ports are assigned dynamically we cannot display it as a channel. This device is not strongly geared towards recognizing applications. Unfortunately the applications we are interested in cannot be easily displayed. Otherwise the monitoring function is actually excellent in this product.

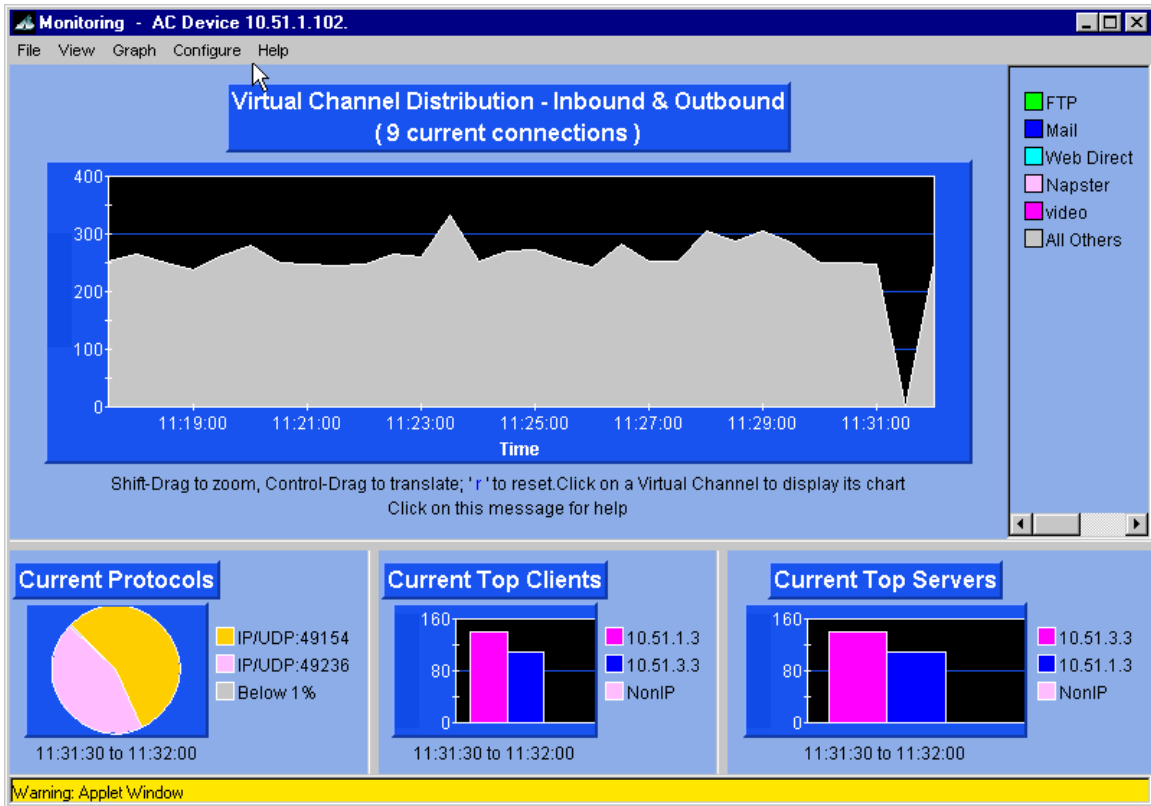


Figure 10: Video traffic on NetEnforcer

The Packeteer device did identify Napster and Gnutella traffic. However, a problem occurred when we set a Napster client to share on port 80 and then pointed another Napster client to it. The PacketShaper recognized it as HTTP due to the order of their traffic recognition. This can be altered by the user as necessary and has been changed in a recent software update. H.323 displayed up as H.323 protocol only during call setup. The actual video transfers displayed as RTP and RTCP. The device switches from TCP flow control to queuing for UDP (including RTP and RTCP).

An interesting discovery during the testing was the amount of NetBIOS traffic that Napster generates while trying to find different users. It appears that Napster clients sometimes try to resolve the NetBIOS names of their peers even across a WAN link. It might be useful to actually deny ports 137 and 139 on sites with high Napster usage.

