Journal of Network and Systems Management, Vol. 10, No. 4, December 2002 (© 2002)

FOR PROOFREADING ONLY

An Integrated Framework for Implementing Quality of Network Concepts

Clayton Ferner[1,2] and Ron Vetter[1]

This paper presents an integrated framework for describing and implementing quality of network (QoN) concepts. It focuses on identifying a systems engineering framework for better understanding how to specify and implement QoN concepts. We demonstrate, by means of several examples, that the quality of service interfaces between the end user and the underlying network are very important. Finally, we provide insights as to future development efforts needed to realize the goal of high quality networks.

KEY WORDS: Quality of service; computer network; network architecture; network quality; network engineering; systems engineering.

1. INTRODUCTION AND BACKGROUND

Network reliability and quality are becoming increasingly important in our information-oriented society. As our world becomes more interconnected and interdependent, our reliance on computers and telecommunications systems will greatly increase. Network response time and equipment failures will significantly impact our ability to accomplish work and carry out business activities.

Recognizing the importance of building and maintaining fault-tolerant computer and telecommunication networks, the National Science Foundation sponsored a workshop on "Quality and Networks" at Arizona State University, Tempe, Arizona on February 25–26, 2000. The workshop brought together leading researchers and practitioners from various areas including: network and telecommunications engineers; quality, systems, and software engineers; producers and consumers of networks; and government agencies interested in systems approaches

---

[1] Department of Computer Science, University of North Carolina at Wilmington, Wilmington, North Carolina.

[2] To whom correspondence should be addressed at Department of Computer Science, The University of North Carolina at Wilmington, 601 South College Road, Wilmington, North Carolina 28403. E-mail: cferner@uncwil.edu

1064-7570/02/1200-0439/0 © 2002 Plenum Publishing Corporation

to quality. The workshop was divided into four panel areas: Quality and Process, Networking, Industry, and Government. Each of the panel areas consisted of recognized experts in their fields, and each panel member was invited to provide a statement on the general topic of quality and networks.

One immediate observation, evident from the comments made by the panelists, was that the term *network quality* and *quality of service* had different meanings to different people. A few of the panelists focused on total quality management, some focused on user behavior and expectations of the network, while others focused on application-level quality of service specification and implementation issues. It became clear that any attempt to view the conception, design, development, and maintenance of networks in isolation was inadequate. Rather, a systematic view of network quality within an interdisciplinary framework that encompasses end user and system level quality attributes is required. As a result of the discussions at this workshop, the term "quality of network (QoN)" was coined to refer to the idea of a generalized quality of service (QoS) architecture.

This paper does not attempt to provide a report of the workshop proceedings. Rather, the conclusions reached by the workshop participants provided the motivation for the development of a systematic and integrated framework in which quality of service research should be conducted. Other research venues have also reached similar conclusions (e.g., the Tenth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, 1999 [1]). A recent paper by Khan [2] also highlights some of the complementary techniques that must be integrated to provide different users and different applications with specific network performance guarantees. In addition, there are numerous research articles that focus on one aspect of the many problems associated with providing end-to-end QoS [3–13]. The goal of this paper is not to discuss the specific details any a single layer, but rather to provide a common QoN framework in which other researchers can work and thereby advancing overall research coordination in this area.

Many computer network designers view the quality of a network in terms of very specific metrics. On the other hand, end users think about the quality of a network in terms of *cost*, *availability*, *reliability*, *usability*, and *response time*. Concepts such as *jitter*, *cell loss ratio*, *network contention*, *token-bucket depth*, *maximum packet size*, *fault-tolerance*, and so on, are of little importance to most end users. They prefer to think of the quality of a network by asking questions such as: Is the network up or down? Will it stay up during the entire file transfer? Is a voice-over-IP (VoIP) phone call going to sound acceptable? How long do I have to wait to down load this web page? Can I access my files from anywhere any time? Furthermore, the definitions for acceptable parameters for these questions change from user to user.

As a direct result of this discrepancy in the definition of quality, the communication between the end user and the underlying network system is disconnected. At present, the end user does not have the ability to specify their expectations or requirements of the network, plus there is no feedback that would allow the user to make informed decisions about their requirements. Suppose a user is trying to view a web page that is taking a very long time to download. The absence of feedback leads to frustration, whereas the user would benefit from knowing that there are several network reservations currently tying up resources and that those reservations are set to expire in an hour. The user can then decide to continue with the slow download, try again in an hour, or attempt to make an advance reservation for network bandwidth.

