

Case Study ■

Exploring Issues of Quality of Service in a Next Generation Internet Testbed: A Case Study Using PathMaster

MARK A. SHIFMAN MD, PhD, FREDERICK G. SAYWARD, PhD,
MARK E. MATTIE, MD, PhD, PERRY L. MILLER, MD, PhD

Abstract This case study describes a project that explores issues of quality of service (QoS) relevant to the next-generation Internet (NGI), using the PathMaster application in a testbed environment. PathMaster is a prototype computer system that analyzes digitized cell images from cytology specimens and compares those images against an image database, returning a ranked set of “similar” cell images from the database. To perform NGI testbed evaluations, we used a cluster of nine parallel computation workstations configured as three subclusters using Cisco routers. This architecture provides a local “simulated Internet” in which we explored the following QoS strategies: (1) first-in-first-out queuing, (2) priority queuing, (3) weighted fair queuing, (4) weighted random early detection, and (5) traffic shaping. The study describes the results of using these strategies with a distributed version of the PathMaster system in the presence of different amounts of competing network traffic and discusses certain of the issues that arise. The goal of the study is to help introduce NGI QoS issues to the Medical Informatics community and to use the PathMaster NGI testbed to illustrate concretely certain of the QoS issues that arise.

■ *J Am Med Inform Assoc.* 2002;9:491–499. DOI 10.1197/jamia.M1030.

This case study describes a project that is exploring issues of quality of service (QoS) relevant to the Next Generation Internet (NGI), using the PathMaster application in a testbed environment. PathMaster is a computer system that analyzes digitized cell images from cytology specimens and compares those images against a database, returning a ranked set of “similar” cell images, together with associated diagnoses.

Practical issues involving network QoS relevant to the NGI have not yet affected many members of the Medical Informatics community to a major degree,

and many may not be familiar with the issues involved. This case study has two major goals. First, it is designed to help introduce issues of network QoS to the Medical Informatics community. Second, it uses the PathMaster application to illustrate the issues concretely. The study illustrates issues that our field needs to confront as new generations of the Internet offer increasingly robust capabilities, and as Medical Informatics practitioners start to use the NGI for advanced applications that require these capabilities.

The rapidly increasing growth of network-based computer applications has generated a variety of challenges for the field of informatics. The demand for network bandwidth (i.e., the amount of data that can be conveyed in a fixed period of time) has increased tremendously, together with the demand for fast and reliable performance of applications. Critical applications must be guaranteed the network resources they require, while other applications need adequate network resources for consistent, pre-

Affiliations of the authors: Yale University, New Haven, Connecticut.

Correspondence and reprints: Mark A. Shifman, MD, PhD, Center for Medical Informatics, Yale University School of Medicine, PO Box 208009, New Haven, CT 06520-8009; e-mail: <mark.shifman@yale.edu>.

Received for publication: 10/2/01; accepted for publication: 4/12/02.

dictable behavior. The study of network QoS has evolved out of the need to address these issues.

QoS involves a collection of technologies that facilitate the “differentiation” of network traffic, i.e., providing different types of enhanced service to selected applications. A variety of strategies are available to provide different QoS capabilities:

1. Many applications may perform adequately within the current “best effort” Internet protocol, which provides service on a “first-come-first-served” basis.
2. Certain applications require guaranteed bandwidth, for example, for real-time telesurgery.
3. Other applications may benefit from differentiated service where the application is given some form of preference over other traffic without a specific, fixed guarantee of network resources.

QoS functionality may occur both locally within a single network device (e.g., programmed into a network router) as well as globally (e.g., programmed into distributed network devices in a coordinated fashion). For example, a local device may handle traffic with a variety of queuing and congestion management protocols. Guaranteeing resources and globally differentiating between types of applications, however, requires the coordinated action of multiple devices throughout a network.

As part of the National Library of Medicine’s (NLM) NGI initiative, we are using the PathMaster application to explore issues of QoS. PathMaster is a prototype image database application that takes as input cell images from cytology specimens and compares these to images in an image database.¹ PathMaster has been expanded into a distributed parallel network-based application that can process and compare multiple cell images simultaneously. As a potential clinical application, the issues of reliable performance and predictable behavior are clearly important.

To explore QoS issues within the distributed version of PathMaster, we implemented a testbed environment using a cluster of parallel workstations connected by QoS-enabled Cisco routers. This architecture provides a local “simulated Internet” for evaluating different QoS strategies. As described below, in these testbed evaluations we used several QoS strategies: (1) first-in-first-out queuing, (2) priority queuing, (3) weighted fair queuing, (4) weighted random early detection, and (5) traffic shaping. The study describes the results of using these strategies in the

presence of different amounts of competing network traffic and discusses certain of the issues that arise.

