An Architectural Framework for Support of Quality of Service in Packet Networks

Hui-Lan Lu and Igor Faynberg, Bell Laboratories/Lucent Technologies

ABSTRACT

This article gives an overview of the effort underway in ITU-T SG 13 on an architectural framework for QoS support in packet networks, with a focus on IP. Provisionally named Y.qosar, the framework is to be published as a new ITU-T Recommendation. At the center of the architectural framework is a set of QoS network mechanisms distributed across three logical planes (the control, data, and management planes) to control network performance. Ultimately the network mechanisms are to be used in combination to deliver the satisfactory collective effect of service performance. The article also provides pointers to standards efforts dealing with specific QoS network mechanisms.

INTRODUCTION

Quality of service (QoS) has been a subject of active research and standardization since the advent of telecommunications technology. The International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) has done much work on QoS, whether in relation to performance metrics or network mechanisms to deliver the required performance or a comprehensive definition. Among the related standards, ITU-T Recommendation E.800 defines QoS as "the collective effect of *service performance* which determines the degree of satisfaction of a user of the service." Given that E.800 considers support, operability, serviceability and security all part of service performance, this QoS definition is comprehensive in scope. Expanding on the E.800 QoS concept, ITU-T Recommendation G.1000 breaks down *service performance* into functional components and links it to network performance such as defined in ITU-T Recommendations I.350, Y.1540, and Y.1541. Complementary to G.1000, which defines a framework, ITU-T Recommendation G.1010 specifies end-user-centric application requirements in terms of broad categories (such as interactive and error tolerant).

In relation to QoS network mechanisms, ITU-T was initially focused on those specific to the public switched telephone network (PSTN). The PSTN employs a routing model based on the concept of *circuits* (or end-to-end physical connections). In such a network, it is possible to determine whether a session that requires certain performance objectives can be established and whether the performance can be guaranteed throughout the duration of the established session. The routing model, stemming from the support of telephony, has been applied to the development of a *virtual circuit* for data communications technologies such as X.25, frame relay, and broadband integrated services digital network (B-ISDN).

Concerning B-ISDN, based on the output of the ATM Forum, ITU-T Recommendations I.356 and I.610 specify performance measurement methods and QoS objectives for end-to-end connections, and define operations and management tools to monitor these parameters. In addition, the ITU-T Q.29xx Recommendation series prescribes mechanisms for negotiating traffic parameters and QoS performance objectives. Because of its connection-oriented virtual-circuit approach, support of QoS in asynchronous transfer mode (ATM) is more or less straightforward: the characteristics of the virtual circuit are what must be negotiated among the participants of a session (and between each participant and the network).

The IP routing model, by design, has avoided stressing any in-built mechanism for creation and maintenance of virtual circuits. IP networks have supported what is called *best-effort* (with no guarantee whatsoever) packet transfer. There is neither differentiation among various types of traffic, nor guarantee of in-sequence packet deliveries, nor guarantee of the arrival of each packet. Whatever the end-to-end performance requirements may be, at the network layer packets travel from router to router. Each router queues newly arriving packets for transmission over the link to the most suitable (according to the routing table) router or destination host. With this inherently connectionless and stateless nature, guaranteeing service or network performance in an IP network is a much more complex