Another example of the importance of feedback to the end user can be seen from a usability standpoint. Suppose that several users want to make a reservation for network bandwidth for a video conference call. If the reservation is simply denied without any feedback, the users either have to try later or make due with Best-effort service. Users would greatly benefit from some sort of feedback from the network, such as knowing that the video is the culprit for the denial of service and the reservation would go through if they would drop the video stream from the application. Perhaps this is a viable compromise, or perhaps it is not. Most likely, the video quality will be poor with Best-effort service anyway. Users would also benefit from feedback such as the reservation cannot be made now, but that it could be supported sometime in the near future. The feedback can also involve QoS *renegotiation*, such as for a lower bit rate video, especially if the application can employ video compression whose output rate can be varied on demand.

This paper presents an integrated framework for describing and implementing quality of network (QoN) concepts. Based on discussions at the NSF Workshop, we categorize the major issues in QoN as follows:

- End user expectations, quality characteristics, and behavior—this includes both qualitative and quantitative measures such as system usability and response time.
- Network architecture and design—this includes how core network architecture elements and protocols must be changed in order to take QoN concepts into account.
- Software and systems—better understanding how software components and reuse can be used to improve overall application quality. Additional areas of work include: performance tuning, conducting reliability studies, exploring the tradeoffs between memory and bandwidth, and investigating how new techniques such as active networking might be used to improve network quality.
- Network modeling and design techniques—the focus here is on engineering quality into the design of computer and telecommunication networks. For example, the official U.S. government modeling and analysis standard uses the Integration DEFinition (IDEF) approach for engineering quality into computer-based systems [14].

- Application-level quality of service (QoS) specification and implementation—how should an application specify its QoS attributes and how can they be consistently applied by the underlying network elements.
- Network operations, management, and measurement—deals with the design of tools and methodologies for efficient network operation. Additional data collection activities are needed to understand the interactions between multimedia data types and various system components (e.g., server bottlenecks vs. network bottlenecks).
- Human resource issues—how do we teach and/or train the designers of tomorrow's networks using today's technology (e.g., training vs. education; overall system view vs. specific skill/technology view).

Network engineering efforts must naturally span across multiple disciplines. Neither systems engineering nor process improvement is new. Many articles have been written reporting the success in improving processes by taking a systematic approach to solving complex problems [15]. Systems engineering often involves new challenges and solutions to old problems. This paper focuses on identifying a systems engineering framework for better understanding how to specify and implement QoN concepts.

The contribution of this paper is at least threefold:

- We promote and advocate a systems approach to engineering high quality networks.
- We illustrate how to conceptualize QoN concepts via an integrated systems framework.
- We provide insights as to future development efforts needed to realize the goal of high quality networks.

The rest of this paper is organized as follows: Section 2 presents the architectural model for quality of network; Section 3 discusses the details of each layer in the architectural model; Section 4 presents a typical scenario and discusses the issues involved with that example; and Section 5 concludes the paper.

2. ARCHITECTURAL MODEL

The network model we envision needs to provide an interface to the end user in terms to which he or she can relate. One concept that has generally been missing from existing communications network systems is that of *feedback*. Many large-scale communication systems have built-in measurement and monitoring systems to provide feedback on the real-time conditions and operations of the network. However, these feedback mechanisms often do not go all the way back to the end user of the system.

Figure 1 shows our proposed QoN architectural model. Each layer provides the QoS parameters that it considers important to the next lower layer. Each lower