It is important to point out that our testbed is deliberately limited in its scope. The testbed cluster provides an artificial, controlled environment. This limitation reflects the fact that the project is part of phase 2 of the NLM’s NGI initiative, which specifically focuses on local testbeds. More broadly based testing in a real-world environment would also be interesting. Such an endeavor would be a bigger and different project than the one we have performed and would require that QoS capabilities be installed and operational on the entire network. We were able to test QoS capabilities in our testbed because we had complete control of all components of the network. We believe that the project is of interest as a case study since it serves to introduce QoS concepts to the Medical Informatics community and provides a concrete medical example that illustrates the issues involved.

Background

The national NGI initiative focuses on the development and coordination of a national computer network infrastructure with high-speed data capability that is flexible, extensible, and robust in the capabilities that it provides. One mandate is to develop high-performance testbed environments that demonstrate advanced network technologies and advanced applications. The NLM’s NGI initiative currently supports a number of biomedical applications as testbeds to improve understanding of what NGI capabilities are required to serve the needs of biomedicine. The NLM’s goal is to support innovative medical projects that demonstrate and use NGI capabilities such as (1) QoS, (2) network management, (3) medical data privacy and security, (4) nomadic computing, and (5) infrastructure technology for scientific collaboration.² This study focuses on the two closely related issues of QoS and network management.

Several network testbeds have been described recently, with a variety of different scopes and test objectives. Testbeds addressing QoS and implementation issues between international sites have been described³ as well as technical aspects of international networks for computational physics with computational fluid dynamics and molecular dynamics as test applications.⁴ Internet2 and DOE testbeds explored the issues of developing and deploying the differentiated services model over wide area networks.^{5,6} Testbeds focused on more specific applications such as medical videoconferencing,⁷ medical

imaging,⁸ video simulations for anatomy and surgery education,⁹ and satellite-based IP for multimedia applications¹⁰ have also been presented.

QoS focuses on providing required levels of enhanced service to selected network applications.^{11,12} One strategy for providing dependable service involves congestion management. Several queuing algorithms are available at the level of the network router, which can deal with an overflow of arriving traffic. These algorithms utilize several techniques for prioritizing outgoing traffic.¹³ Another local strategy involves congestion avoidance, which tries to monitor and avoid congestion before it occurs.¹¹

Techniques for guaranteeing service across multiple devices in a network are also available. RSVP is an Internet Standard protocol for dynamically reserving bandwidth across the Internet.¹⁴ Another approach involves assigning different applications or users to differentiated classes of service, where each class has different performance levels.¹⁵ In this model an application is not guaranteed bandwidth, but higher classes of service are given preferential treatment relative to the lower classes.

The PathMaster NGI Testbed

This section describes the four main components of our NGI testbed system: (1) the PathMaster system, (2) the distributed, parallel implementation of PathMaster, (3)

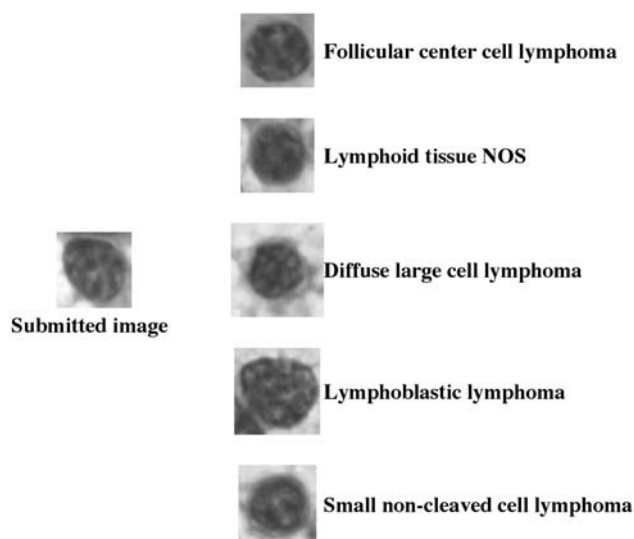


Figure 1 To illustrate PathMaster's operation, this figure shows an example of a cell image that might be passed to PathMaster for analysis and examples of several images that PathMaster might return from its image database.

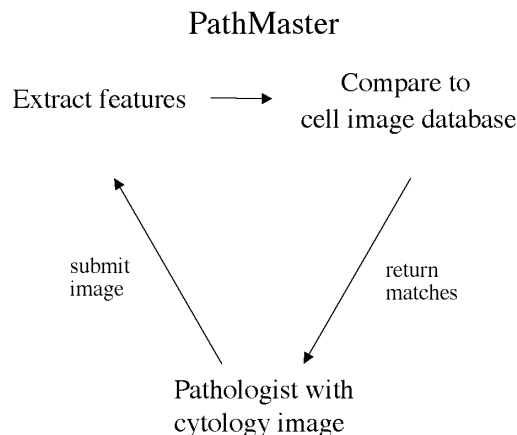


Figure 2 Schematic diagram illustrating the nonparallel version of PathMaster in operation.

the parallel NGI testbed hardware, and (4) the “clinical scenario” used in our testbed evaluations.