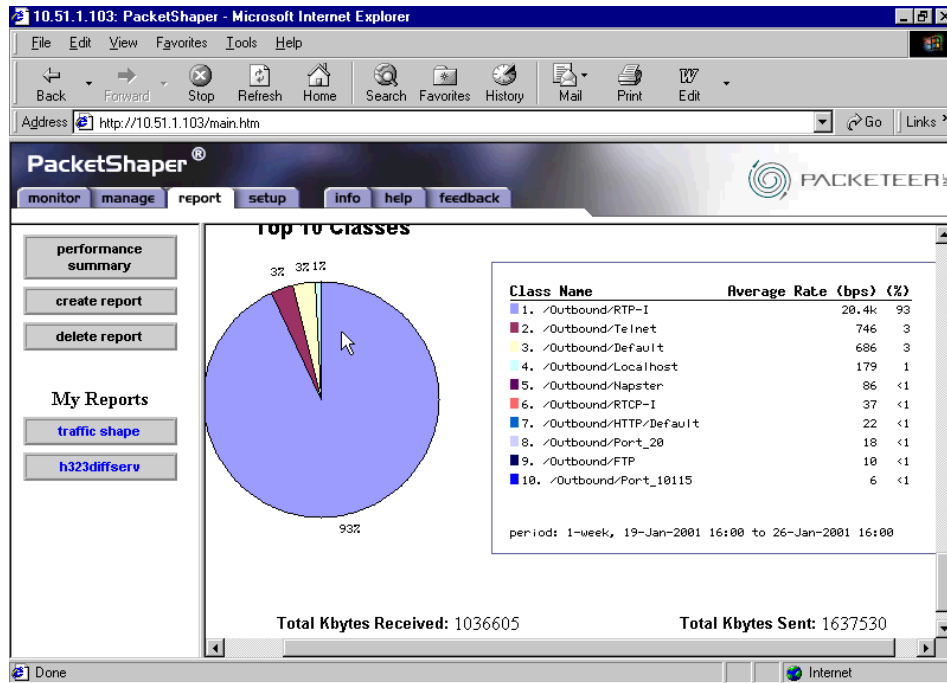


Figure 11: PacketShaper Traffic Identification

Packet Marking

We tested the products' packet marking abilities by setting policies for different types of packets and observing the packet information from a Radcom protocol analyzer. For more information about Radcom products, go to: <http://www.radcom-inc.com>. Three of the four products were able to mark the TOS byte: NetEnforcer, PacketShaper and AppSwitch. All of these products can mark for standard precedence/TOS or DiffServ. The NetScreen device has an option to prioritize traffic but cannot mark the TOS byte. This would not scale outside of an Enterprise network, as an Internet standard is required to prioritize packets for other systems.

Marking with the Packeteer device proved to be a little complicated. It was originally assumed that individual rules within the traffic classes would have the same policies filter down to them. Within the H.323 class there are 8 rules (sub-protocols). When setting a policy for H.323 TOS marking, we found that it did not apply to the UDP protocol listed under H.323. To mark the UDP video packets, it is necessary to mark all RTP packets with the same priority or mark by specific IP numbers. The NetEnforcer shows this problem as well. This finding was discouraging and will have to be addressed if these devices are to be implemented in the field.

At this point it may be prudent to summarize what we have so far. Three of the four products can mark packets based on specific IP numbers. Since marking packets is one of the three objectives of this study we did not include the NetScreen product in our shaping tests.

AppSwitch can identify and shape H.323 throughout the entire session but the others can only prioritize and shape RTP components, separate from the TCP components. The PacketShaper can successfully identify Napster and Gnutella, unless they choose a well-known port. The rest of the tests, including shaping by priority, rate-limiting, and guaranteed bandwidth, all depend somewhat on the way each device recognizes traffic.