0163-6804/03/$17.00 © 2003 IEEE

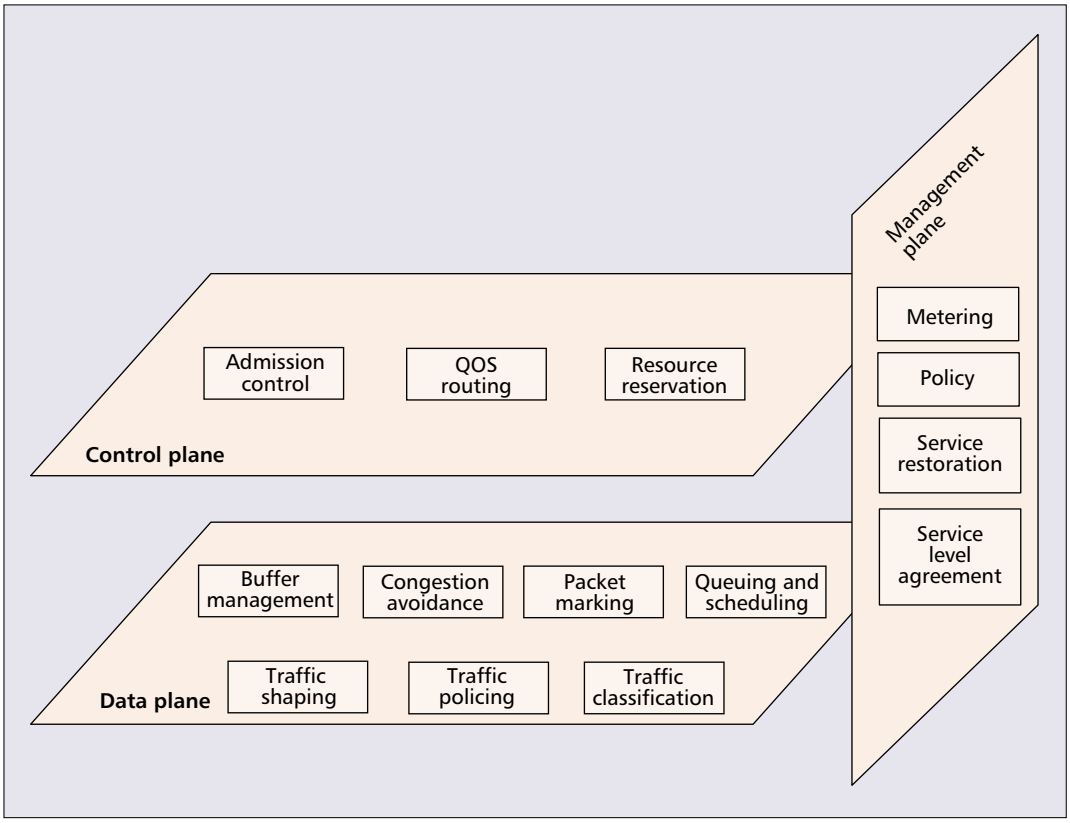

Figure 1. *QoS building blocks.*

IETF RFC 2990 summarizes the possible characteristics of a controlled service response: consistent and predictable, at a level equal to or above a guaranteed minimum, or established in advance.

matter. This explains why IP QoS remains a subject of ongoing standardization in the ITU-T, Internet Engineering Task Force (IETF), and other standards bodies.

This article describes the effort underway in ITU-T SG 13 on an architectural framework for QoS support in packet networks, with a focus on IP. Provisionally named Y.qosar, the framework will be published as a new ITU-T Recommendation. It is intended to identify a set of generic QoS network mechanisms and provide a structure for them. Ultimately, the network mechanisms will be used in combination to collectively deliver satisfactory service performance. Different services (or applications), however, may have quite different needs. For example, for telemedicine the accuracy of the delivery is more important than overall delay or packet delay variation (i.e., jitter), while for IP telephony, jitter and delay are key and must be minimized.

Given the trend of providing a wide range of applications of varying performance requirements over IP networks, the framework is envisaged to include a diverse set of generic QoS network mechanisms. Note that there are defined standards or ongoing standards efforts dealing with specific QoS mechanisms. The article provides pointers to them as appropriate.

An Overview of the Architectural Framework

Central to the QoS architectural framework is a set of generic building blocks for controlling and delivering the network service response to a service request, especially when there is network resource contention. IETF RFC 2990 summarizes the possible characteristics of a controlled service response: consistent and predictable, at a level equal to or above a guaranteed minimum, or established in advance.

An initial set of QoS building blocks has been identified. As depicted in Fig. 1, the building blocks are organized according to three logical planes:

Control plane: Contains mechanisms dealing with the pathways through which user data traffic travels. These mechanisms include admission control, QoS routing, and resource reservation.

Data plane: Contains mechanisms dealing with the user data traffic directly. These mechanisms include buffer management, congestion avoidance, packet marking, queuing and scheduling, traffic classification, traffic policing, and traffic shaping.

Management plane: Contains mechanisms dealing with the operation, administration, and management aspects of the user data traffic. These mechanisms include metering, policy, service level agreement (SLA), and traffic restoration.

We will further discuss some of the building blocks for each plane in later sections. For now let us examine their general properties.

A QoS building block may be specific to a network node (as exemplified by buffer management) or applicable to a network segment (as exemplified by QoS routing). The latter, in particular, requires signaling between network nodes, whether they are part of a network segment that is end to end, end to edge, edge to

The service provider establishes with each user a service level agreement, which, among other things, specifies how much traffic a user may send within any given class of service. The traffic is then policed at the border of the service provider's network.

edge, or network to network. Signaling can take place in any of the three logical planes. When taking place in the control or management plane, signaling entails the use of a signaling protocol. Because of its unique properties, we will discuss signaling in a separate section.

To illustrate how various QoS approaches can make use of the building blocks, we consider as examples three standardized approaches: integrated services (IntServ), differentiated services (DiffServ), and multiprotocol label switching (MPLS).