PathMaster

As illustrated in Figures 1 and 2, PathMaster compares submitted cell images against a database of cell images and returns a ranked set of “similar” images together with their diagnoses. PathMaster is a program written in C that extracts a variety of mathematical features, including geometric, statistical, and nuclear texture features, from a set of cell images from a cytology case. For each of these “test” images, PathMaster analyzes the image itself, together with the background of that image (a blank slide with the same lighting) and a segmentation image that masks the regions of the image (the background, cytoplasm, nucleus and nucleolus). A subset of the features is then used to compare the test image to the images in PathMaster's database. For the purposes of the NGI testbed, the five most closely matching database images are returned for each test image. The PathMaster system itself was described in a previous paper¹ and is undergoing continuing development and refinement.

Distributed Parallel-Computing Implementation of PathMaster

A version of PathMaster has been extended to be a distributed parallel-computing application (Figure 3). In this implementation, a PathMaster server farms the submitted images out to a pool of networked workstations (“workers”) for feature extraction and database comparison. This allows different images to be analyzed in parallel (simultaneously), thereby speed-

Parallel PathMaster

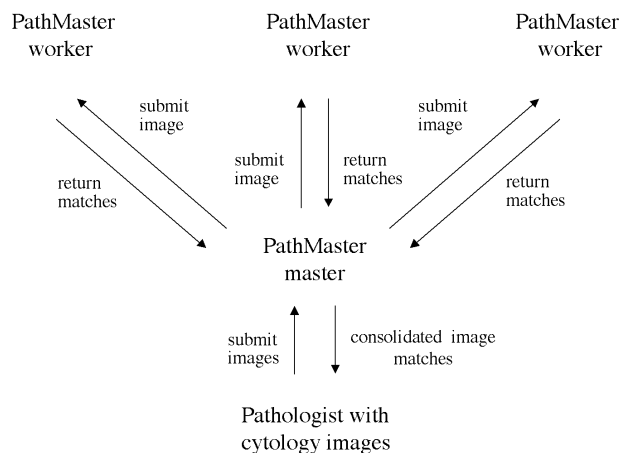


Figure 3 Schematic diagram illustrating the parallel version of PathMaster in operation.

ing up the overall PathMaster computation. Each worker is passed a subset of the submitted images and returns the list of top matches to the server. The server then passes the collated results received from the workers to the client. The parallel image processing portion of the server is written using Network Linda, a parallel programming environment.

Our NGI Testbed: A Simulated Internet on a Parallel Cluster

We run the parallel version of PathMaster on our NGI testbed, which consists of a cluster of nine Pentium III workstations linked via Cisco 1600 series routers to provide a "simulated Internet." Figure 4 provides a schematic outline of the cluster's architecture. Figure 5 shows a photograph of the physical cluster itself. Each workstation runs under Red Hat Linux 6.2 and uses Network Linda for parallel computing. The workstations are connected by a local ethernet at 10 Mbps.

As shown in Figure 4, the cluster of nine workstations is divided into three subclusters. Each subcluster contains three workstations linked via a hub to a router. The three routers are also linked via a hub. The network is configured so that each subcluster of three workstations, together with its router, forms an independent subnet. On each subnet, one "server" node runs the PathMaster server, one "worker" node provides extra computing power for that cluster's instance of PathMaster, and one "utility" node is reserved for functions such as generating competing

network traffic and measuring bandwidth. A database of image features is available locally on each server node and on each worker node. Competing network traffic is generated by downloading large files across the network between the utility nodes of the three subclusters.

Testbed Clinical Scenario

To perform our evaluation of the distributed PathMaster testbed described above, we developed the following clinical scenario (Figure 6). Each subnet can be thought of as a geographically separate local area network containing an independent cytology image database (for example, three databases might be in Connecticut, New York, and California). A pathologist with a set of images from an unknown cytology case submits those images to a server. This server then submits the unknown images to each of the three databases for analysis in parallel and retrieves images of the closest matches from each of the databases as they become available. (In our testbed, the same PathMaster program and image database are running in each subcluster. We envision a future scenario, however, in which several institutions have their own image databases and their own image comparison algorithms.)

