FIRST QUARTER 2005, VOLUME 7, NO.1

IEEE COMMUNICATIONS SURVEYS

The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

ON THE BUILDING BLOCKS OF QUALITY OF SERVICE IN HETEROGENEOUS IP NETWORKS

JOHN SOLDATOS, ATHENS INFORMATION TECHNOLOGY
EVANGELOS VAYIAS, INTRACOM SA
GEORGE KORMENTZAS, UNIVERSITY OF THE AEGEAN

ABSTRACT

After more than a decade of active research on Quality of Service in IP networks and the Internet, the majority of IP traffic relies on the conventional best-effort IP service model. Nevertheless, some QoS mechanisms are deployed in current networking infrastructures, while emerging applications pose QoS challenges. This survey brings into the foreground a broad range of research results on Quality of Service in IP-based networks. First, a justification of the need for QoS is provided, along with challenges stemming from the convergence of IP and wireless networks and the proliferation of QoS-demanding IP applications (such as VoIP). It is also emphasized that a global uniform end-to-end IP QoS solution is not realistic. Based on this remark, packet-level QoS mechanisms are classified as certain building blocks, each one fulfilling different objectives in certain parts of a heterogeneous IP network. This taxonomy, being in line with the ITU-T initiative toward a QoS architectural framework for IP networks, gives rise to a thorough presentation of QoS "building blocks," as well as their associated mechanisms. This presentation is followed by an illustration of how the various building blocks are combined in the scope of modern IP networks. However, offering QoS in a large scale IP-based network demands that additional (i.e. non-packet-level) QoS mechanisms are deployed in some parts. Therefore, we also present prominent technologies and mechanisms devised to augment the QoS capabilities of access, wireless, and optical networks. We illustrate how these mechanisms boost end-to-end QoS solutions and reveal interworking issues with packet-level mechanisms.

Quality of Service (QoS) in IP-based networks and the Internet has been a vision for the Internet research and engineering community for more than a decade. A large number of architectures, technologies, and mechanisms enabling IP Quality of Service have been devised toward enhancing the conventional best-effort IP service model (e.g., IETF RFC 1633, IETF RFC 2475, [1, 2]). Some of these mechanisms are already deployed in broadband IP networks. For instance, policing, scheduling, and queue management are occasionally applied toward improving network performance. Also, several IP network operators rely on marking the ToS (Type of Service) byte of IP packets toward supporting more than one class of service. Moreover, Multiprotocol Label Switching (MPLS) (IETF RFC3031) and its associated traffic engineering technologies have considerable penetration in service provider infrastructures [3].

Nevertheless, large portions of IP networks still do not employ any QoS mechanisms. The majority of service providers over-provision their backbones toward providing quality networking services to their customers [4]. Over-provisioning the IP backbones is still a very low-cost option, given that recent advances in optical transmission technology have increased the per-fiber available bandwidth. Moreover, congestion problems in large ISPs are usually affecting a small part of the network and can be simply alleviated by adding more bandwidth to that small part, rather than deploying network-wide traffic control mechanisms.

Over-provisioning presents inherent disadvantages, basically because it does not automatically ensure the necessary QoS, due to the best-effort handling of data. Delay jitter cannot be controlled since IP traffic is characterized by frequent generation of short-lived flows and burstiness at various timescales. These factors frequently lead to poor application performance. Therefore, QoS research targeting IP networks still remains in the foreground. The interest in IP QoS is expected to gradually increase due to the proliferation of mission-critical IP applications (e.g., Voice-over-IP, Internet telephony, tele-conferencing, virtual private networks, e-commerce), which require stringent QoS guarantees in terms of network latency and delay jitter. The most prominent of these applications is VoIP (Voice-over-IP), which enterprises see as having huge potential for cost savings.