Primarily for supporting real-time delay-sensitive applications, the IntServ approach is built on the understanding that a flow serviced at a rate slightly higher than its data rate has a bounded delay, and the network can guarantee the delay bound of a flow by per-flow resource reservation. With this approach, an application, before sending data, first signals to the network the desired service request, including specifics such as its traffic profile and bandwidth and delay requirements. The network then determines whether it can allocate adequate resources (e.g., bandwidth or buffer space) to deliver the desired performance of the service request. Only after the request is granted can the application start to send data. As long as the application honors its traffic profile, the network meets its service commitment by maintaining per-flow state and using advanced queuing disciplines (e.g., weighted fair queuing) for link sharing. The building blocks relevant to the IntServ approach include admission control, queuing, resource reservation, traffic classification, and traffic policing.

The concept behind the DiffServ approach is treating a packet based on its class of service as encoded in its IP header. The service provider establishes with each user a SLA (or service level specification, SLS), which, among other things, specifies how much traffic a user may send within any given class of service. The traffic is then policed at the border of the service provider's network. Once the traffic enters the network, routers provide it with differentiated treatment. In contrast to the IntServ approach, the treatment is based not on a per-flow basis, but solely on the indicated class of service. The overall network is set up to meet all SLAs. The building blocks relevant to the DiffServ approach include buffer management, packet marking, SLA, traffic metering and recording, traffic policing, traffic shaping, and scheduling.

Initially developed for the purpose of interworking between IP and ATM (or frame relay) networks, MPLS achieves substantial gains in packet forwarding speed through the use of short layer-2-like labels. Upon entering the MPLS network, a packet is assigned once and for all a forward equivalence class (FEC), which is encoded as a fixed-length string known as a label. When the packet is forwarded to the next hop, the label is sent along with it. At the next hop, the label is used as an index into a preconfigured table to identify the following hop and a new label. The old label is replaced with the new label, and the packet is forwarded to the following hop. The process continues until the packet reaches the destination. In other words, packet forwarding in MPLS is entirely label-driven, whereby packets assigned the same FEC are forwarded the same way. Furthermore, labels are meaningful only to the pair of routers sharing a link, and only in one direction: from a sender to the receiver. The receiver, however, chooses the label and negotiates its semantics with the sender by means of a label distribution protocol. MPLS in its basic form is particularly useful for traffic engineering. To provide explicit QoS support, MPLS makes use of certain elements in the IntServ and DiffServ approaches. The label distribution protocol, for example, can be based on a resource reservation protocol (RSVP) [1]. With it, required network resources along a label switched path can thus be reserved during its setup phase to guarantee the QoS of packets traveling through the path. In addition, by using the label and certain EXP bits of the shim header that carries the label to represent the DiffServ classes, packets of the same FEC can be subject to DiffServ treatment [2]. The relevant building blocks for MPLS include buffer management, packet marking, QoS routing, queuing, resource reservation, traffic classification, and traffic shaping.

Control Plane Mechanisms

Admission Control

This mechanism controls the traffic to be admitted into the network, preferably in such a way that newly admitted traffic does not result in network overload or service degradation to existing traffic. Normally admission control is policy-driven [3]. Policies are a set of rules for administering, managing, and controlling access to network resources. They can be specific to the needs of the service provider or reflect the agreement between the customer and service provider, which may include reliability and availability requirements over a period of time and other QoS requirements. To satisfy reliability and availability needs for certain services (e.g., emergency communications), associated traffic can be given higher than normal priority for admission to the network. Specifically, Y.qosar has defined four priority levels for admission control.

An admission decision can also depend on adequate network resources being available to meet the performance objectives of a particular service request. In this case, there are two common approaches: parameter-based and measurement- based. The parameter-based approach derives the worst case bounds for a set of metrics (e.g., packet loss, delay, and jitter) from traffic parameters and is appropriate for providing *hard* QoS for real-time services. It is often used in conjunction with resource reservation in order to effect the guaranteed bounds. In contrast, the measurement-based approach uses measurements of existing traffic for making an admission decision. It does not guarantee throughput or hard bounds on certain metrics, and is appropriate for providing *soft* or relative QoS. This approach generally has higher network resource utilization than the parameter-based one.