The performance of the overall distributed PathMaster system depends on a variety of factors,

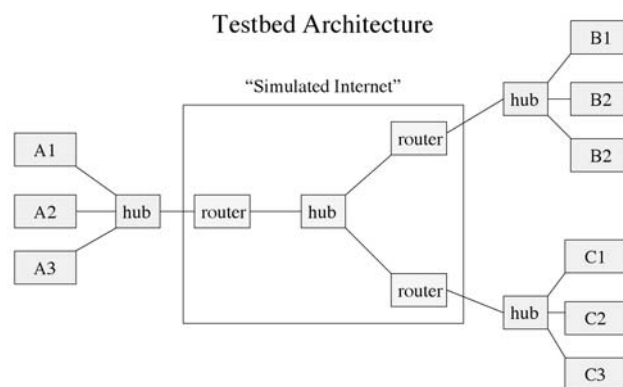


Figure 4 Our NGI testbed cluster consists of nine Pentium III workstations. They are grouped into three subclusters, each containing three workstations linked to a Cisco 1600 series router via a hub, thus forming three subnets. The three routers are in turn linked via a hub to form a simulated Internet for the purposes of our NGI testbed. QoS strategies are implemented in the routers. (An independent Windows 2000 workstation, not shown in this figure, is linked to the simulated Internet and runs the Cisco QPM software that configures the various QoS strategies on the routers.)

including the size of each image, the processing time to analyze an image, the time to compare the results against the database, and the amount of competing network traffic. For the testbed evaluations, we varied the amount of competing network traffic. Since downloading of large images can result in the consumption of large amounts of bandwidth, we simulated competing network traffic by download-

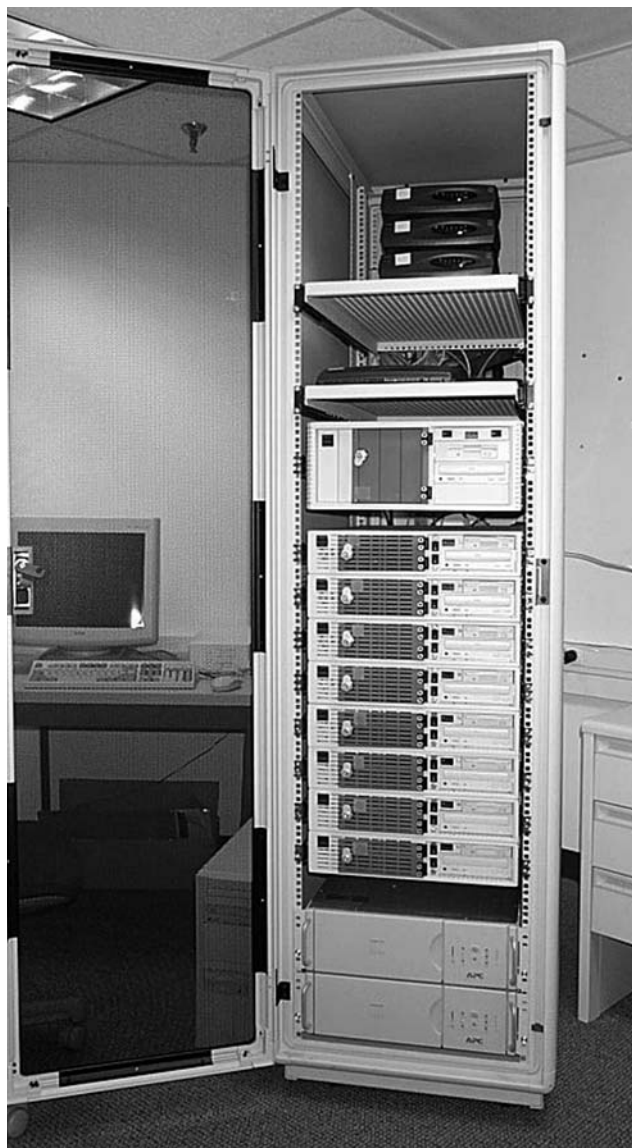


Figure 5 NGI testbed cluster hardware, described schematically in Figure 4. All of the testbed hardware is mounted in a single vertical rack. Looking at the cluster from top to bottom, the top shelf contains the four hubs. The next shelf contains the three routers. The master node is next, followed by the other eight nodes. The master node is larger since it contains a second Ethernet card for connection to the Internet. The bottom two devices are uninterruptible power supplies.