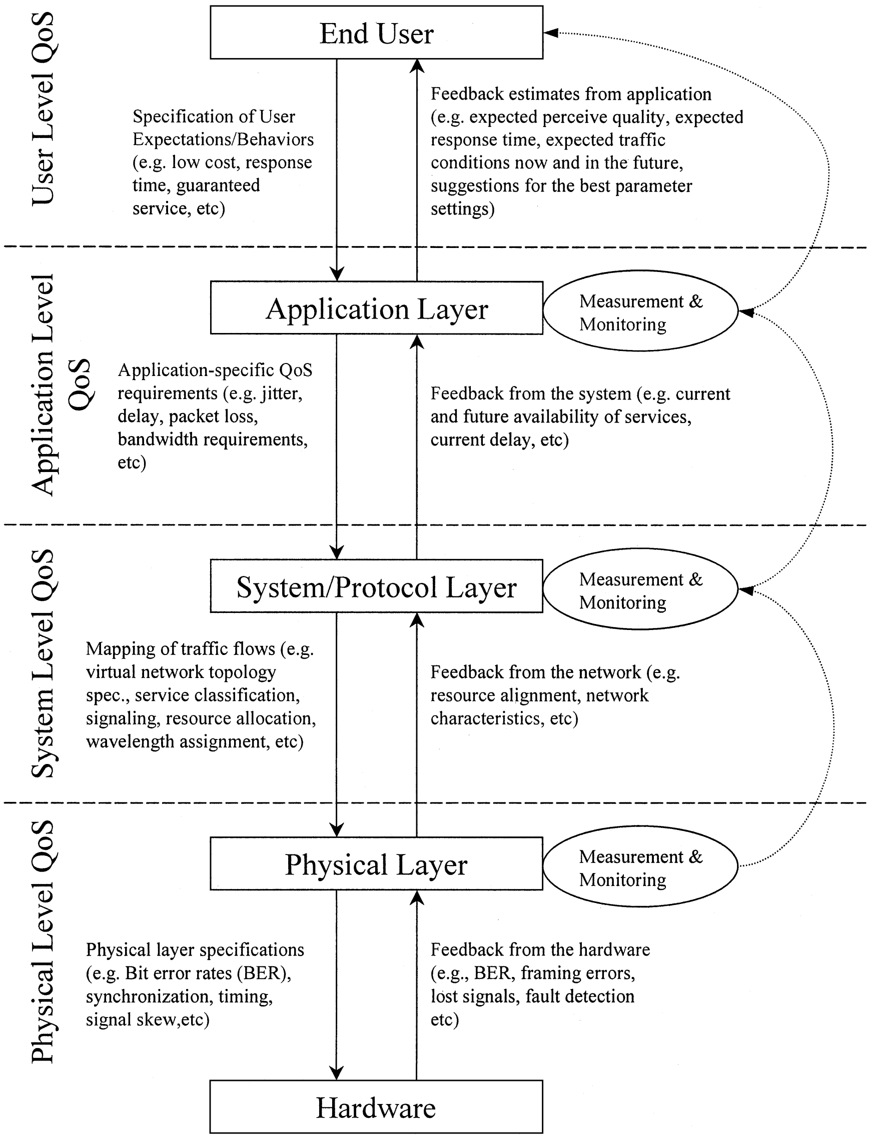

Fig. 1. QoN architectural model.

layer is responsible for translating the QoS parameters into a language it understands. It will in turn pass those QoS parameters to the layer below it in order to provide the QoN requested. Each layer will also need to pass information back to its higher layer. This information can be as brief as a pass/fail code or as elaborate as suggestions for parameter changes that will increase the chances for success.

Measurement and monitoring systems are designed to deal with the dynamic nature of networks [3, 4]. Not only do traffic conditions change but so does the topology of the network. Routers and servers (as well as connections) often go down or are taken off line with little or no notice. As the network changes, the QoN goals need to change to reflect the current conditions. The purpose of the monitoring subsystem is to provide to each layer the opportunity of adjusting the QoS parameters as conditions change. However, adaptation thresholds should be set with care to avoid traffic *oscillations*, which may result as different layers, while attempting to adapt to microchanges in the network traffic, cause microchanges in the network traffic due to the adaptation. Such behavior was observed on the Internet when earlier versions of TCP were quickly reacting to small changes in perceived network congestion.

3. A SYSTEMATIC VIEW OF QUALITY OF NETWORK

3.1. End User Layer Issues

The basic need that end users have for the network is communication. That communication comes in various forms: e-mail, facsimiles, voice/video calls, movies, pictures, downloadable files, etc. The form of communication usually dictates the need for quality guarantees. For example, Best-effort service is very appropriate for e-mail transmissions, but is less than sufficient for a voice telephone. However, users quite often want options or at least for the network to perform as advertised. Instead of the application completely dictating the QoN goals, users will likely want to fine-tune the goals to meet their particular needs.

For example, many vendors are now providing VoIP telephone services. VoIP service is ideal for those users who are willing to sacrifice quality for the benefit of costs savings. (VoIP calls are still substandard in quality when compared to traditional public-switched telephone services.) The other side of that spectrum is also possible. For example, a video conference call between corporate executives negotiating an important contract is a case where the end users will likely want the best QoN that money can buy.

In traditional data communication networks, QoS management was performed at the network device level without any interaction or interfacing with the end user. The emphasis was to provide reliability and throughput. With the growth of interactive multimedia applications, which have more stringent real-time requirements, interest has shifted to providing more tightly managed control over

Table I. End User-Level QoN Parameters

Levels of QoN	Costs	Availability	Reliability	Usability	Response time
Guaranteed-Reserved	$$$	Very low	High	High	Very good
Reserved	$$	Low	Medium	Medium	Good
Priority	$	High	Low	Low	Adequate
Best-effort	—	Very high	Very low	Very low	Poor

the QoS parameters. In addition, it is desirable, from the user standpoint, to have more flexible negotiation of QoS parameters in terms of cost/quality/performance with multimedia applications that can support *graceful* degradation of quality of their media streams.