QoS Routing

QoS routing concerns the selection of a path satisfying the QoS requirements of a flow. The path selected most likely is not the traditional shortest path. Depending on the specifics and the number of QoS metrics involved, computation required for path selection can become prohibitively expensive as the network size grows. Hence practical QoS routing schemes consider mainly cases for a single QoS metric (e.g., bandwidth or delay) or for dual QoS metrics (e.g., cost-delay, cost-bandwidth, and bandwidth-delay). To further reduce the complexity of path computation, various routing strategies exist. According to how the state information is maintained and how the search of feasible paths is carried out, there are strategies such as source routing, distributed routing, and hierarchical routing [4]. In addition, according to how multiple QoS metrics are handled, there are strategies such as metric ordering and sequential filtering, which may trade global optimality with reduced computational complexity [5].

The path selection process involves the knowledge of the flow's QoS requirements and characteristics and (frequently changing) information on the availability of network resources (expressed in terms of standard metrics, e.g., available bandwidth and delay). The knowledge is typically obtained and distributed with the aid of signaling protocols. For example, RSVP can be used for conveying a flow's requirements and characteristics and Open Shortest Path First (OSPF) extensions as defined in IETF RFC 2676 for resource availability. Compared with shortest path routing that selects optimal routes based on a relatively constant metric (i.e., hop count or cost), QoS routing tends to entail more frequent and complex path computation and more signaling traffic [6].

It is important to note that QoS routing provides a means to determine only a path that can likely accommodate the requested performance. To guarantee performance on a selected path, QoS routing needs to be used in conjunction with resource reservation to reserve necessary network resources along the path. ITU-T SG 2 has an effort underway to develop a comprehensive set of Recommendations on QoS routing.

QoS routing can also be generalized to apply to traffic engineering, which concerns slowly-changing traffic patterns over a long time scale and a coarse granularity of traffic flows. To this end, routing selection often takes into account a variety of constraints such as traffic attributes, network constraints, and policy constraints [7]. Such generalized QoS routing is also called constraint-based routing, which can afford path selection to bypass congested spots (or to share load) and improve overall network utilization as well as automate enforcement of traffic engineering policies.

Resource Reservation

This mechanism sets aside required network resources on demand for delivering desired network performance. Whether a reservation request is granted is closely tied to admission control. All the considerations for admission control therefore apply. But, in general, a necessary condition for granting a reservation request is that the network has sufficient resources.

The exact nature of a resource reservation depends on network performance requirements and the specific network approach to satisfying them. For example, in the IntServ approach, simplex flows are what matter and are characterized in terms of parameters describing a token bucket, and receiver-initiated reservations are done on demand according to peak rate requirements to guarantee delay bounds. Regardless of the specifics, it is important for service providers to be able to charge for the use of reserved resources. Therefore, resource reservation needs support for authentication, authorization, and accounting and settlement between different service providers.

Resource reservation is typically done with a purpose-designed protocol such as RSVP [8]. To date, however, no existing resource reservation protocol is regarded suitable for large-scale deployment. An effort underway in the IETF may lead to the development of an improved resource reservation protocol, which is further discussed in the section on QoS signaling.

Resource reservation is typically done with a purpose-designed protocol such as RSVP. To date, however, no existing resource reservation protocol is regarded as suitable for large-scale deployment.

Data-Plane Mechanisms

Buffer (or Queue) Management

Buffer (or queue) management deals with which packets, awaiting transmission, to store or drop. An important goal of queue management is to minimize the steady-state queue size while not underutilizing links, as well as preventing a single flow from monopolizing the queue space. Schemes for queue management differ mainly in the criteria for dropping packets and what packets (e.g., the front or tail of the queue) to drop. The use of multiple queues introduces further variation, for example, in the way packets are distributed among the queues.

A common criterion for dropping packets is reaching a queue's maximum size. A scheme based on such a criterion tends to keep the queue in the full state for a relatively long period of time, which can cause severe network congestion in case of bursty traffic. This explains why queue management is often associated with congestion control.

Active queue management addresses the full queue problem by using a criterion more dynamic than the fixed maximal queue size. Random Early Detection (RED) [9] is an example of active queue management schemes. RED drops incoming packets probabilistically based on an estimated average queue size. The probability for dropping increases as the estimated average queue size grows. Specifically, RED uses two parameters to control the probability. One specifies the average queue size below which no packets are dropped; the other specifies the average queue size above which all packets are dropped. For a queue of average size between the two thresholds, the packet dropping probability is proportional to the average size. Naturally, the effectiveness of RED depends on how the relevant parameters are set. There is no single set of parameters that work well for all traffic types and congestion scenarios. Thus appear RED

To avoid the potential for excessive delays due to retransmissions after packet losses, explicit congestion notification (ECN) schemes have recently been developed. IETF RFC 3168 specifies an ECN scheme for IP and TCP.

variants, which introduce additional control to RED by, for example, providing differential drop treatment to flows based on their buffer usage or priority.

