

Use of QoS to Manage Traffic Congestion on Network Edge Links

T. Kelleher, B. Saunders, S. Ostermann
Ohio University

terry.kelleher@ohiou.edu, brandon.saunders@ohiou.edu,
ostermann@cs.ohiou.edu

Abstract

Quality of Service (QoS) implementations such as Diffserv and Intserv have focused on an end-to-end solution to providing alternative levels of service. Applications needing guaranteed bounds on things like throughput, delay, and packet loss were the target of such implementations. QoS can also be used to solve a different class of problem: that of network contention. There also exists a need for Quality of Service on edge links, where bandwidth is scarce. In a University setting, the network is serving the needs of both the mission of the university (teaching and research) as well as the residential life of its students, faculty, and staff. The traffic contending for limited network resources is caused by a mixture of work and pleasure. An effective Quality of Service system at a University should both prioritize traffic and apportion bandwidth to that traffic accordingly. Additionally, a complete Quality of Service solution will include network infrastructure as well as network policy.

This paper describes the results of applying QoS techniques to a University egress router. Two methods were used, both applied to a Cisco 7200 router. Committed Access Rate (CAR) was used first, but was then replaced by Class Based Weighted Fair Queuing (CBWFQ). This paper seeks to demonstrate how these QoS mechanisms applied to the egress router can predictably affect the packet loss ratio of certain classes of traffic.

1 Introduction

During the first part of the 2000/2001 academic year, Ohio University saw a dramatic increase in its Internet usage. Three factors contributed

to this saturation of the network: the dramatic increase in the use of peer-to-peer applications like Napster, the topology of the network, and its limited interconnect resources with its ISP. At that time, the topology of Ohio University's network consisted of a pair of redundant backbone routers that connected Ohio University to its ISP. The routers had high-speed connections from various internal routers coming into them but shared a slow link (a fractional DS3) to the ISP. Because of the asymmetry of link speed between incoming and outgoing interfaces, the backbone router became a bottleneck for all traffic leaving and entering Ohio University's internetwork. During the fall quarter, the backbone router had links of 10 Mb/s, 100Mb/s, and 1Gb/s coming in from the rest of the campus and an outgoing link speed of 24Mb/s. The traffic going from Ohio University to the rest of the Internet (the outbound traffic) kept the 24Mb/s link between Ohio University and its ISP full. As a result, queues on the backbone router were continually full and many packets were dropped due to congestion. Additionally, an analysis[1] of the yearly outbound traffic for 2000 showed the following breakdown of packets:

- ffi 81% of all outbound packets came from dorms at ports above 1024

- ffi 10% of outbound packets came from dorms from well-known ports

- ffi 4% came from non-dorms from ports above 1024

- ffi 5% came from non-dorms from well-known ports

This breakdown suggests that a vastly disproportionate amount of Ohio University network resources was being devoted to low priority traffic¹. Applications like Napster, which was a heavy producer of network bandwidth at the time, were using high port to high port connections. Ohio University faculty, staff and students who were using the Internet to do work were being squeezed out by the low priority traffic. Not only was the ISP link unable to accommodate the volume of the university's offered traffic, work and research related traffic seemed to suffer more.

¹As a qualitative policy, we define low-priority traffic as network traffic that has more to do with recreation than with educational or research pursuits

At the time, Ohio University had no resources to invest in a bigger network link to its ISP. An alternative proposal involved re-prioritizing the outbound traffic. Developing a strategy whereby traffic was prioritized seemed a logical alternative. QoS mechanisms seemed the best way to apply this strategy. As both a stopgap measure and proof-of-concept, a rate limiting algorithm was applied to the border routers affecting outbound traffic.

The application of Committed Access Rate (CAR) to the border routers had an immediate effect. This effect suggested a more formal QoS strategy should be pursued. Additionally, new distributed applications such as Kazza and Gneutella began to grow in popularity. Ohio University's network utilization rate was only going to get worse. This prompted Ohio University to begin investigating queuing functions.