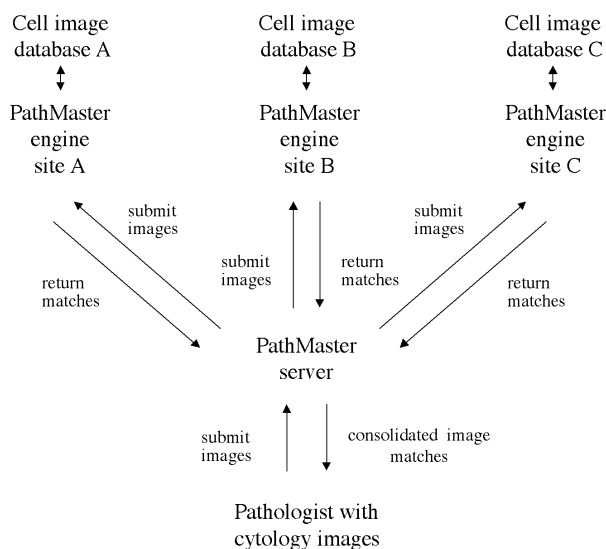


Figure 6 Schematic diagram of the operation of the "clinical scenario" described in the text.

ing large files across the network and measured the amount of bandwidth this consumed.

Internet Routing and QoS Strategies

The Internet is a collection of interconnected networks that allow computers to communicate with each other using the TCP/IP protocol. This protocol breaks up data coming out of each computer into packets that are sent over the network. These packets are labeled with the originating address and destination address and are reassembled at the destination computer. A router is a device that connects two or more networks and forwards packets to the next machine en route to their ultimate destination using a routing table stored in the router. During periods of heavy traffic, incoming packets are stored in queues prior to forwarding. A variety of QoS protocols are available to help the router deal most efficiently with these queues.

The Cisco IOS software running on the series 1600 routers in our testbed includes several built-in strategies for managing congestion and shaping traffic:

1. *First-In-First-Out (FIFO) queuing* is the default router configuration that stores and forwards packets in the order of their arrival at the router. All packets are treated equally, which can lead to significant, unpredictable delays during periods of high traffic.

2. *Priority queuing* gives priority to specific traffic based on features such as source address, destination address, or port numbers. The packets are assigned to one of four queues based on their priority. The higher priority queues are processed before the lower priority queues are processed. A problem with priority queuing of this sort is that higher priority traffic may effectively consume all of the bandwidth, thereby completely shutting down the lower priority traffic.

3. *Weighted fair queuing* remedies some of the problems of FIFO and priority queuing by sharing more fairly the available bandwidth based on traffic priority and the relative sizes of the various traffic flows.

4. *Weighted random early detection* is a strategy that attempts to reduce network congestion by taking advantage of a feature built into the TCP network protocol. Lower priority traffic is selectively discarded when the router starts becoming congested. When this occurs, the network TCP protocol is designed to automatically decrease its transmission rate (and to resend the discarded data), thus adaptively reducing congestion.

5. *Traffic shaping* limits the bandwidth available for selected applications and can be used in conjunction with other strategies. This approach can help protect bandwidth from being completely dominated by potential bandwidth hogs. For example, one can limit the high-volume network traffic to a fixed amount of the available bandwidth entering a local area network. This in turn reserves a fixed amount of bandwidth for other network traffic.

The development of sophisticated QoS strategies is an active research field. The routers that we use allow us to experiment with certain QoS strategies in a well-controlled environment. More expensive, high-end routers have additional QoS capabilities. In addition, more sophisticated QoS strategies are being actively developed by network research and development groups.

In our testbed cluster, we use QPM-PRO, a Cisco software product that facilitates the creation and distribution of QoS policies and configurations on a network. It allows one to define QoS rules for traffic flows based on features such as the source or destination, IP or port number, and priority. Separate rules for both incoming and outgoing traffic can be defined. QPM-PRO also simplifies the process of configuring (programming) the routers for each of the strategies defined above.

Methods

For our testbed evaluation runs, we implemented the clinical scenario described above as follows:

1. First, an initial set of six cell images was passed to the master PathMaster server. We chose six cell images because this is a typical number of cell images that might be submitted to PathMaster by a clinical user for a real case.

2. Copies of the six images were then sent (via the simulated Internet) to the PathMaster servers on each of the subclusters. Each cell image was approximately 100 KB in size and, as described above, was accompanied by a background intensity image and a segmentation mask image of similar size. Thus the transmission of the six cell images involved transmission of approximately 1.8 MB to each of the subclusters.

3. The six cell images were then analyzed independently on each of the three clusters. This analysis was performed in a parallel computation by the server and the worker node of each subcluster. When PathMaster was first developed, the analysis of a single cell image typically took 45–90 minutes, which was why we initially developed a parallel computing approach. This time was in large part a reflection of the use of interpreted MathLab and Visual Basic. The analysis program was successively optimized several times and is currently written entirely in C, with the result that the analysis of an image now takes roughly 5 seconds. As a result of these dramatic performance improvements, parallel computation is no longer critical to PathMaster's performance as it was initially, and our evaluation has evolved to focus on the distributed clinical scenario described above.