Shaping/Rate-Limiting

All of the tested products can deny completely or rate-limit traffic based on IP address. NetEnforcer and AppSwitch can rate-limit Napster if the session occurs on a known Napster port (6666, 8888, etc.), while Packeteer can rate-limit Napster and Gnutella traffic on a variety of ports but not on a well-known port (such as port 80).

Figure 12 shows a Napster session that has just begun, emphasizing the initial NetBIOS flood. The AppSwitch identified Napster in this case and rate-limited even the NetBIOS traffic to 32k correctly. However if the session had not initiated on a port defined in the ADL it will not have been able to control associated sub-protocols within the session.

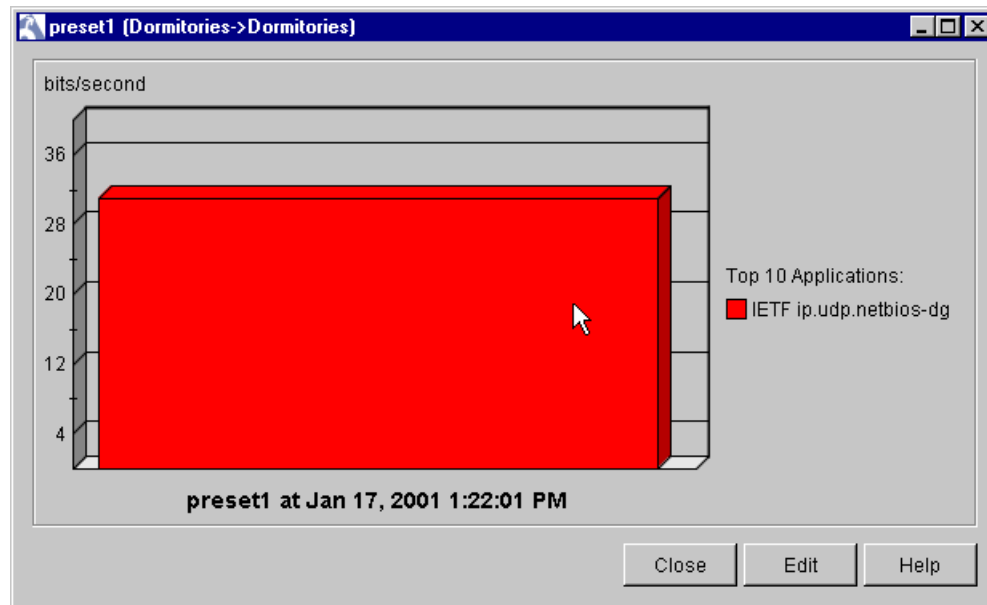


Figure 12: Napster 32k

The same criterion applies to guaranteed bandwidth. The product's capabilities are directly related to the method by which it identifies traffic. All the devices can identify traffic flows by IP number. Only the AppSwitch could guarantee H.323 bandwidth for all sub-protocols through an entire session. It might be enough to rate shape RTP traffic for MOREnet customers. This would need to actually be tested in the field.

Traffic Shaping

All the products can shape traffic using a combination of rate-limiting and/or queuing priority. However, the effectiveness of the shaping depends on the ability of the product

to classify the traffic. Figure 13 shows Top Layer with video at high priority; Napster down to 0 priority, and any other traffic at best effort. The UDP traffic that is shown lower than Napster is very small session setup traffic. The same device may have to shape by IP addresses if Napster was set to use non-standard ports. The Napster traffic would be treated as best effort if not recognized.

