

Packeteer Technical White Paper Series

UDP Traffic Management

May 2002

Packeteer, Inc.
10495 N. De Anza Blvd.
Cupertino, CA 95014
408.873.4400
info@packeteer.com
www.packeteer.com



Company and product names are trademarks or registered trademarks of their respective companies. Copyright 2002 Packeteer, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, transmitted, or translated into another language without the express written consent of Packeteer, Inc.

Table of Contents

UDP Traffic Management	3
UDP and its Challenges	3
Outbound UDP Traffic	3
Inbound UDP Traffic.....	4
UDP Control Mechanisms	4
Example of Putting the UDP Controls Together	5
The PacketShaper Advantage	5

UDP Traffic Management

The majority of software applications use TCP (Transmission Control Protocol) for data transmission because TCP establishes an end-to-end connection and uses built-in mechanisms to verify that data reaches its destination undamaged.

While TCP dominates protocol usage, UDP (User Datagram Protocol) serves a significant set of applications. In addition to its TCP bandwidth-management features, Packeteer's PacketShaper provides mechanisms to manage UDP traffic. This paper describes how PacketShaper handles UDP, giving you the ability to distribute bandwidth in accordance with your business priorities.

UDP and its Challenges

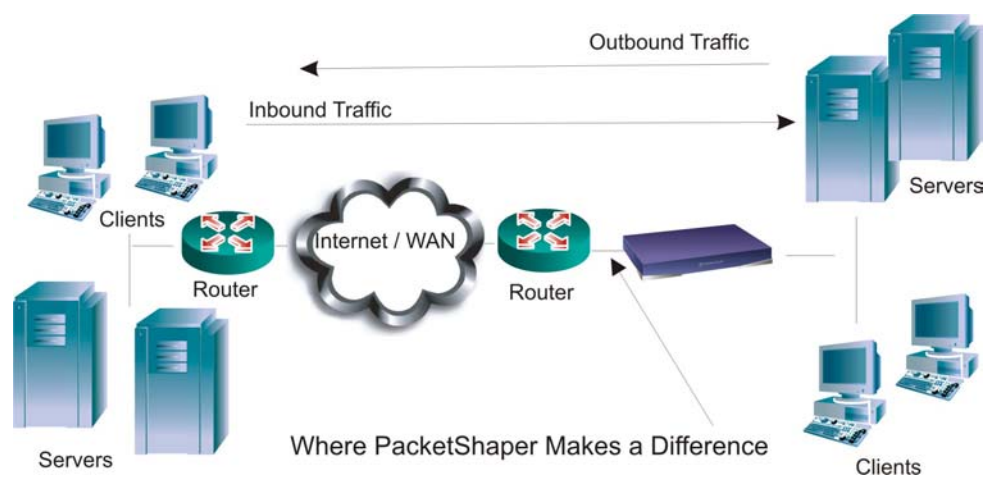
Unlike TCP, UDP sends data to a recipient without establishing a connection and does not attempt to verify that the data arrived intact. Therefore, UDP is referred to as an unreliable, connectionless protocol. The services that UDP provides are minimal — port number multiplexing and an optional checksum error-checking process — so UDP uses less time, processing, and bandwidth overhead than TCP.

While UDP doesn't offer a high level of error recovery, it still has appeal for certain types of operations. UDP is used mostly by applications that might or might not be transaction-oriented, but do require fast delivery and are not concerned with reliability — DNS, for example. Some UDP applications, such as RealAudio and VoIP, generate persistent, session-oriented traffic. Whenever an application uses UDP for transport, the application must take responsibility for managing the end-to-end connection, handling packet retransmission and other flow-control services native to TCP.

Because UDP doesn't manage the end-to-end connection, it doesn't get feedback regarding real-time conditions, and it can't prevent or adapt to congestion. Therefore, UDP can end up contributing significantly to an overabundance of traffic, impacting all protocols, UDP and TCP included. In addition, latency-sensitive flows, such as VoIP, can be so delayed as to be useless; but they still continue to be forwarded, oblivious to the fact they're contributing to the delay problem even though they are unwanted.

Outbound UDP Traffic

PacketShaper is especially effective for controlling outbound UDP traffic. When an Internet client requests data from a server, PacketShaper intervenes and paces the flow of outbound data, regulating the flow of UDP packets *before* they traverse the congested access link.



Inbound UDP Traffic

Management of inbound traffic presents a bigger challenge. By the time inbound UDP traffic reaches PacketShaper, it already has crossed the expensive, probably congested, access link and PacketShaper cannot directly control the link rate. For inbound TCP traffic, PacketShaper can influence the rate at which traffic enters the network on the other end using TCP's flow control mechanisms. UDP offers no such vehicle to be exploited. However, PacketShaper can control inbound UDP traffic's rate to the destination machine.