4. After the six images were analyzed, the resulting mathematical features were compared with the image database that is stored on each server and each worker node. This comparison takes well under a second.

5. For each of the six test images, the five most closely matching cell images from the database were then sent back to the master PathMaster server. (In this transmission, there is no need to include background and segmentation maps images.) In actual clinical use, all or some of these images would then be returned to the clinical user along with the associated diagnoses. For the purpose of these testbed runs, we focused solely on testing PathMaster within our testbed cluster.

In these testbed runs, PathMaster servers were run on nodes A1, B1, and C1. Nodes A2, B2, and C2 were worker nodes in each subnet (Figure 4). The total time for completing the entire distributed computation described above was recorded. This test was performed ten times for each of the strategy-traffic combinations.

Competing network traffic was generated from processes running on nodes B3 and C3 that continuously downloaded 20 MB files, from the master node. The downloading was performed using Wget.¹⁶ This process is identical to transferring files using FTP. Available bandwidth was degraded by increasing the number of download processes. Bandwidth between nodes was measured using Iperf.¹⁷

Testbed runs were performed with FIFO, priority queuing, weighted fair queuing, and weighted random early detection. Different priorities were assigned to PathMaster file downloads and the competing network traffic. Traffic shaping was evaluated with weighted fair queuing and several limiting bandwidths for the competing network traffic.

Results

Figure 7 shows the average times for ten testbed runs using FIFO, priority queuing, weighted fair queuing, and weighted random early detection. Figure 8 shows the standard deviations in the runtime for the same testbed runs shown in Figure 7. At low traffic

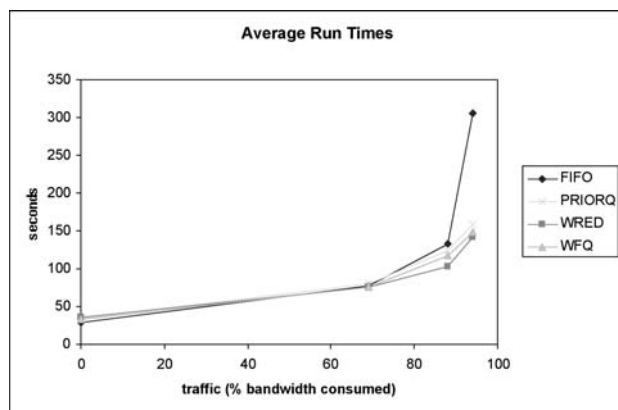


Figure 7 Results for testbed runs with various strategies. Each point on the graph is the average of ten testbed runs with the given conditions. The four QoS strategies used were FIFO, priority queuing (PRIORQ), weighted fair queuing (WFQ), and weighted random early detection (WRED).

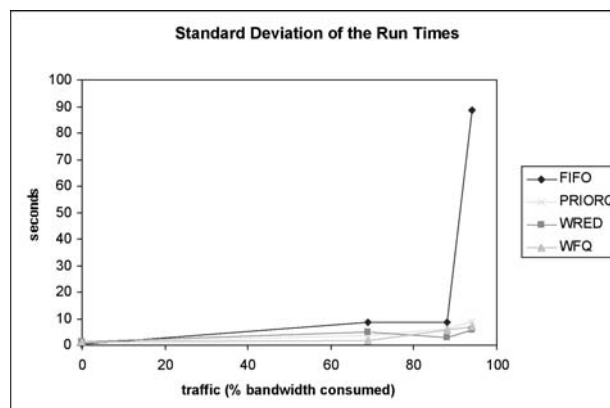


Figure 8 The standard deviations for the same testbed runs seen in Figure 7.

levels, all the strategies performed similarly. As the traffic increased, the times began to diverge. The poorer performance of FIFO can be appreciated as the traffic significantly increases. This occurs in concert with marked increase in the standard deviation for the times of the 10 runs (see Figure 8). Priority queuing performed somewhat better than FIFO at higher traffic and had significantly less standard deviation. Weighted fair queuing performed the best of the queuing strategies on average and weighted random early detection yielded the shortest times during the highest traffic.

The results with traffic shaping under weighted fair queuing are seen in Figure 9. Each curve represents the average time for 10 test runs, given a specific limitation on the bandwidth available for competing network traffic. As the bandwidth available for the competing traffic decreases from 4000 Kbps to 200 Kbps, the effect of increasing that competing traffic is markedly reduced. Table 1 shows the actual numbers that are presented graphically in Figures 7, 8, and 9.