Additional IP QoS challenges are emerging as a result of the advent of Grid Networking infrastructures. Optimizing IP infrastructures for Grid computing demands that network QoS schemes are re-examined in the scope of Grid-enabled applications and middleware. Furthermore, there is a need to associate QoS with middleware services to allow for a dynamic exploitation of available networking services (e.g., [5]).

Apart from posing additional challenges, the evolution of IP networks and applications has resulted in a pressing demand for expanding QoS research to other areas. The IP protocol serves nowadays as a ubiquitous internetworking mechanism bridging heterogeneous network segments. The most prominent example is the convergence between wireless networks and the Internet, which raises issues relating to delivering QoS to roaming or wireless users accessing an IP network. Also, since most backbones are gradually migrating to all-optical solutions, QoS mechanisms need to take into account the evolution of the respective networking technologies, such as Generalized MPLS (GMPLS).

As a result, current IP-based infrastructures are likely to comprise several heterogeneous segments (e.g., access, wireless, optical), where packet-level QoS mechanisms are not sufficient. Toward delivering QoS to end users, additional QoS mechanisms are researched and standardized for these segments. Most of these additional mechanisms are not controlling traffic at the packet level, and therefore differ from IP QoS schemes. In several cases, these QoS mechanisms have to deal with different challenges as well. For example, QoS in wireless/mobile environments is harder to achieve, since resources (e.g., bandwidth) are much more limited and thus over-provisioning is prohibitive. Wireless networks are associated with fundamental limitations in power, available spectrum, over-provisioning cost, and mobility, and therefore tend to have less predictable availability, less connection stability, less bandwidth, and more latency. However, as wireless networks proliferate, there is a great demand for providing wireless QoS levels. In this context, there are considerable efforts to bring QoS into various wireless technologies. Indicative examples of these efforts are:

- the IEEE 802.11e standard, which defines service differentiation mechanisms in order to guarantee access delay bounds to WLAN stations,
- the introduction of service bearers for QoS provision in UMTS settings.

Moreover, the standard Rev. A of cdma2000 1xEv-Do offers high data rates (i.e., 3.1 Mb/s/1.8 Mb/s downlink/uplink data rates) and QoS support through selectable levels of priority and latency.

Contrary to wireless networks, where over-provisioning is prohibitive, pure optical solutions provide immense capacity, and the challenge there is to automate the provisioning and management of lightpaths. Therefore, end-to-end QoS in large-scale IP-based networks demands that additional non-packet-level QoS mechanisms are provided along with their interworking with packet-level mechanisms.

Having justified the value of QoS mechanisms and identified the need for further research, this survey reviews and discusses results relating to QoS mechanisms and develops a new focus toward advancing this research. The primary focus is on surveying QoS mechanisms operating at the packet level. Most of these mechanisms have their roots in conventional QoS frameworks such as B-ISDN, IntServ, and DiffServ. We adopt an approach that is perfectly aligned to problems encountered in most state-of-the-art networking infrastructures. Most of these infrastructures are highly heterogeneous and tend to serve different goals depending on the business objectives of the network operator or service provider. In such heterogeneous environments, it is not realistic to seek a uniform global framework for end-to-end QoS, which was more or less the vision of B-ISDN, ATM, and the IntServ. Rather, it seems more pragmatic to focus on mechanisms that deal with specific problems of particular technologies (e.g., scalable per-flow scheduling, traffic engineering, etc.), along with interworking issues (e.g., QoS interworking between different domains). In the scope of this article, we refer to these mechanisms as QoS "*building blocks*." This approach is in line with state of the art IP networks (i.e., which span several segments and particular technologies), as well as with emerging needs.