Table I shows several levels of QoN from the end user's point of view. *Best-effort* is the service that the Internet currently provides. *Priority* is a level that accommodates many of the applications for which Best-effort is not sufficient (e.g., this might be viewed as differentiated services where there is a relative priority of service requests). *Reserved* would be a level that uses a resource reservation mechanism (e.g., this might be viewed as integrated services where there are prior resource reservations). *Guaranteed-Reserved* would be for the users who demand a high-level of service and are willing to pay what ever it takes to achieve their desired QoN.

At present, there is no means by which end users can reserve IP network services for future use in the context of their application, however, researchers are experimenting with technologies to implement this concept (e.g., Multicast Backbone (MBONE) technologies). This is an important feature that needs to become widely available. For example, a training session that is scheduled for a particular date and time should be reserved ahead to avoid the embarrassing situation where the resource reservation mechanism is unable to reserve the bandwidth at the very moment the session is suppose to begin.

Users are likely to also be concerned with very specific application-oriented issues such as response time, cost, screen layout, color depth, image size, audio quality, video frame rate, and so on. These user requirements, which serve as a service contract, must somehow be translated into QoS parameters that can be communicated to the application program and ultimately controlled by the underlying networked system. The translation step between each of the different layers of the QoN framework is called a *QoS mapping* [16]. During the connection establishment phase, network service negotiation takes place and the QoS parameters are mapped from the higher to lower layers. The primary goal of the network service negotiation step is to establish the parameters to be enforced within each layer of the QoN framework in order to attain the integrated QoS the end user desires.

While many of the end user requirements are quantifiable, many are qualitative. Somehow, these qualitative policies must also be mapped into specific

Table II. Application Layer QoS Parameters

Parameter	Units
Bandwidth	Bytes per unit time
Maximum end-to-end delay	Time
Jitter	Variation in delay
Packet loss	%

application and/or system layer to guarantee some *minimum* acceptable level of service for those policies. It is important to note that this mapping from the end user to the communication network subsystem is not yet well understood and is currently an active area of research [5]. The concept of electronic pricing and billing has also not been studied extensively, although a recent study [17] made some progress toward understanding user perspective of QoS and pricing.

3.2. Application Layer Issues

Most computer-based applications have differing traffic characteristics and hence unique QoS needs. For example, e-mail transmissions have much different QoS needs than do live audio transmissions. Table II lists some of the QoS parameters that are important from an application layer point of view. These QoS parameters are not necessarily independent. For example, the packet loss (and associated retransmission) may affect both end-to-end delay as well as jitter.

Table III lists several multimedia applications and the level of importance of each QoS parameters mentioned above.

How an application determines its QoS parameters (if not dictated by the user and/or not well-defined by the specific application domain area) is still an area of active research. Not only does an application need to determine default QoS parameters, but they must also allow direct input from a user. An application program might provide a setup window where the user can specify the requirements for various parameters or perhaps the values can be read from a user-specified

Table III. Multimedia Applications and the Importance of Each QoS Parameter

Application	Bandwidth	End-to-end delay	Jitter	Packet loss
Batch data transfer (e-mail, fax, downloads)	Low	Low	Low	Moderate[b]
Live audio	Moderate	High[a]	High	Moderate
Live video	High	Moderate	High	Moderate
Prerecorded audio	Moderate	Low	Low	Low[b]
Prerecorded video	Moderate	Low	Low	Low[b]
Real-time data transfer	High	High	High	High
Mobile services & wireless services	High	High	High	High

[a] <400 ms.

[b] Retransmission is possible.

configuration file. However, the service provided may or may not match users' expectations. For example, if a user specifies Best-effort and wants low delay and high bandwidth, the user may be disappointed in the perceived QoS when the network traffic is heavy. On the other hand, if a user specifies Guaranteed-Reserved and wants a large amount of bandwidth, they are likely to be denied service because the network simply cannot provide it *and* guarantee the quality. This is where some sort of feedback would be helpful.

It is difficult to define a generic set of QoS parameters that can be used for every kind of application and/or specific media type. The objective is to list a set of QoS parameters that the network will support so that the QoS mapping functions of the previous section can be applied. Moreover, it is unlikely that the same application would require and/or use the same QoS parameter specification in all instances. The user requirements as well as the capabilities of the underlying transport system will greatly impact the parameter values used to fulfill a specific service contract. For example, a given user may be tolerant of poor quality video if it means reducing costs. Also, an application might easily support a given QoS specification over a high speed wired network, but might have difficulty if the underlying transport system becomes wireless as in the case of a user who suddenly becomes mobile. Clearly, application programs need to be able to specify QoS parameters that are suitable to the end user and the operating environment in which they are going to be applied. Furthermore, QoS renegotiation becomes more and more important as resources vary. As resources become unavailable, the QoS will degrade and the QoS parameters should be adjusted accordingly. Conversely, as resources become available, the QoS can be upgraded.