Of the queuing functions supported by Cisco, Class Based Weighted Fair Queuing (CBWFQ) appeared to be the best fit for Ohio University. It allows the traffic to be split into multiple queues which are serviced in a weighted round robin format. The primary advantage is that the unused bandwidth from higher priority queues could be consumed by the lower priority queues.

2 Rate Limiting

A relatively simple QoS strategy was first pursued in solving the network bottleneck at Ohio University. Something easy to implement may offer both quick relief and quick proof that QoS might indeed offer a real solution. The vendor of the router equipment already in place at Ohio University (Cisco) dictated the set of QoS mechanisms from which to choose this first simple mechanism.

Cisco offers two classes of per-hop QoS: scheduling and queue management[2]. Scheduling allows the router to determine how packets are queued hence when the packets are transmitted. Queue management allows the router to manage queue size by buffering or dropping packets when the queue size is exceeded.

Queue management mechanisms seemed the easier of the two to implement. Two general per-hop mechanisms fall under the queue management class. Traffic shaping and traffic limiting can restrict the amount of bandwidth that certain traffic flows are allocated. Traffic

limiting will drop the packets once the rate has been exceeded whereas traffic shaping will buffer the exceeded packets and transmit them later in an effort to bring the flow to within the set rate. Both mechanisms enforce the limit whether there is congestion at the router or not. Thus traffic could not use more bandwidth than its threshold allows even if more bandwidth is available on the pipe[2].

Traffic limiting was used to restrict the amount of high-port to high-port traffic leaving OU's border router. Analysis of traffic being rate limited demonstrates how this QoS mechanism applied to the egress router can predictably affect the packet loss ratio of certain classes of traffic.

3 Class Based Weighted Fair Queuing

Although traffic limiting had an immediate effect, it did not appear to be the ideal solution since it may actually unnecessarily penalize low priority traffic when high priority traffic is light. Instead the ideal QoS solution would allow Ohio University to devote more bandwidth to high priority traffic to ensure it gets through, but not penalize low priority traffic in the case when high priority traffic flow is light or non-existent. More of the outbound high priority traffic occurs during workdays. A solution should not restrict the bandwidth low priority traffic could use but should guarantee the bandwidth the high priority traffic will get. For instance, at night, low priority traffic would be free to use all available, remaining bandwidth.

Cisco offers three QoS scheduling mechanisms: priority queuing, custom queuing, and weighted fair queuing (WFQ) and it's variant Class-Based Weighted Fair Queuing (CBWFQ). The default behavior of scheduling in routers is first-in-first-out (FIFO). Packets are placed into queues in the order in which they are received. Priority queuing, a rigid scheduler, places packets in one of four queues labeled high, medium, normal, or low. Packets in the normal or low queue will not be processed in a timely fashion. Priority queuing is most suitable for time-critical but low bandwidth traffic, which is not a common case at Ohio University. Custom queuing and WFQ place traffic in one of several queues and each queue is serviced in a round-robin fashion according to configurable weights applied to each queue. This prevents

any one queue from starving, as is the case with priority queuing. The weights, in essence, provide the bandwidth limitation. The difference between the two lies in how they are implemented in the router. Cisco contends that CBWFQ is more efficient and thus recommends it over custom queuing. Both have the necessary properties to solve Ohio University's network resource problem[2].