Discussion

The spectrum of potential QoS capabilities in contemporary network-based computer applications is broad, ranging from (1) guaranteeing high performance to (2) providing reliable service with predictable degradation during times of peak congestion, and (3) classic "best effort" Internet. Critical applications, such as real-time telesurgery, demand absolutely reliable performance with bandwidth explicitly dedicated for the application. Some applications may function acceptably while sharing available band-

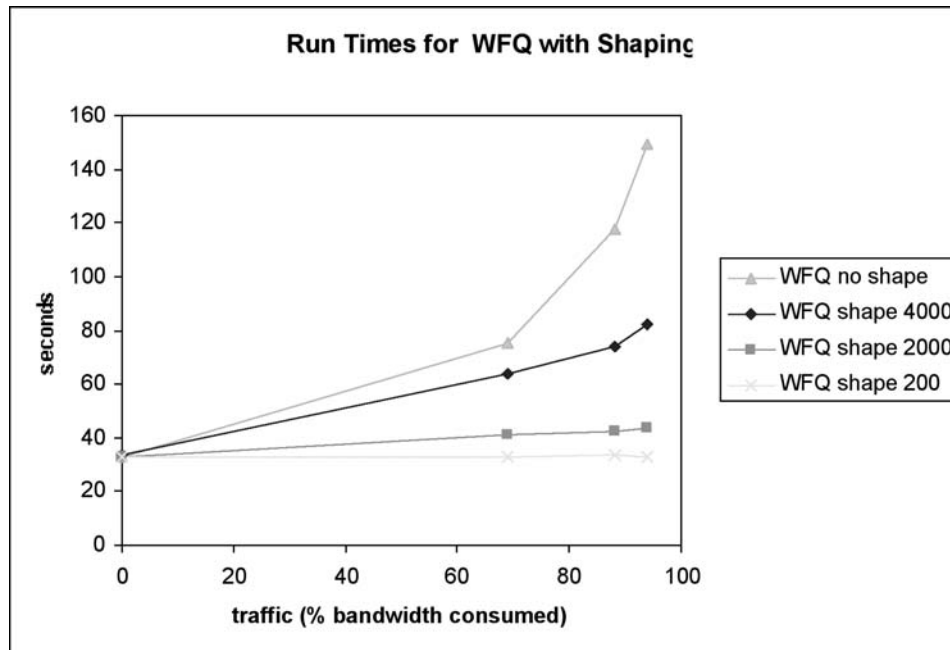


Figure 9 The results for testbed runs with weighted fair queuing (WFQ) and different parameters used for traffic shaping. The traffic shaping involved successively decreasing the bandwidth limit (Kbps) allowed for the competing network traffic. Each point on the graph is the average of ten testbed runs.

width as long as they have some degree of priority above background traffic. Suboptimal performance may be acceptable for other applications as long as the performance falls within a predictable range. For example, a pathologist may be willing to wait an average of 10 minutes for PathMaster's results at times of high competing bandwidth use but presumably would much prefer that the standard deviation of that time be small. In other words, he would much rather know that the results will be predictably ready within 9–11 minutes rather than knowing that the results might be ready in a few seconds or unpredictably require several hours or more.

PathMaster is an application that could potentially be used as a diagnostic tool or as a reference/teaching aid. As a diagnostic tool, rapid performance would be important in certain circumstances; for example, if it is used during a diagnostic procedure, such as a needle aspirate of the thyroid, in which the results of PathMaster's analysis were desired during the procedure. In addition, for any clinical use, a predictable response time (i.e., a low standard deviation in response times) would be important, as discussed above. It would also be essential that communication not be completely shut down by transient high bandwidth applications. The requirements for PathMaster as a reference/teaching aid might be reduced relative to those above. In this case a predictable response within fairly narrow limits would probably be acceptable.

Our results with the QoS strategies yield interesting results. FIFO clearly performs poorly as traffic increases. The average time is clearly longer than with other strategies, especially at high levels of traffic. The standard deviation of these times is also consistently above that of the other strategies. Thus not only does the application take longer but it also has a

Table 1 ■

Numeric Results Presented Graphically in Figures 7–9*

% Bandwidth	0	69	88	94
Average run times (sec)				
FIFO	28.1	77.3	132.4	305.4
PRIORQ	34.4	80.7	125.0	158.4
WRED	35.6	76.2	103.5	141.0
WFQ	32.9	75.1	117.3	149.3
Standard deviation of the run times (sec)				
FIFO	0.6	8.6	8.8	88.5
PRIORQ	1.0	3.8	6.2	8.8
WRED	1.4	5.1	2.7	5.7
WFQ	1.2	1.8	5.5	7.0
Run times for WFQ with shaping (sec)				
WFQ no shape	32.9	75.1	117.3	149.3
WFQ shape 4000	33.4	63.9	73.9	82.4
WFQ shape 2000	32.7	40.8	42.6	43.5
WFQ shape 200	32.9	33.1	33.2	33.1

*Each reported time represents the average of 10 runs. For each set of runs, the table indicates the % of the bandwidth consumed by competing network traffic as described in the text.

much larger range of running times. During periods of excessive traffic, PathMaster is effectively starved for bandwidth using the FIFO strategy.