Research into these issues at the application level is very active. References [6–8] are recent papers that are representative of the ongoing work into QoS at the application level.

3.3. System/Protocol Layer Issues

At this layer in the QoN architecture, we are concerned with several issues. First, we must be able to translate specific application layer QoS parameters into system resource allocation requests and ensure that system resources (CPU, memory, bandwidth, I/O ports, etc) are allocated efficiently. Second, we must classify each of the various media flows into appropriate service classifications and map those flows onto underlying physical network circuits. Third, we must be able to reconfigure the network by establishing virtual paths through the system that reserve network resources in such a way that they conform to the service contract specified by the application and/or end user layer. Finally, network functions, such as connection admission control, usage parameter control, signaling procedures, and feedback control are often implemented by this layer. Reference [9] is an example of research on QoS at the system layer.

The system layer must support renegotiation of QoS for the application layer. This is important because of the dynamic nature of networks. However renegotiation is essential for wireless networks, where the mobility itself of the end user can cause changes in the network capabilities. For example, as an end user moves from one zone to another, packet delay and packet loss can change. Even the distance to a receiving station can affect network performance.

Until recently there has not been a standard API that allowed applications to invoke system/protocol layer functions. A new Windows-based networking API, called Winsock2, is changing this situation. Winsock2 defines the data structures and calls that enable applications to signal their requirements (e.g., features like enhanced multiprotocol standards, multicast capabilities, and QoS requirements) to the underlying network architecture [10]. This will greatly assist applications developers to write QoS-enabled applications and contribute to overall network quality.

Several other new networking developments are also worth mentioning. A recent article [18] describes how differentiated optical services might be provided to the application layer. The article illustrates how an underlying wavelength division multiplexed (WDM) network can support QoS-aware differentiated services by allocating end-to-end all-optical light paths with each light path providing unique QoS characteristics (e.g., specific bit error rates (BER), jitter, delay, security level, etc).

Equally interesting is the concept of *active networking* [19]. Active networks remove the distinction between the network and connected computing devices by enabling the dynamic execution of program code on the network elements. With the network becoming programmable, users and applications will be able to customize the network and allocate resources dynamically as their needs change. Using *smart packets* [19], users will be able to dynamically download code (i.e., functionality) onto internal network elements to enable the networking infrastructure itself to directly provide the features most suited for a particular application. When combined with software agent technology, active networking promises to add a new dimension to the design and management of networks. Little work has been reported on using these technologies in the area of network quality and network management.

Additional research is required before we can better understand how we might use the concept of optical differentiated optical services and active networking with mobile agent techniques to provide additional flexibility and higher quality to the system/protocol layer.

3.4. Physical Layer Issues

Table IV lists some of the QoS parameters we consider important at the physical layer. The physical layer is an area that has been well studied and is

Table IV. Important Physical Layer QoS-Related Parameters

Equipment capabilities	• Programmable switches • All-optical wavelength routing • Multicast capability
Multiplexing methods	• Frequency division multiplexing (FDM) • Time division multiplexing (TDM) • Wavelength division multiplexing (WDM)
Error detection	• Forward error control • Feedback error control
Fault tolerance	• Redundant communication channels • Real-time monitoring • Measurement and statistics gathering

generally well understood. We only discuss the issues here that we feel are of importance to the higher layers, particularly the end-user.

The underlying capabilities of the network equipment allow the network to adapt to the specific needs of an application. For example, programmable switches allow users to download application-specific code directly into internal network elements thereby enabling new features that will benefit the application. All-optical wavelength routing is capable of directing an input signal to a specified output port based on its wavelength [20]. Additionally, a virtual network topology can be setup over a set of independent wavelengths with each wavelength characterizing a unique QoS specification. Wavelength routers and switches can route messages directly in the optical domain without ever converting the signals to the electrical domain [21]. Other WDM architectures, such as broadcast-and-select WDM networks, provide an inherent multicast capability directly. Such a capability could be put to good use if applications knew of the networks ability to provide such a service.

The multiplexing method should also be chosen according to the needs of an application. A point-to-point audio transmission is an obvious case for time division multiplexing (TDM). Frequency division multiplexing (FDM) is a good choice for mobile applications or satellite transmissions. WDM has been used to increase the total bandwidth for optical network. However, there is a trend toward using the different wavelengths to provide different levels of service.