Congestion Avoidance

Congestion avoidance deals with means for keeping the load of the network under its capacity so that it can operate at an acceptable performance level. Traditionally, congestion is avoided by requiring that the sender reduce the amount of traffic entering the network when network congestion occurs (or is about to occur) [10]. (Ideally the source of the traffic reduction comes from a user whose admission control priority is not critical. This may permit higher-priority traffic to continue to receive normal service.) Unless there is an explicit indication, session packet loss or acknowledgment-timer expiration is normally regarded as an implicit indication of network congestion in an IP network. How the traffic source throttles back depends on the specifics of transport protocols. In a window-based protocol such as TCP, this is done by multiplicatively decreasing the size of the window. When congestion subsides, a sender then cautiously ramps up the traffic.

To avoid the potential for excessive delays due to retransmissions after packet losses, explicit congestion notification (ECN) schemes have been developed. IETF RFC 3168 specifies an ECN scheme for IP and TCP. With the scheme, incipient network congestion is indicated through marking packets rather than dropping them. Upon receipt of a congestion experiencing packet, an ECN-capable host responds essentially the same way as to a dropped packet.

Packet Marking

Packets can be marked according to specific service classes they will receive in the network on a per-packet basis. Typically performed by an edge node, packet marking involves assigning a value to a designated header field of a packet in a standard way. (For example, the type of service in the IP header or the EXP bits of the MPLS shim header is used to codify externally observable behaviors of routers in the DiffServ [11] or MPLS-DiffServ [2] approach.) If done by a host, the mark should be checked and may be changed (either promoted or demoted) by an edge node according to SLAs or local policies. Sometimes, special values may be used to mark non-conformant packets, which may be dropped later due to congestion.

Queuing and Scheduling

This mechanism deals with selection of packets for transmission on an outgoing link. The most basic queuing discipline is first-in first-out in which packets are placed into a single queue and served in the same order as they arrive in the queue. Under this discipline all packets are treated equally, and a sender can obtain more than a fair share of network bandwidth by simply transmitting packets excessively. To introduce flexible treatment of packets and fairness, various advanced queuing disciplines involving multiple queues come into existence.

Fair queuing: Packets are classified into flows and assigned to queues dedicated to respective flows. Queues are then serviced round-robin. Fair queuing is also called per-flow or flow-based queuing.

Priority queuing: Packets are first classified and then placed into different priority queues. Packets are scheduled from the head of a given queue only if all queues of higher priority are empty.

Weighted fair queuing: Packets are classified into flows and assigned to queues dedicated to respective flows. A queue is assigned a percentage of output bandwidth according to the bandwidth need of the corresponding flow. By distinguishing variable-length packets, this approach also prevents flows with larger packets from being allocated more bandwidth than those with smaller packets.

Class-based queuing: Packets are classified into various service classes and then assigned to queues dedicated to the respective service classes. Each queue can be assigned a different percentage of the output bandwidth and is serviced round-robin.

Traffic Classification

Traffic classification can be done at the flow or packet level. At the edge of the network, the entity responsible for traffic classification typically looks at *multi-fields* (i.e., a combination of header fields, including source address, destination address, source port number, destination port number, protocol number, and DiffServ code point) of a packet and determines the aggregate to which the packet belongs and the associated SLS. According to the SLS, classifiers steer packets to an appropriate traffic conditioning element for further processing.

Traffic Shaping

Traffic shaping deals with controlling the rate and volume of traffic entering the network. The entity responsible for traffic shaping buffers nonconformant packets until it brings the respective aggregate in compliance with the traffic. The resulted traffic thus is not as bursty as the original and is more predictable. Shaping often needs to be performed between the egress and ingress nodes.

There are two key methods for traffic shaping: leaky bucket and token bucket. The leaky bucket method regulates the rate of the traffic leaving a node. Regardless of the rate of the inflow, the leaky bucket keeps the outflow at a constant rate. Any excessive packets overflowing the bucket are discarded. Two parameters are characteristic to this method and usually user configurable: the size of the bucket and the transmission rate.

In contrast, the token bucket method is not as rigid in regulating the rate of the traffic leaving a node. It allows packets to go out as fast as they come in provided that there are enough *tokens*. Tokens are generated at a certain rate and deposited into the token bucket till it is full. At the expense of a token, a certain volume of traffic (i.e., a certain number of bytes) is allowed to leave the node. No packets can be transmitted if there are no tokens in the bucket, but multiple tokens can be consumed at once to allow bursts

to go through. This method, unlike the leaky bucket method, does not discard packets. Two parameters are characteristic to the token bucket method and usually user-configurable: the size of the token bucket and the rate of token generation.