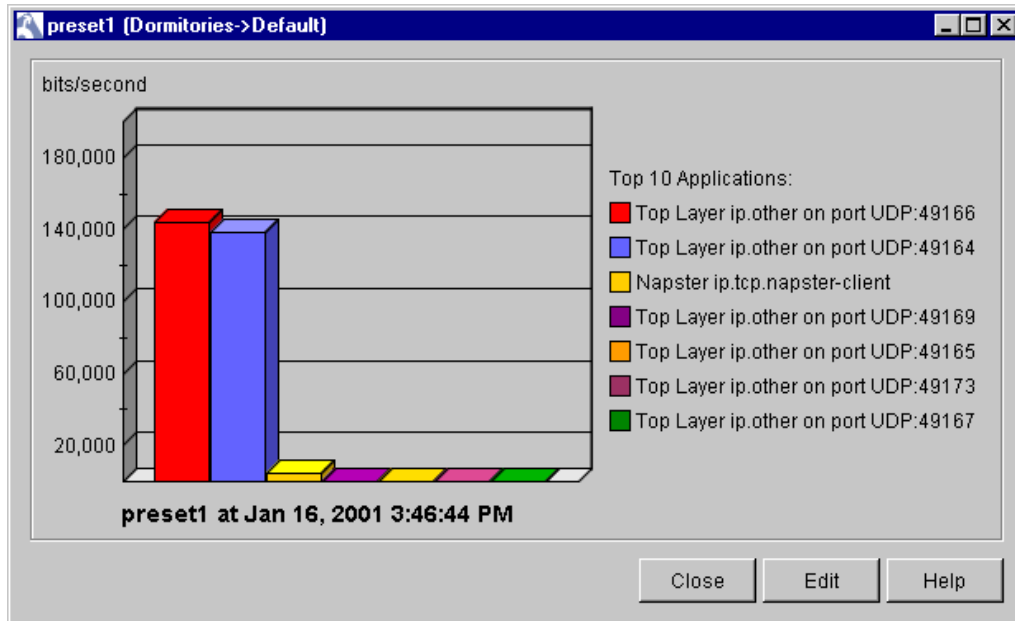


Figure 13: AppSwitch Priority shaping

The PacketShaper data was misleading at first. It presents the H.323 class as inclusive of all sub-protocols. We mistakenly assumed that this would apply to the UDP traffic. When testing the Packeteer device for priority shaping, it initially appeared to be working correctly on the video data. With video set to high priority, Napster to 0, and all other to best effort, the traffic was shaped as shown in Figure 14 below. We assumed that video data packets were being marked at the same priority as defined by the policy for the class. After close examination we determined that the UDP traffic was being serviced by a different rate control policy. This actually seemed to work fairly well but would not scale to a more diverse environment.

To mark the video packets, the PacketShaper must mark all RTP packets or define specific IP numbers to prioritize.