Similarly, the application should have some control over the error detection method used. When transmitting data files where retransmission is permissible, feedback control is adequate. However, with real-time/interactive data a more aggressive approach, like forward error control, is a better choice.

Fault tolerance is also important to provide high quality services. Equipment and software must continually monitor the network and provide information back to the user. Redundant communication channels can be used to improve network path reliability, real-time monitoring can inform system operators of fault conditions, and measurement and statistics gathering can be used to better understand how system resources are being used.

4. AN EXAMPLE: A DISTANCE LEARNING APPLICATION

As stated previously, an interdisciplinary systems approach to network quality that encompasses end user, application, system, and physical layer quality attributes is required to engineer high quality networks. To better understand the overall nature of the issues involved, consider the following example:

> A professor at a university is offering a night course in educational development, which is primarily attended by teachers. However, the majority of her students do not live in the same city nor do they physically attend class. Some live as far away as 1000 miles. She prepares for her class in the usual way: writing lecture notes, creating examples, choosing exercises from the textbook, etc. However, she develops the materials electronically and loads them on a web page where all the students can gain access.
>
> In order for the class to provide the same level of interaction as with a normal class, the professor sets up a discussion session involving live voice and video. She registered for the bandwidth a week earlier using an adaptive voice/video application. This application has an option window where the user can request certain QoS parameters. She selected the Reserved option and provided the date and hour the class would meet. The remaining options for delay, bandwidth, and packet loss she left with their default values. She also had to provide a file of IP addresses for each machine that would be used by her students.
>
> The voice/video application then proceeded to establish the bandwidth by making a request to the network subsystem. Each router between the professor's computer and the students' computers accepted the request and returned a success status plus an electronic bill for the reservation. The application displayed a window to the professor indicating that her request was granted and the total cost, and requesting payment information that would then be returned to the routers. The cost of the reservation had already been included in the tuition for the course. Finally, she emails the reservation number and access key to all each of her students.
>
> When class begins, the students provide the reservation information to their voice/video applications to start the online session. A video window with the professor's image appears on their screens, and they can hear her voice over their PC speakers. The students can also speak in turn so that the entire class can interact. Each student also opens a chat session specifically for questions in order to conserve bandwidth.
>
> As she presents the lecture materials, the slides appear on each student's machine as well as a second mouse pointer so that she can highlight certain points or make annotations to the slides. The professor then presents the students with a worksheet and asks several of them to fill in missing information. The students can save the slides to their machines to study later.
>
> During the lecture, several students complain via the chat session that the audio and video is becoming choppy. Shortly thereafter, a window appears on the professor's computer indicating that a router is experiencing timeouts with its transmissions to another router and suspects that it may be down. The router has in the meantime switched to another route, but can no longer guarantee the same the level of QoS. The professor decides to stop transmitting the video so see if that frees up some bandwidth.

> The students inform her that the voice quality has improved slightly to an acceptable level.
>
> Towards the end of the class, the professor uses several video segments of real world examples to further emphasize her lessons. At the end of the video, she facilitates further discussion.

The example above has several interesting qualities. First, the interactive session involving voice and video requires fairly stringent QoS parameters. The real-time nature of the application, the synchronization of audio with video, and the mixing of different voices require that the jitter and end-to-end delay be kept to a minimum. Also, the fact that the class is scheduled means that the professor must reserve the bandwidth ahead of time to insure that the session can go as planned. Naturally, this will be fairly expensive. One option to reduce the bandwidth requirement and relax the QoS parameters slightly would be to have a voice-only session.

Second, this example uses real-time data collaboration. The same network reservation could be used for this application, but the QoS requirements for the data collaboration are different that those of the voice and video. Delay and jitter are less important, but data integrity is very important. Unfortunately, because this session must be real-time means that Best-effort will likely not be sufficient. There are two options the professor has: 1) she can use the same reservation for the voice and video for the data-collaboration, which have different QoS parameters, or 2) she can make a new reservation or use the Priority server, which will add to the cost of the session.

Third, this example uses a chat session. The chat session can use Best-effort service in order to conserve the bandwidth used for the more expensive applications. This is because the voice application is further complicated by the requirement of mixing the voices. This adds additional buffers to the voice transmissions. Using the chat session specifically for questions will reduce the number of speaking voices and will improve the quality. The professor can answer some of the questions orally if she chooses.

Fourth, this example uses file transfers for downloading html, assignments, lecture notes, video segments, etc. plus uploading assignments. Best-effort service will be sufficient for this, especially if the students can log on before class begins to get the materials. Even the recorded video can be downloaded ahead of time to reduce the costs.