UDP Control Mechanisms

PacketShaper's traffic class and policy features make it easy for you to define bandwidth-allocation configurations. First, you create a UDP traffic class, listing the characteristics that identify the traffic. Many times, PacketShaper does this automatically for you by with its automatic traffic-discovery features. For example, PacketShaper spots the various types of VoIP traffic and creates the corresponding traffic classes — all automatically. After that, you apply policies or rules that define how you want that particular traffic flow to be handled. Either priority or rate policies are appropriate for UDP traffic classes, depending on the UDP traffic and its goals. A priority policy is used to control UDP traffic that is transaction-oriented. A rate policy is most appropriate for persistent UDP traffic.

PacketShaper's Control Features for UDP

Feature	Description
Priority (in both Priority Policy and Rate Policy)	Establishes a priority (0, the lowest, to 7, the highest) to be used when considering which traffic gets more bandwidth access
Rate Policy's Guaranteed bps	Delivers a minimum rate for each UDP flow Although TCP Rate Control can't influence the rate at which UDP traffic arrives at PacketShaper, a guaranteed rate can ensure that a UDP flow gets a needed rate when it does arrive. For example, you can give 24 Kbps to an important streaming application.
Rate Policy's Limit	Contains each UDP flow to a configurable maximum rate
Rate Policy's Delay Bound	PacketShaper accumulates incoming UDP packets on a flow-by-flow basis when they are not scheduled for immediate transfer (based on priority, competing traffic, and so on). The UDP <i>delay bound</i> defines how long packets can remain buffered before they become too old to be useful. For example, a delay bound of 200 ms is appropriate for a streaming audio flow.
Rate Policy's Admission Control	Determines how to handle additional requests for an oversubscribed service If the bandwidth for a traffic class is used up satisfying a guaranteed rate for many connections, you can determine how PacketShaper should treat successive connection requests — refuse a new connection or squeeze in the new connection to a fixed "trickle" rate.
Discard Policy	Tosses all packets for a traffic class, thereby blocking the service Use a discard policy for an unsanctioned application that you would prefer to not support on your network.

Static Partition	Protects or caps all the traffic in one class You specify the size of the reserved virtual link, choose if it can exceed that size, and optionally cap its growth. Partitions function like frame relay PVCs, but with the added important benefits that they cost less and they share unused bandwidth with other traffic.
Dynamic Partition	Creates per-user subpartitions dynamically, as needed, when users initiate traffic of a given class Use dynamic partitions in situations where per-user bandwidth equity is important, more important than the performance of a specific application.

Example of Putting the UDP Controls Together

VoIP voice clients typically use UDP streams. H.323, the industry standard, starts a conversation on one port (H.323), jumps to another port (Q.931), and eventually splits up into a data flow (RTP) and control flow (RTCP).

Define a partition for all VoIP traffic to protect a portion of your WAN access link for VoIP. When there is insufficient VoIP traffic to use the whole partition, the unused portion will go to others.

The setup traffic (H.323 and Q.931) is characterized by small somewhat urgent flows. PacketShaper creates the traffic classes automatically. You can assign priority policies at priority 5.

The traffic flows that handle the control responsibilities of the voice streams (RTCP) are small and intermittent. PacketShaper creates the traffic class automatically. You can assign the same policy you did for H.323.

The voice streams themselves (RTP) are large and data laden. PacketShaper creates the traffic classes automatically. A per-flow bandwidth guarantee in a rate policy can deliver smooth, jitter-free reception. Typically, if a manufacturer claims that its voice flow requires 8 Kbps, it will actually need 17 to 21 Kbps due to additional overhead and forward error correction. In addition, it is best to overstate the guarantee of UDP policies by 15 to 20 percent. Based on all this, you can assign a rate policy with 24 Kbps guaranteed, burstable at priority 7.

If you wish, you can apply a limit to the rate policy to prevent one high-capacity VoIP user from claiming an unnecessarily large portion of bandwidth.

Leaving the rate policy's default delay bound of 200 milliseconds is appropriate.

Employ Admissions Control to refuse new connections once the number of VoIP streams climbs sufficiently to use the entire VoIP partition (or allow the partition to burst to link size if desired).

The PacketShaper Advantage

UDP traffic management is part of a comprehensive strategy to manage the bandwidth and performance of many types of traffic and applications. When used in combination with the variety of other PacketShaper techniques, all applications can be the beneficiaries of consistent, predictable, appropriate performance. For more information, consult Packeteer's web site at www.packeteer.com or call 408-873-4400 or 800-697-2253.