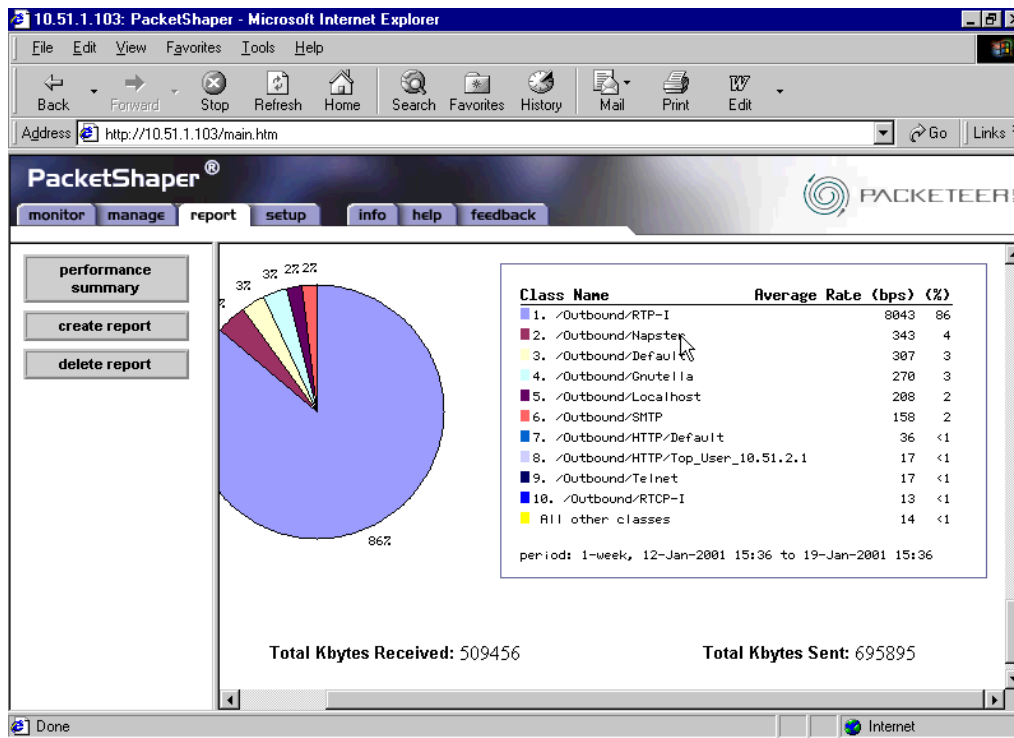


Figure 14: Packeteer shaping with RTP rate control.

Latency

All the products are, in essence, wire speed switches and thus show low uncongested latency. Latency in any bandwidth constraining device will vary widely depending on in/out bandwidth, packet size, priority, and queuing of the individual traffic stream. The PacketShaper is unique in managing TCP/IP sessions according to each session's measured latency and resetting the window size so latency is reduced for all TCP/IP traffic with little to no queuing delay. Packeteer documentation states that their latency is under 2 milliseconds (ms) in all cases. One independent test placed it around 0.5 ms. The NetScreen is also considered to be about 2 ms. in independent tests. The NetEnforcer has been clocked at 0.2 ms. across a wide range of packet sizes. The AppSwitch has been clocked as low as 0.02 ms. for simple forwarding to 0.38 ms. during heaviest stress. In comparison, a typical router hop can introduce around 50 ms. latency.

Discussion

This evaluation brought to light many issues associated with bandwidth management, especially the different methods for controlling TCP and UDP traffic. Guaranteed delivery of multimedia traffic is one of the driving forces behind this study and this type of traffic proved to be the most difficult to shape due to the complexity of the H.323 protocol. As discussed earlier H.323 uses TCP to set up the session but UDP for the actual data flow. Both H.323 TCP and UDP use dynamically allocated ports. A fairly sophisticated technique is required to identify the H.323 flow and control the delivery throughout the session.

TCP and UDP are transport protocols (OSI layer four) that provide data transmission over an IP network. TCP is connection-oriented while UDP is not. UDP can efficiently transport large amounts of data but cannot guarantee delivery.

Real Time Protocol (RTP) is the standard protocol for delivery of video and audio packets as defined in RFC 1890. Specific implementations of the RTP protocols are determined by the application. RTP uses UDP as a transport protocol and consists of 2 parts: data and control. Audio and video data are carried as RTP data packets while the control information is contained in RTCP packets. They typically use two consecutive ports with the data taking the even, lower port. The RTCP packets contain information that is useful for QoS, including an SDES (source description) packet which can contain information about the application sending the data. TopLayer's AppSwitch understands this information and uses it to shape traffic for H.323 sessions. In the video traffic we monitored, we found that the actual data packets were marked correctly with the priorities set at the beginning of the session.

The Packeteer product performed differently. The TCP packets generated by the call setup protocols were marked with high priority. The UDP packets carrying the RTP and RTCP information were not marked. Instead, a generic rate control was applied to all RTP packets. Therefore, all video traffic appeared to flow at a priority rate. In addition, smart UDP applications have built-in flow control mechanisms that will adjust to congestion on the network. The only problem with this approach is that other multimedia applications, including Real audio, use the RTP protocol. Implementation of this approach would have to combine other variables such as IP address to ensure that not all multimedia applications have high priority.

The Packeteer device, however, could truly identify Napster and Gnutella traffic. We are attributing its successful identification of Napster and Gnutella to an ability to look inside the data portion of the packet, although this theory has not been confirmed. The other products could control Napster in many normal environments by using the standard ports but would not be effective if any variations were implemented, which is commonly the case.