The idea of exploring QoS mechanisms based on the notion of building blocks bears similarity to recent ITU-T Study Group 13 initiatives aiming at providing an architectural framework for QoS support in packet networks, with a focus on IP [6]. This framework provides a taxonomy of the building blocks and an initial survey of algorithms for their realization. Moreover, it supports integration of standards efforts addressing specific QoS network mechanisms. The present article complements these efforts through a thorough survey of QoS research results and their applicability to various QoS provisioning contexts. Thus, the article bridges building blocks with research results produced over more than a decade. Apart from reviewing results and associating them with QoS building blocks, a rationale on the applicability of these results is provided. In particular, the article suggests how building blocks can be combined toward meeting the variety of objectives that are usually set for an operational network. We strongly believe that insight into these building blocks is a key prerequisite for dealing with network performance and optimization problems. Engineers should be aware of the blocks/tools at hand for building the networks, while researchers should be able to identify where further work is needed.

We also briefly review non-packet-level QoS mechanisms specified in the scope of access, mobile, and all-optical networks. Acknowledging the fact that IP QoS building blocks cannot provide end-to-end QoS in a heterogeneous IP-based network (such as the Internet), we discuss QoS mechanisms devised for other network segments such as 3G/4G networks, access networks, and optical networks with a control plane (such as Automatically Switched Optical Networks — ASONs— based on GMPLS). Apart from presenting QoS support in these networks, we also discuss mechanisms required for their interworking with the IP QoS building blocks.

The rest of this article has the following structure. We review the most prominent QoS frameworks for IP networks and identify the main building blocks comprising these frameworks. Having identified building blocks, we focus on algorithms devised for these blocks. In particular, QoS research results relating to the various building blocks are surveyed and future directions are highlighted. We illustrate the interworking of various building blocks and illustrate their synergy toward tackling particular QoS problems and operational objectives. We complement the discussion of packet-level QoS mechanisms with a brief review of QoS results addressing access, optical, and 3G/4G wireless network technologies. Along with QoS issues, internetworking schemes are discussed toward QoS solutions spanning several network segments. Finally, we conclude the article.

OVERVIEW OF QOS FRAMEWORKS

The concept of QoS in multi-service packet-switched networks was first emphasized during the efforts for a B-ISDN framework. ATM, the basis of the B-ISDN, could combine real-time and non-real-time media streams in a fast packet switching transfer mode [7]. Since the performance of packet-switching is statistical, several traffic management mechanisms had to be engineered in order to guarantee the Quality of Service required, especially for the real-time streams.

The QoS framework of ATM was chiefly built upon its connection-oriented nature. When a new Virtual Channel connection is to be established through the network, the process of connection admission control (CAC) takes place. CAC is a set of actions that either accepts or rejects the newly requested connection depending on the status of network resources; the connection is accepted only if the network has sufficient resources to guarantee the QoS of the new connection, while maintaining the QoS agreed for the already established connections. To enable CAC to make reliable acceptance/denial decisions, a "traffic contract" has to be negotiated between the "user" and the "network" which consists of:

- Specific limits on the profile of the traffic that the connection will offer to the network in terms of well chosen traffic descriptors (e.g., peak cell rate, sustainable cell rate, maximum burst size, etc.)
- An ATM service class that is required by the traffic stream and the values of the associated QoS measures: delay, jitter, and loss.

The significance of the traffic contract is that the network will guarantee that the connection will receive the QoS denoted by the agreed class, as long as the traffic offered conforms to the agreed traffic descriptors. In case the connection violates the agreed traffic contract, it is possible that not only its QoS will be affected, but the QoS of other connections as well. To avoid such situations, the network needs a way to "police" the offered traffic in order to monitor or even enforce that it conforms to the agreed contract. Policing of traffic in ATM is termed usage parameter control (at the network access) or network parameter control (in the network core) and is based on the dual leaky bucket algorithm. The user can also monitor or enforce the conformance of the traffic offered by shaping the traffic. Shaping is a function similar to policing, their difference being that policing drops or tags cells/packets non-conforming to the traffic contract, while shaping delays their transmission so that they become conforming.

The traffic management framework currently in effect in ATM networks has been specified in the ATM Forum's Traffic Management 4.0 specification and in ITU-T Recommendation I.356.