The Weighted Fair Queuing (WFQ) algorithm was first described in the work of Demers, Shenker and Keshav[3] and further refined under the name Generalized Processor Sharing (GPS) by Parekh and Gallager[4, 5]. WFQ is based on the notion of max-min fairness as it relates to traffic flows. Each flow receives a fair share of the bandwidth. Fairness is a very relevant concept in relation to Internet traffic because of a phenomenon known as "lock-out" or "packet train"[4, 6, 2]. "A single connection or a few flows [can] monopolize queue space, preventing other connections from getting room in the queue"[6]. Small packets/flows will either experience greater delay relative to larger flows since they will have a longer wait in the queue, or they will be dropped because the queue is full. If they are TCP packets, their resending will be further delayed by TCP's congestion control mechanism. Small packets and short flows can thus suffer at the hands of long flows[4, 6]. Min-max fairness dictates that packets/flows with the smallest resource demand are serviced first. GPS, which is more theoretical than practical, implements fairness in the following way. To be treated fairly, each packet is put into its own queue. Each queue is serviced once per round, and during each servicing, one bit from each packet is removed from the queue. Since every packet is visited exactly once per round, each receives a fair share of the resources. The bit-by-bit processing means that the smallest packet would finish first. Because queue processors in routers have no notion of bit-by-bit processing and because packets are variable length, WFQ assigns a weight to each packet that approximates the time it would take to finish processing if GPS were used, thus ensuring fairness. This weight, or timestamp, assigned to packets is "based on their arrival rate at the router, their scheduled departure time [if GPS were used] and their length"[7]. The order in which packets are put in the departure queue is based on this timestamp. Those packets with the smallest timestamp get put into the departure queue first[7, 4, 5].

Two other properties of WFQ make it an attractive QoS algorithm. The first, termed flow protection, is related to fairness. Fairness guarantees that a flow will receive its allotted bandwidth no matter what other flows are being serviced by the router. An errant or misbehaving flow cannot disrupt another flow, as is the case in FIFO[7, 4, 5]. Second, WFQ is a work-conserving algorithm[7, 4, 5]. Work conserving means the router does not sit idle if packets are in any queues, as is the case in schemes like priority queuing.

CBWFQ has all the properties of WFQ, plus the ability to guarantee bandwidth to designated traffic[2]. With CBWFQ, a router can be configured to add weights to the timestamp. These weights are based on classes of traffic that a user defines. Thus certain classes of traffic may receive more bandwidth (because of how they are placed on the queue) than they would receive under WFQ. CBWFQ can ensure that Ohio University's high priority traffic receives a certain level of bandwidth without being unfair to other traffic. It also means that if high priority traffic is light (or non-existent), then the bandwidth can be used by all other traffic.

Selection of the interfaces on which to apply CBWFQ is critical to the success of the system. CBWFQ should be placed on the congested interface with the least amount of bandwidth in the critical path between network cores. In Ohio University's case, the congestion point occurs at Ohio University's interface to the ISP's ATM network. Analyzing traffic flow supported this selection and it eliminated other interfaces as candidates for differentiated queuing.

On the ISP's ATM network, two permanent virtual circuits exist, one for Internet 1 service, and a second for Internet 2 service. Because of the static nature of this ATM network, using the ATM QoS features does not appear to be a viable solution.

Because of the ATM QoS functions, queuing functions like CBWFQ were not originally supported by Cisco on ATM interfaces. In more recent versions of the Cisco IOS, it is possible to apply differentiated queuing functions like CBWFQ on ATM sub-interfaces. This ability will allow for the desired traffic shaping functionality.

The traffic selection schema separates the traffic into three classes. The first class contains traffic from well-known peer to peer applications (distinguishable by port number), and is allotted 5% of the bandwidth.

The second class gathers port agile applications like Audio Galaxy. This presented several problems that will be discussed later. Because of these issues, this class receives 30% of the bandwidth. The remainder of the traffic is delivered as best effort with the remaining 65% of the bandwidth.

Traffic is selected for the second class if both the source and destination ports are greater than 1024. This criterion was selected by observing that the majority of desirable services operate on ports less than 1024. Unfortunately, it also selects applications like Passive FTP[8] and several custom applications. In extreme cases, an exception in the classification Access Control Lists (ACL) had to be installed. These exceptions marginally increased the CPU utilization of the router and increased the complexity of the system and its management; in general it is not an extensible solution.

During the initial design, using a separate set of QOS classes for the dormitory traffic was evaluated. The only method for selecting this traffic was by IP address. Because of the layout of the University's address space, a very long ACL would have been needed to select the student traffic. The size of the ACL's and the number of queues has a direct effect on the CPU utilization. As the complexity of the system increased, the CPU utilization quickly approached a point that the router operation was being affected.