Company background

When deciding what QoS device to use in a strategic situation, it is appropriate to consider the longevity and strength of the manufacturer. The companies behind all these products are relatively new on a traditional time scale, but they are not newcomers when considering the age of the QoS technology itself. Packeteer and Allot are both 5 years old; Top Layer (formerly BlazeNet) is 4; and NetScreen is 3.5. Although the youngest, NetScreen is the largest with 250 employees; Packeteer and Top Layer currently employ just under 200; and Allot is the smallest at 110. Packeteer is the only publicly traded company and an investigation of its stock history reveals that its stock generally reflects the ups and downs of all the "dot.coms". The other three companies plan to go public shortly. Allot's home office is in Israel.

Conclusion

The objective of this study was to determine the ability of these 4 devices to identify, mark, and shape traffic. Three of the four can do this by port or IP number but, as mentioned earlier, so can some of the Cisco devices currently at customer sites. The PacketShaper could identify and control Napster and Gnutella traffic over the greatest range of ports. This product also did a fairly good job of controlling video traffic, but there are some issues with implementation as discussed above. The TopLayer product (AppSwitch) could not consistently identify Napster or Gnutella (in this version) but did the best job of controlling video traffic. However, the TopLayer product was also the most complicated to configure and administer. NetEnforcer and NetScreen recognize only a fraction of the applications recognized by PacketShaper or AppSwitch and NetScreen cannot mark traffic.

MOREnet's implementation of QoS is still being determined, and at this point we may need to make a philosophical choice. Do we ensure the delivery of high priority traffic and let all other traffic go best effort, or do we try to shape non-desirable traffic as well? We are not sure that the latter is realistic based on the quickly changing characteristics of non-desirable traffic as well as the ability and cost associated with the products trying to keep up with it.

Appendix A – Decision-makers guide to QoS Devices

Based on the MOREnet review of Allot's NetEnforcer, NetScreen, Packeteer's PacketShaper and Top Layer's AppSwitch.

Exclusions:

Need packet priority marking? – Exclude NetScreen.

Need application recognition beyond port number? – Exclude NetEnforcer and NetScreen.

Requirements:

Must operate as a local QoS aware switch? – Select AppSwitch.

Must do port redirection by application, port shadowing, or port sharing? – Select AppSwitch.

Must do multi-port firewalling? – Select AppSwitch.

Must do external VPN support? – Select NetScreen.

Must to TCP-based rate control? – Select PacketShaper.

Desirables:

Firewall? – NetScreen or AppSwitch.

Common QoS needed on multiple paths to gateway router? – PacketShaper or AppSwitch.

Best Napster and Gnutella discrimination and control? – PacketShaper.

Best H.323 session control? – AppSwitch.

Ease of configuration? – PacketShaper.

Ease of application level data collection and reporting? – AppSwitch.

Best online data reporting? – NetEnforcer.

Ease of identifying current bandwidth hogging users? – NetEnforcer or PacketShaper.

Ease of identifying current bandwidth hogging applications? – NetEnforcer or AppSwitch.

Ease of identifying historical bandwidth hogging users or apps? – AppSwitch.

Appendix B – Bandwidth Management Devices Comparison

Background:

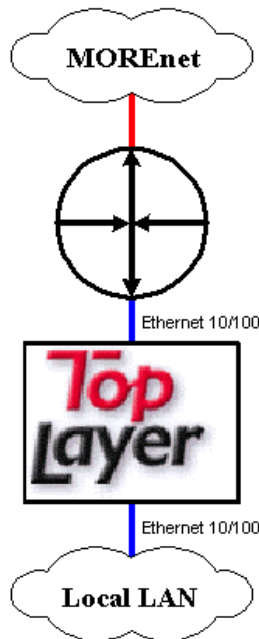
Comparison Chart:

Feature/test	Top Layer AppSwitch 2500/3500	NetScreen NetScreen 100	Packeteer PacketShaper	Allot NetEnforcer	Palasades PacketHound
Monitoring:					
Monitoring features	Good 6-sec refreshing graph	Requires a lot of setup to monitor applications, hard to compare apps. Log shows multiple apps.	Spreadsheet look to interface. Good Top User report.	Good multi-chart display of current traffic. Includes top users and top servers.	Log file based.
Application discrimination:					
Discrimination features	Huge (500+) list of predefined apps. 126 byte deep packet analysis. Stateful analysis.	44 predefined apps.	Huge (500+) list of predefined apps. Deep packet analysis. Stateful analysis	120+ protocols/apps	Few, specialized predefined apps. Deep packet analysis. Stateful analysis.
Napster	Yes, unless client is using well known port.	Only custom setup by port.	Yes, unless client is using well known port in current version.	Only by port.	Yes.
Gnutella	Only custom setup by port in current version.	Only custom setup by port.	Yes, unless client is using well known port.	Only custom setup by port.	Yes.
H.323	Only custom setup by port range. Found 6 49xxx ports.	No detection.	Labels as H.232 and RTP.	Only custom setup by port range. Found 2 49xxx ports.	No.
Chariot Apps	Yes		Yes	Yes	No.
Other Apps					Real products.
Policing:					
Policing features	Follows dynamic port settings through session.	Deny and allow by port.	TCP based rate control.		Firewall-like deny through TCP reset
Ease of setting	Simple port setting	Simple port setting	Simple port setting	Simple port setting	
4 Mbps to router	Yes	Yes	Yes	Yes	
8 Mbps to router	Yes	Yes	Yes	Yes	
Shaping:					
Shaping features	Follows dynamic port settings through session.	Min., Max. and Burst.	TCP based rate control.	Guaranteed, Max. & 8 priority settings	None
Ease of setting	Complicated		Easy	Fairly complicated	
Flexibility	Very flexible	Limited	Very Flexible	Limited flexibility	
Enhance Chariot App					
Decrease Napster	Yes but not others	No	Yes + gnutella	Only by ports	
Maintain H.323	Yes	No, only by IP	Yes	Only by ip number	

Marking:					
Marking features	TOS (including precedence), DiffServ, 802.1P based on service class.	none		TOS (including precedence), DiffServ.	none
Ease of setting	Simple	N/a	Simple	Simple	-
Mark application	Yes	No	Yes	Yes	-
ReMark application	Yes	-N/a	Yes	No	-
Delete marking		-			-
Routing:					
Protocols:	Static with filters	Static			
Authentication:					
Authentication available	LDAP	LDAP		External radius, ndap, secureid, LDAP	
Filtering:					
Filtering available				External WebSense	
Logging:					
Logging features	MS Access			Email delivery	
Capture	Redirected to workstation				
Clearing	Overwrites each time after start.				
Physical:					
Ports	14	2 to 3	2 to 6 in in/out pairs	2	1
Auto Ports	10/100	10/100	10/100	10/100	
Power	Redundant possible				
Failure mode	Bypass		Bypass		
Other:					
Initial setup	Works only with a specific Java machine.			Install java and security key.	
Administrative interface	Limited access due to Java specific version.	Good HTML layout.	Spreadsheet look to interface.	Slow Java based.	
Database access	MS Access, SQL coming				
Redirection	Can be setup for external cache, intrusion detection etc appliances, maintains virtual addresses for shared devices	HTML to WebSense?		Can be redirected to cache server.	
Security	DOS prevention features. Other firewall features. Deny applications.	NAT, DOS prevention features and others. VPN. Deny by port.	Deny specific applications.	Deny by port.	
Other	Group by ports and				

	then set policies for different zones.				
--	---	--	--	--	--

Appendix C – QoS Device General Information



Existing MOREnet connection and router

Top Layer Appswitch 3500 can connect via
10 Mbps,
100 Mbps or (after upgrading router)
1000 Mbps

AppSwitch is 2 rack units high (~2.5 inches)
and can be standalone or mounted in 19" rack.

Typical installation will use two Ethernet ports,
leaving 10 Eth-10/100 ports and one Gig-E port
available for further LAN connections or future
expansion.