Apart from the CAC, policing and shaping functions, which are the foundations of the ATM QoS framework, other functions were also required in ATM networks in order to support QoS. One such important function is the calculation of network resources to be allocated to a connection. Several resource allocation methods have been proposed (e.g., [8–10]). Another function relates to the selection of a routing path for the connection through the network. Other functions have to take place within ATM switches at the cell level related to the scheduling of cells for transmission. Finally, combined cell-based and frame-based queue management functions such as early packet discard (EPD) and partial packet discard (PPD) [11] were devised in order to boost frame-level throughput across ATM networks.

ATM's QoS framework targeted what is sometimes called "hard QoS guarantees," i.e. a set of service classes were defined, for which QoS is specified in terms of quantitative measures such as the cell transfer delay, the cell delay variation, and the cell loss ratio; the network has to guarantee specific bounds for some or all of these quantities depending on a connection's service class. Also, network elements are explicitly instructed to reserve and allocate resources for a connection; these resources are withheld for serving the connection, while it remains established and are only released when it is torn down. This model is well suited for a connection-oriented protocol such as ATM, since network elements have to maintain per-connection state anyway. Obviously, it is not efficient in a connectionless IP environment, where per-flow state is not an inherent requirement. In general, it is still in effect today in the scope of ATM's current field of application (xDSL, 3G backbones, migration from ATM to IP-based multi-service backbones, etc.).

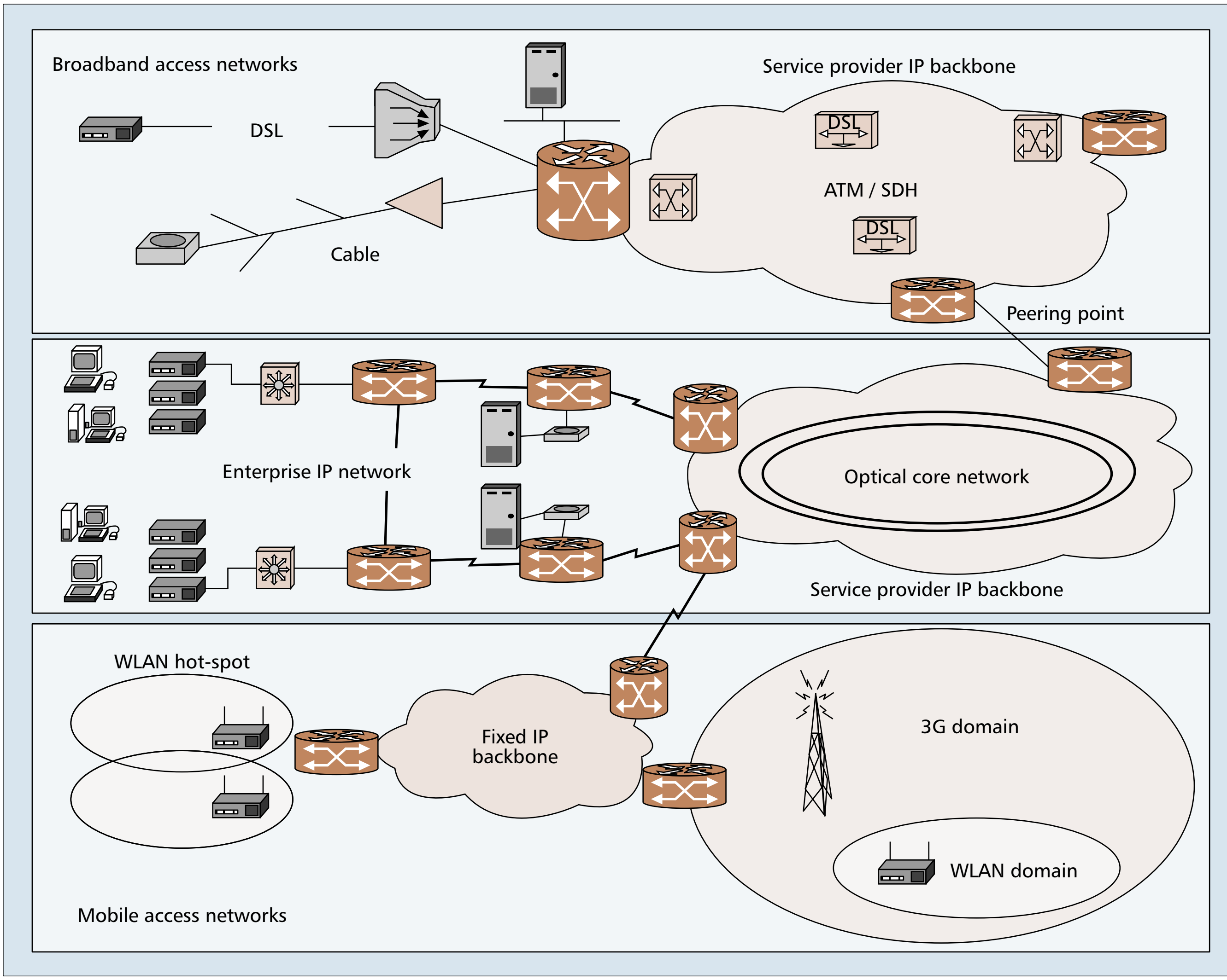
Nevertheless, the building blocks introduced by ATM's traffic management framework were also at the basis of the first proposals for devising QoS architectures for the Internet and IP-based networks. Specifically, admission control, policing, shaping, resource reservation and allocation, routing, and scheduling are at the heart of the Integrated Services (IntServ) framework (IETF RFC 1633), which is discussed in the sequel.