The leaky and token bucket methods can be used together. In particular, traffic can be shaped first with the token bucket method and then the leaky bucket method to remove unwanted bursts. Two token buckets can also be used in tandem.

Management Plane Mechanisms

Metering

Metering concerns monitoring the temporal properties (e.g., rate) of a traffic stream against the agreed traffic profile. Depending on the conformance level, a meter can invoke necessary treatment (e.g., dropping or shaping) for the packet stream.

Service Level Agreement

A SLA typically represents the agreement between a customer and a provider of a service that specifies the level of availability, serviceability, performance, operation or other attributes of the service. It may include aspects such as pricing that are of business nature. The technical part of the agreement is the SLS [12], which specifically includes a set of parameters and their values that together define the service offered to a customer's traffic by a network. SLS parameters may be general, such as those defined in ITU-T Recommendation Y.1540, or technology-specific, such as the performance and traffic parameters used in IntServ or DiffServ.

Traffic Restoration

Restoration is broadly defined as the mitigating response from a network under conditions of failure. Potential methods for failure recovery include automatic protection switching for line or path protection and shared mesh restoration. There are two types of network failures:

Failure of an element (e.g., router card) in a network node or office. This type of failure is normally handled by designing redundancy features in network elements to minimize failure impact. Catastrophic failures such as power outages and natural disasters, however, may take down an entire network node. In this case, through traffic can be rerouted over spare links designed around the failed node.

Failure of a link connecting two network nodes. Typically links can fail due to link element failure (e.g., line card), which can take down a single link, or, more seriously, a fiber cut, which can disrupt a large number of links. Service providers can design additional spare capacity to mitigate the impact of such failures and restore traffic flows until the failure is repaired.

As in the case of admission control, certain traffic streams related to critical services may require higher restoration priority than others. A service provider needs to plan for adequate levels of spare resources such that QoS SLAs are in compliance under conditions of restoration. Common parameters for measuring service restorability are time to restore and the percentage of service restorability.

QoS Signaling

QoS signaling is mainly for conveying application (or network) performance requirements, reserving network resources across the network, or discovering QoS routes. Depending on whether the signaling information is part of the associated data traffic, QoS signaling may be effected in or out of band.

In band: The QoS signal is part of the associated data traffic, typically presented in a particular header field (e.g., the TOS field in IPv4 as in DiffServ and 802.1p) of the data packets. Taking place in the data plane, in-band signaling neither introduces additional traffic into the network nor incurs setup delay for the data traffic. Naturally such signaling is not suitable for resource reservation or QoS routing, which needs to be done a priori before data transmission.

Out of band: The QoS signal, being carried by dedicated packets, is separate from the associated data traffic. As a result, out-of-band signaling introduces extra traffic into the network and incurs an overhead for delivering desired network performance. In addition, it entails the use of a signaling protocol and further processing above the network layer, which tends to render slower responses than in-band signaling. Nevertheless, out-of-band signaling lends itself naturally to resource reservation or QoS routing.

Similarly, depending on whether the signaling path is closely tied to the associated data path, QoS signaling may be viewed as path-coupled or decoupled.

Path-coupled: QoS signaling messages are routed only through the nodes that are potentially on the data path. As such, in-band signaling by definition is path-coupled, but out-of-band signaling may or may not be. Path-coupled signaling implies that signaling nodes must be collocated with routers. Such an arrangement has on one hand the advantage of reduced overall signaling processing cost (since it leverages network-layer routing tasks), but on the other hand the disadvantage of inflexibility in upgrading routers or in integrating control entities (e.g., policy servers) not on the data path (or nontraditional routing methods). If a path-coupled mechanism involves a signaling protocol, routers need to support the protocol and be able to process related signaling messages. An example of a path-coupled signaling protocol is RSVP.

Path-decoupled: QoS signaling messages are routed through nodes that are not assumed to be on the data path. As such, only out-of-band signaling may be path-decoupled. (To date, most out-of-band QoS signaling schemes are path-coupled.) Path-decoupled signaling implies that signaling nodes should be dedicated and separate from routers. In contrast to path-coupled signaling, it has the advantage of flexibility in deploying and upgrading signaling nodes independent of routers or in integrating control entities not on the data path, but the disadvantage of added complexity and cost in overall processing and operational tasks.

There are standards efforts underway specifi-

Restoration is broadly defined as the mitigating response from a network under conditions of failure. Potential methods for failure recovery include automatic protection switching for line or path protection and shared mesh restoration.