One time cost ~ \$20,095, maint at \$1,845/year

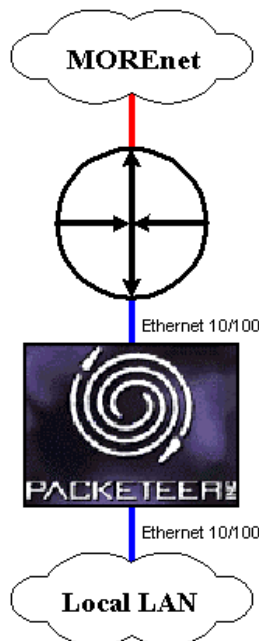
AS3502	\$18,995
TN-AHR-3500-1Y	<u>1,100*</u>
First year:	\$20,095

TN-AHR-3500-1Y	1,100*
TN-SSS-1y	<u>745*</u>
Yearly after first:	\$ 1,845

(* discounts up to %20 available
for longer term contracts, TN-AHR
is one day advanced replacement,
TN-SSS is software maintenance)

Suggested additional:

Service-Kit64	\$ 695
(backup 64MB SanDisks)	
TN-AOC-OS-NA	6,000
(2 day on site training for 12)	



Existing MOREnet connection and router

Packeteer PacketShaper 4500 can connect via
10 Mbps or
100 Mbps

PacketShaper is 3 rack units high (~3.5 inches)
and can be standalone or mounted in 19" rack.

Typical installation will use two Ethernet ports,
the PacketShaper can use up to two optional
in/out ethernet modules for parallel isolated
connections to the router.

One time cost ~ \$16,800, maint at \$2,400/year

PS4500-LD45M	\$16,000
ARS	<u>800</u>
First year:	\$16,800

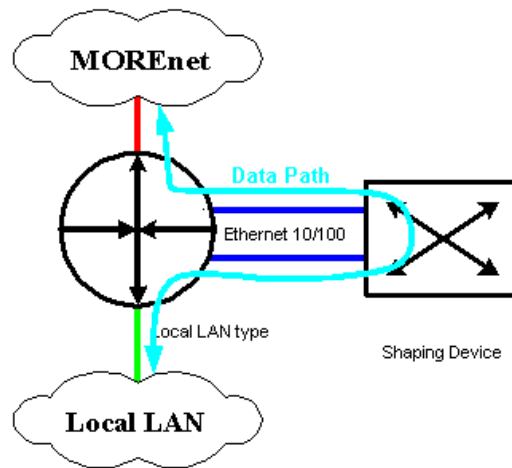
CSP	<u>2,400</u>
Yearly after first:	\$ 2,400

(CSP combines 1 day advanced
replacement (ARS) with software
maintenance)

Suggested additional:

Training (per person)	1,500
(plus travel to Cupertino, CA)	

Appendix C – QoS Device General Information: Non-Ethernet Sites



To use an Ethernet-only Bandwidth Shaping Device at a non-Ethernet LAN site an additional local router would have to be added or additional Ethernet ports added to the existing site router. Adding another router would be a substantial cost (\$6,400 list price for FE-TR, more for FE-ATM) and introduce another single point of failure. To use the original router the LAN port would be bridged or policy routed directly to a new Ethernet port. The routers second Ethernet port would be bridged or policy routed to the original WAN port.

<u>Quantity</u>	<u>Part number</u>	<u>Cost</u>	<u>Total</u>	<u>Discounted</u>
2	Cisco PA-FE-TX	\$2,500 each	\$5,000	\$2,950

Or possibly*:

<u>Quantity</u>	<u>Part number</u>	<u>Cost</u>	<u>Total</u>	<u>Discounted</u>
1	Cisco PA-2FE-TX	\$3,800	\$3,800	\$2,242

*The PA-2FE is supported on the Cisco 7200 and 7200VXR, with the network services engine (NSE-1) network processing engine (NPE)-225, NPE-300, NPE-400