Priority queuing, weighted fair queuing, and weighted random early detection perform better than FIFO. The average time is modestly shorter for weighted fair queuing and weighted random early detection. The standard deviations of all three strategies are comparable over the range of competing traffic used in this study.

Traffic shaping with weighted fair queuing also produced interesting results. As the amount of the bandwidth available to competing network traffic decreased, the average times for completion became constant with respect to that traffic.

With the above results in mind, a reasonable approach for an application like PathMaster would be to use a strategy such as weighted fair queuing or weighted random early detection with traffic shaping for specified high volume traffic flows. Restricting traffic flows not only increases the complexity of the system but also may markedly reduce the performance of competing applications that require the high volume data. Thoughtful selection of bandwidth limits will be required in such situations.

Modeling and generating competing network traffic are areas of active research with a variety of models advocated for different types of traffic flows (see reference 18 for a discussion of one model). We choose a simple file transport/bulk model for competing network traffic because it is simple to understand and because it models a real situation that can cause problems in medical center networks (e.g., the transmission of large radiology images). Bandwidth could be easily and predictably consumed by increasing the number of file downloads thereby facilitating comparison of various QoS strategies.

QoS research and development groups are currently developing "intelligent" QoS strategies that dynamically adjust certain of these types of parameters based on the nature of the actual competing demands at a given time. This work holds promise for applications such as PathMaster, which do not require dedicated resources but which nevertheless would bene-

fit from a fair and intelligent allocation of the network resources that are available.

This work was supported in part by NIH contract N01 LM93540 from the National Library of Medicine.

References ■

1. Mattie ME, Staib L, Stratmann E, et al. PathMaster: Content-based cell image retrieval using automated feature extraction. *J Am Med Informatics Assoc.* 2000;7:404-15.
2. NLM Home Page for Next Generation Internet (NGI). Available at <<http://www.nlm.nih.gov/research/ngiinit.html>>.
3. Nakagawa S, Machizawa A, Aoki M, et al. CRL next generation internet project and apii-testbed project. *J Commun Res Lab.* 2001;48:35-45.
4. Resch MM, Rantzau D, Stoy R. Metacomputing experience in a transatlantic wide area application test-bed. *Future Generation Computer Systems.* 1999;15:807-16.
5. Teitelbaum B, Hares S, Dunn L, et al. Internet2 QBone: Building a testbed for differentiated services.. *IEEE Network Mag.* 1999;Sept/Oct:8-16.
6. Burrestia J, Leighton J. DOE NGI Testbed: Project close out report. Available at <http://www.es.net/hypertext/welcome/pr/DOE_NGI_TESTBED_REPORT3_Dec01.pdf>.
7. Crudele M, Cinque G, Neglia G, et al. IP videoconferencing using a quality of service public network. *Medinfo.* 2001;10 (Pt 1):8-12.
8. Fan Y, Hwang K, Gill M, Huang H. Some connectivity and security issues of NGI in medical imaging applications. *J High Speed Networks.* 2000;9:3-13.
9. Dev P, Heinrichs WL, Srivastava S, et al. Simulated learning environments in anatomy and surgery delivered via the next generation internet. *Medinfo* 2001;10(Pt 2):1014-8.
10. McWherter DT, Sevy J, Regli WC. Building an IP network quality-of-service testbed. *IEEE Internet Computing.* 2000; July/Aug:65-73.
11. Weiss W. QoS with differentiated services. *Bell Labs Tech J.* 1998;Oct/Dec:48-62.
12. Xiao X, Ni L. Internet QoS: A big picture. *IEEE Network Magazine* 1999;Mar/Apr:8-18.
13. Cisco Documentation, Quality of Service (QoS) Networking. Available at <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm>.
14. Braden R, Zhang L, Berson S, et al. Resource ReSerVation Protocol (RSVP), RFC 2205. Available at <<http://www.ietf.org/rfc/rfc2205.txt>>.
15. Internet2 QoS Working Group Charter. Available at <<http://www.internet2.edu/qos/wg/charter.shtml>>.
16. Wget—A Utility to Retrieve Files From the World Wide Web. Available at <<http://www.gnu.org/software/wget/wget.html>>.
17. Iperf 1.2—The TCP/UDP Bandwidth Measurement Tool. Available at <<http://dast.nlanr.net/Projects/Iperf/>>.
18. Heegaard PE. GenSyn - a Java based generator of synthetic Internet traffic linking user behaviour models to real network protocols. Available at <<http://www.item.ntnu.no/~pouhl/publications/ITC-special.pdf>>.