Since IP networks are connectionless, the notion of "flows" was introduced. A flow is a sequence of IP packets having the same source and destination IP addresses and protocol ports. Related research focused on mechanisms for per-flow packet classification and scheduling [12, 13] with a view to bounding the queuing delay and/or control congestion loss. Classification categorizes packets depending on the end-to-end flow to which they belong, while scheduling ensures that packets of each flow get the network resources required to achieve the QoS desired for the flow. A mechanism was needed to specify and reserve resources for a flow. The reservation mechanism for IntServ was the RSVP protocol (IETF RFC 2205). Using this protocol, an end system can specify the resources required for a flow using a token bucket specification (the equivalent of an ATM traffic descriptor), as well as its QoS service class and associated parameters.

RSVP works in conjunction with unicast and multicast IP routing protocols to reserve resources across the path that a flow's packets will take. Since a flow's path across an IP network is not pinned throughout its lifetime, RSVP messages are periodically transmitted to ensure that the reservation remains up-to-date across the current routing path. Upon receiving a RSVP message a network element has to make an admission control decision depending on whether it has the resources requested. Also, in order to enforce (or monitor the conformance to) the specified traffic profile, shaping (policing) can be employed by the end system (network element).

Unlike the ATM QoS architecture, the IntServ specification itself defines only the framework for supporting QoS and is not tied to any particular service class. Two service class specifications accompanied the IntServ standard: the "guaranteed service" specification (IETF RFC 2212) defined a service class similar to ATM's "hard QoS" service classes offering tight quantitative bounds on delay and loss; the "controlled load service" (IETF RFC 2211) defined a more relaxed service class offering qualitative guarantees. Another key difference with ATM is that IntServ, and RSVP in particular, use a "soft-state" mode to maintain resource reservations in network elements, i.e. resources are released unless their reservation is refreshed. This was needed in order to cope with possible route changes during the lifetime of a flow.