Selecting by location is actually preferable to selection by application. Selection by application will tend to cause application rotation and masking, which will make traffic harder to characterize[9]. It is generally believed in the Internet 2 community that applications will begin using HTTPS or SSH to avoid application recognition. This would make the traffic completely indistinguishable by even state-full classification machines

4 Current and Future QoS at Ohio University

Ohio University's queuing strategy is similar to the Q-Bone Scavenger Service implemented on Internet 2. To participate in this system, Ohio University needs to begin marking traffic with the QBSS code point[10].

An integral part of the Ohio University network engineering philosophy is the development of a no single point of failure system. This philosophy also extends to the Internet service. This means not only fault tolerant connections to our primary ISP, but also connections to a secondary ISP. Because of the cost, the secondary ISP supplies a fraction of the bandwidth that the primary ISP offers. This bandwidth represents the amount necessary to operate business critical services. During normal operations, congestion on this connection is alleviated with routing changes, which favor the primary ISP. Currently the queuing structure on this connection mirrors that of the primary ISP. A more advanced queuing structure will be developed in the next revision of the system.

References

- [1] "Ohio University Internet Usage Yearly Summary 2000," April 16 2001, toddstest3.cats.ohiou.edu/osterman/internet/mrtg/work/132.235.195.29.12.html.
- [2] Cisco Systems, "Planning for Quality of Service," April 9 2001, www.cisco.com/univercd/cc/td/doc/prdution/rtrmgmt/qos/qpm1.1-/using_qo/c1plan.htm.
- [3] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," *SIGCOMM Symposium on Communications Architectures and Protocols*, 1989.
- [4] S. Bhatti and J. Crowcroft, "QoS-Sensitive Flows: Issues in IP Packet Handling," *IEEE Internet Computer*, pp. 48–57, July-August 2000.
- [5] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in the integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.
- [6] B. Braden et al., "Recommendation on Queue Management and Congestion Avoidance in the Internet," April 1998, RFC2309.

- [7] C. Metz, "IP QoS: Traveling in First Class on the Internet," *IEEE Internet Computing*, pp. 81–88, March–April 1999.
- [8] J. Postel and J. Reynolds, "File Transfer Protocol (FTP)," October 1985, RFC 959.
- [9] "Qbss deployment recommendation," 9/1/02, <http://www.internet2.edu/qos/wg/wg-documents/qbss-deployment-recommendation.txt>.
- [10] "Qbone scavenger service (qbss) definition," 9/1/02, <http://qbone.internet2.edu/qbss/qbss-definition.txt>.

is also currently active in using network packet traces for security analysis and intrusion detection system.

5 Author Bibliographies

Terry Kelleher is a Senior Network Engineer for Ohio University's Communication Network Services and a graduate student in the school of Electrical Engineering and Computer Science. Her research interests include IPv6, network monitoring and Quality of Service.

BrandonSaunders is a Senior Network Engineer for Ohio University's Communication Network Services and a Graduate student in the school of Electrical Engineering and Computer Science. He received a Bachelors Degree in Electrical Engineering with a Specialization in Computer Engineering from Ohio University in 1999. His research interests include network quality of service and fault tolerant network topologies.

Dr. Ostermann is an Associate Professor and the Assistant Department Chair in the school of Electrical Engineering and Computer Science at Ohio University. He is an expert in network protocol research and is also actively involved in teaching network basics and details at the graduate level. His experience studying the TCP protocol in interesting environments lead to the development of tcptrace, a public domain Unix tool for investigating the macroscopic and microscopic nature of network traffic. Dr. Ostermann and his research group have spent several years studying TCP in unusual environments including LEO and GEO satellites, high- error environments, and high-delay environments. Dr. Ostermann has published many conference papers, journal articles and RFCs on network characteristics and the analysis of network traffic and is a regular participant at SIGCOMM. In addition, he