Fifth, this example illustrates the usefulness of feedback. Armed with the knowledge that a router is not responding, the professor has more options to handle the situation of degradation in audio and video quality. One option is to stop transmitting the video to those students involved or to all students. Another would be to end the data collaboration session. Regardless of her options, without this feedback the users are much more likely to become frustrated.

Table V lists the applications, the QoS requirements, and issues for this example.

Table V. QoS Requirements for Distance Education

Layer	Requirements	Feedback
Application: Live conference audio/video		
End-user layer	• Reserve in-advance[a]	• Expected costs
	• Good quality audio/video	• Success/failure
	• Fast response time	• Suggested changes
	• Not excessively expensive	• Electronic pricing
Application layer	• Low jitter/low delay	• Success/failure
	• Bandwidth (MPEG quality is 8–20 Mbps)	• Suggested parameters
	• Multicasting if available	• Electronic pricing
System layer	• Future reservation protocol	• Success/failure
	• Dedicated system resources	• Traffic info.
	• Multicasting if available	• Electronic pricing
Physical layer	• Monitor and collect stats	• Lost signal
	• Minimal error detection	
	• Preferably TDM or WDM	
Data collaboration		
End-user layer	• Real-time	• Success/failure
	• Not excessively expensive	• Traffic info.
Application layer	• Low jitter/low delay	• Success/failure
	• Guaranteed packet delivery	• Suggested parameters
	• Low bandwidth	
	• Multicasting if available	
System layer	• Multicasting if available	• Success/failure
		• Traffic info.
Physical layer	• Feedback error control	
Recorded video		
End-user layer	• Low cost[b]	• Success/failure
	• Near real-time	• Suggested parameters
Application layer	• Best-effort or priority	• Success/failure
	• Low packet loss	• Suggested parameters
	• Multicasting if available	• Packet loss and delay
System layer	• Multicasting if available	• Success/failure
		• Variance in delay
Physical layer	• Forward error control	
File transfer for course materials		
End-user layer	• Low cost	• Success/failure
Application layer	• Best-effort	• Success/failure
	• Guaranteed packet delivery	
System layer	• TCP/IP services	• Success/failure
Physical layer	• Feedback error control	

[a] If reservation is unavailable, try compromising on jitter/delay, dropping video, or changing the class meeting time.

[b] The video segments can be streamed across the network to reduce costs.

5. CONCLUSIONS

A recent NSF workshop exposed the disconnected communication between the different layers of the network architectural model in terms of end user expectations, QoS specification, and feedback required for realizing high quality networks. Most evident was the fact that QoS meant different things to different people. What is needed is an architectural model that can incorporate the issues raised at the workshop within a framework at which others can work. We have presented such an architectural model in this paper. At the very least, the definition of QoS should be differentiated at the different levels. However, having separate QoS definitions is not the complete solution. The quality needs at one layer are related to the quality needs at other layers. It is essential that inter-layer communication exist specific to QoS.

We have demonstrated, by means of several examples, that the communication between the end user and the underlying network systems is very important. There has been great progress through new network features and protocols such as differentiated services, resource reservation protocol (RSVP), and wavelength division multiplexing [11, 20, 22, 23]. However, new features are only part of the solution. Network systems designers should be aware that the end user is the real customer, and it is their satisfaction that is the ultimate goal. For example, the availability of reserving bandwidth for future use and feedback to keep the end user informed of network conditions will go a long ways toward this goal even with the current set of available features.

We have described an architectural model that we feel would greatly benefit the end user. However, there are several things that must be done before the model can be fully implemented. The protocols between layers need to be expanded to include specifications of QoS parameters used by the next higher level. Feedback from each layer as well as from lower layers should be passed upwards. Applications need to be designed to communicate with the end users in a language they understand and to provide feedback that is informative.

One of the important open issues is how much overhead will all of this generate. Obviously, the concepts discussed in this paper will create some overhead at all layers of the network. There will be a tradeoff between the Quality of Network and the improvement to Quality of Service. Where should the tradeoff be made? Will some or all of the concepts provide enough benefit to be worth the effort?

REFERENCES

1. R. Stadler and B. Stiller, Active technologies for network and service management: Report on DSOM'99, *Journal of Network and Systems Management*, Vol. 8, No. 1, pp. 125–130, 2000.
2. M. Khan, QoS: What is it all about?, *Journal of Network and Systems Management*, Vol. 9, No. 4, pp. 369–373, 2001.