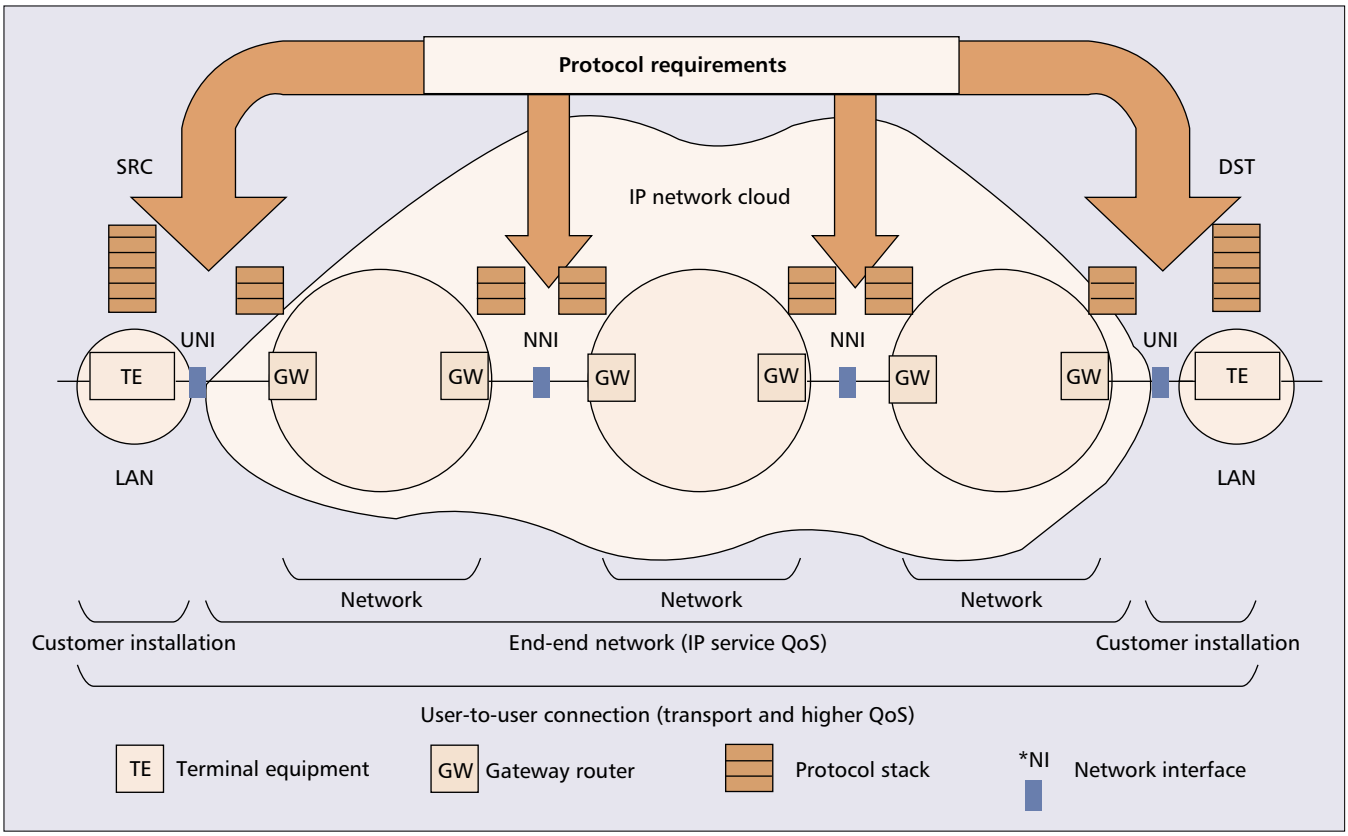

Figure 2. *Signaling requirements for IP QoS (ITU-T Draft TRQ.IPQoS.SIG.CS1).*

cally dealing with QoS signaling. In particular, the IETF *nsis* working group is developing a flexible signaling framework with path-coupled QoS signaling as its initial major application. A QoS signaling protocol defined under the framework is expected to address the limitations of RSVP. On path-decoupled signaling there seems not enough support in the IETF for a new project after some explorative discussion. Also worth noting is the ITU-T SG 11 effort presently defining requirements for end-to-end signaling of the IP QoS class, as defined in Y.1541, and reliability objectives across multiple administrative domains [13]. As shown in Fig. 2, the requirements will cover both the user–network interface and network–network interface. Depending on the final requirements, the effort may result in new QoS signaling mechanisms.