3. Y. Jiang, C.-K. Tham, and C.-C. Ko, Providing quality of service monitoring: Challenges and approaches, Proceedings of the *2000 IEEE/IFIP Network Operations and Management Symposium* (*NOMS 2000*), Honolulu, Hawaii, pp. 115–128, 2000.
4. E. S. H. Tse-Au and P. A. Morreale, End-to-end QoS measurement: Analytic methodology of application response time vs. tunable latency in IP networks, Proceedings of the *2000 IEEE/IFIP Network Operations and Management Symposium* (*NOMS 2000*), Honolulu, Hawaii, pp. 129–142, 2000.
5. http://www.qosforum.com.
6. P. P. Demestichas, V. P. Demesticha, Y. I. Manolessos, G. D. Stamoulis, and M. E. Theologou, QoS management by means of application control, *Journal of Network and Systems Management*, Vol. 7, No. 2, pp. 177–197, 1999.
7. P. G. S. Florissi, Y. Yemini, and D. Florissi, QoSockets: A new extension to the sockets API for end-to-end application QoS management, Proceedings of the *Sixth IFIP/IEEE International Symposium on Integrated Network Management* (*IM'99*), Boston, MA, pp. 655–668, 1999.
8. P. Wang, Y. Yemini, D. Florissi, and J. Zinky, A distributed resource controller for QoS applications, Proceedings of the *2000 IEEE/IFIP Network Operations and Management Symposium* (*NOMS 2000*), Honolulu, Hawaii, pp. 143–156, 2000.
9. P. Moghé and I. Rubin, Managing connection-level QoS through an Overlay service manager, *Journal of Network and Systems Management*, Vol. 4, No. 4, pp. 397–424, 1996.
10. C. Metz, IP QOS: Traveling in first class on the internet, *IEEE Internet Computing*, Vol. 3, No. 2, pp. 84–88, 1999.
11. Y. Bernet, The complementary roles of RSVP and differentiated services in the full-service QoS network, *IEEE Communications Magazine*, Vol. 38, No. 2, pp. 154–162, 2000.
12. C. Aurrecoechea, A. Campbell, and L. Hauw, A survey of QoS architectures. In *Multimedia Systems*, *Volume 6*, Springer-Verlag, New York, pp. 138–151, 1998.
13. G. Huston, Next Steps for the IP QoS Architecture, *Internet Engineering Task Force (IETF) Request for Comments (RFC) 2990*, November 2000.
14. http://hissa.ncsl.nist.gov/~ftp/idef/.
15. S. Bhattacharya, J. M. Capone, K. Dooley, S. Palangala, and H.-S. Yang, The network maturity model for Internet development, *IEEE Computer Magazine*, Vol. 32, No. 10, pp. 117–118, 1999.
16. J. Domingo-Pascual and J. Mangues-Bafalluy, A framework for adaptive applications, Technical report of the Department of Architecture and Computers, Report number: UPC-DAC-1998-7, Barcelona, Spain, November 5, 1998. Available at ftp://ftp.ac.upc.es/pub/reports/DAC/1998/UPC-DAC-1998-7.ps.Z.
17. A. Bouch and M. A. Sasse, It ain't what you charge, it's the way that you do it: A user perspective of network QoS and pricing, Proceedings of the *Sixth IFIP/IEEE International Symposium on Integrated Network Management* (*IM'99*), Boston, MA, pp. 639–654, 1999.
18. N. Golmie, T. D. Ndousse, and D. H. Su, A differentiated optical services model fro WDM networks, *IEEE Communications Magazine*, Vol. 38, No. 2, pp. 68–73, 2000.
19. D. Wetherall, U. Legedza, and J. Guttag, Introducing new Internet services: Why and how, *IEEE Network Magazine*, Vol. 12, No. 3, pp. 12–19, 1998.
20. J. M. H. Elmirghani and H. T. Mouftah, Technologies and architectures for scalable dynamic dense WDM networks, *IEEE Communications Magazine*, Vol. 38, No. 2, pp. 58–66, 2000.
21. S. Yao and S. Dixit, Advances in photonic packet switching: An overview, *IEEE Communications Magazine*, Vol. 38, No. 2, pp. 84–94, 2000.
22. C. Metz, RSVP: General-purpose signaling for IP, *IEEE Internet Computing*, Vol. 3, No. 3, pp. 95–99, 1999.
23. P. P. White, RSVP and integrated services in the Internet: A tutorial, *IEEE Communications Magazine*, Vol. 35, No. 5, pp. 100–106, 1997.

Clayton Ferner is an assistant professor of the Department of Computer Science at the University of North Carolina at Wilmington. His research interests are computer networks, parallel and distributed computing, and parallel compilers for parallel and distributed computing. He received his Ph.D. in mathematics and computer science from the University of Denver in 1997.

Ron Vetter is a professor and chair of the Department of Computer Science at the University of North Carolina at Wilmington. His research interests include mobile and wireless networking, web-centric computing, and distance education. He received his Ph.D. in computer science from the University of Minnesota in 1992.