Conclusion and Future Work

ITU-T draft Recommendation Y.qosar identifies a preliminary set of generic network mechanisms that can be used to deliver required network performance. Organized according to three logical planes, the network mechanisms cover a wide range of functions. Most of them belong to the data plane, dealing with the user data traffic directly. The rest inhabit either the control plane, concerning the carriageways of the user data traffic but never the user data traffic itself, or the management plane, concerning the administration and management aspects of the user data traffic. Besides their functional differences as reflected in the logical planes where they reside, the mechanisms also differ in their invocation scope in terms of nodes or network segments. Some mechanisms are specific to a network node. Some mechanisms apply to a network. Still others need interactions with each other to render the desired effects. The latter two cases may dictate signaling between network nodes.

How signaling should be treated in the framework is an interesting question itself. Should it be considered a generic mechanism? If so, it definitely needs to exist in both the control and management planes. Whether it needs to be a distinct entity in the data plane, however, is unclear. On one hand, in-band signaling does not add much overhead to packet processing and thus does not seem to warrant a separate place in the data plane. On the other hand, if in-band signaling is not explicitly included, how can the signaling aspect of the data plane be conveyed? Our solution is to treat signaling as an auxiliary mechanism in support of the main ones. It does not show up explicitly in any of the logical planes. We give it its own section to underline its important and unique function.

Since the framework is still evolving, what we have described represents at best a snapshot of the work in progress as well as its direction. Obviously, further work remains to fill in the visible missing pieces such as policy, measurement, security, and interaction among the various basic mechanisms. The interaction piece is necessary for building a comprehensive QoS solution, especially for an environment including heterogeneous administrative or technology domains.

Standard methods (e.g., data objects or protocols) of enabling the interaction need to be identified. Finally, the framework should continue to take into account the results of other standards efforts on specific QoS network mechanisms to maintain consistency across QoS standards.

REFERENCES

[1] D. Awduche *et al.*, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, Dec. 2001.
[2] F. Le Faucheur *et al.*, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," IETF RFC 3270, May 2002.
[3] R. Yavatkar *et al.*, "A Framework for Policy-Based Admission Control," IETF RFC 2753, Jan. 2000.
[4] S. Chen and K. Nahrstedt, "An Overview of Quality-of-Service Routing for the Next-Generation High-Speed Networks: Problems and Solutions," *IEEE Network*, Special Issue on Transmission and Distribution of Digital Video, vol. 12, no. 6, Nov./Dec. 1998, pp. 64–79.
[5] E. Crawley *et al.*, "A Framework for QoS-based Routing in the Internet," IETF RFC 2386, Aug. 1998.
[6] D. Apostolopoulos et al., "Intra domain QoS Routing in IP Networks: A Feasibility and Cost Benefit Analysis," *IEEE Network*, vol. 13, No 5, Sept./Oct. 199, p. 42.
[7] D. Awduche, "Overview and Principles of Internet Traffic Engineering," IETF RFC 3272, May 2002.
[8] R. Branden *et al.*, "Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification," IETF RFC 2205, Sept. 1997.
[9] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net.*, V.1, N.4, Aug. 199, pp. 397–413.
[10] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM '88*, Aug., 198, pp. 314–29.
[11] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF RFC 2474, Dec. 1998.
[12] A. Westerinen *et al.*, "Terminology for Policy-Based Management," IETF RFC 3198, Nov. 2001.
[13] ITU-T Study Group 11 draft TRQ.IPQoS.SIG.CS1, "Signaling Requirements for IP-QoS," Nov. 2002.

BIOGRAPHIES

HUI-LAN LU (huilanlu@lucent.com) is Bell Labs Fellow and consulting member of technical staff at Bell Laboratories-Lucent Technologies, where she is responsible for multimedia standards and their applications to research and development. She joined Bell Labs in 1990 after receiving her Ph.D. degree in physics from Yale University, New Haven. She is active in the IETF and ITU-T, and has published extensively in the areas of Internet-PSTN interworking and next-generation network architectures. She is Rapporteur for Q.16 in ITU-T SG 13, responsible for developing the new draft Recommendation Y.qosar.

IGOR FAYNBERG is senior manager, standards and technologies at Bell Laboratories/Lucent Technologies and Adjunct Professor of Computer and Information Science, Stevens Institute of Technologies. He is active in the IETF, where he was a founding chair of the PINT working group, and ITU-T, where he serves as Rapporteur in SG 13. Frequently invited to speak at various conferences, he has numerous publications, including two books, *Intelligent Network Standards, Their Applications to Service* and *Converged Networks and Services: Internetworking IP with PSTN*. He holds a Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia.

Since the framework is still evolving, what we have described represents at best a snapshot of the work in progress as well as its direction. Obviously further work remains to fill in the visible missing pieces.