A concern about the IntServ framework was that it



■ **Figure 1.** *Example architectures and underlying technologies of current IP networks.*

required that all network elements maintain per-flow state and support RSVP, admission control, packet classification based on multiple fields of the IP header (MF), and scheduling. This concern, partly justified by the processing capabilities of network elements at the early period of IntServ standardization, led to the definition of the Differentiated Services (DiffServ) framework (IETF RFC 2475). In DiffServ, only the routers at the ingress or egress of a networking domain (termed "edge routers") are required to perform MF classification; packets there will be "marked" so that their ToS byte, termed DiffServ code point (DSCP) in the DiffServ framework, indicates their flow's QoS service class (termed per hop behavior). Internal (or "core") routers are only required to classify packets based on their DSCP and apply the scheduling discipline required in order to satisfy the QoS corresponding to each PHB. DiffServ does not define how resource reservation and configuration of traffic management mechanisms takes place. It is up to the network operators to decide if, where, and how they will apply resource reservation, admission control, policing, and shaping mechanisms.

IntServ, Diffserv, and ATM's QoS framework are the best known cases of standardized QoS frameworks for packet networks. However, it seems that such frameworks are not usually adopted in their entirety, in order to serve the QoS requirements of current networking architectures, which feature an increasing degree of diversity and heterogeneity. In Fig. 1, we illustrate some examples of the architecture and of the underlying technologies that can be currently encountered in IP networks. These may comprise a variety of backbone infrastructures ranging from SDH (Synchronous Digital Hierarchy) and ATM to optical-switched networks that interconnect various access segments such as LANs, wireless/mobile, and broadband networks. Note that the underlying technologies are likely to have diverse characteristics in terms of transmission modes (broadcast, serial, synchronous, asynchronous) and of available capacity.

Both IntServ and DiffServ operate solely at the packet level, free from the implications of lower-layer protocols. However, IntServ's deployment has been hindered by concerns about its scalability and its slow adoption by end-systems and applications, while DiffServ leaves a lot of open issues on how network resources are reserved and allocated.

Even though deployment of QoS-enabled networks does not adopt any of the above frameworks in its entirety or exactly as it was specified, their principles have greatly influenced network design. In particular, the traffic management functions defined by each of these frameworks constitute essential "building blocks" for achieving QoS at the packet level. These blocks can be used by network engineers in order to design a QoS-enabled network fitting their particular requirements.

Figure 2 illustrates these basic building blocks, their functional placement within a network, and the time scale in which they take place. Thus, **shaping**, **policing**, **classification and marking**, **scheduling** and **queue management** operate on a per-packet time scale and reside in a network node; shaping, however, may as well reside in an end-system. **Resource reservation**, **signaling** and **admission control** are invoked upon flow establishment and tear-down and involve interaction among network nodes or between an end-system and a network node. **Routing** and **resource management** may also be triggered by flow establishment and tear-down, as well as by topology changes or on a periodic time scale.

Another important traffic management function, not specified within the above QoS frameworks, is end-to-end **congestion control**. The goal of this function is to try to adapt the rate at which an end-system generates traffic to the varying load conditions in the network. It generally makes use of feedback signals sent to the opposite direction of the traffic flow.

Lately, it has also become apparent that QoS provisioning is not merely a technical matter but also a matter of organizational policy; an organization that operates a network needs a method to control the level and granularity of QoS offered to network users according to internal policy criteria. Thus, QoS **policy management** in conjunction with an organization's directory services has become a hot issue and constitutes another important QoS building block.

QoS pricing is also directly associated with network services that are essentially differentiated from conventional best-effort services. Pricing constitutes a vehicle for maximizing the revenue of the service provider or the network operator. It is also a vehicle for discouraging several users from employing quality services, thus optimizing network utilization. Although not always directly associated with packet-level QoS control, **QoS pricing** is another important component of QoS-enabled networks (i.e., another building block). Observe that pricing is not shown in Fig. 2.

Intensive research efforts have taken place during the last two decades in order to devise efficient mechanisms for most of these building blocks. In the following section, we give insights into the main results of these efforts. Since some of the building blocks are similar in nature (e.g., policing and shaping) or their performance is intertwined (e.g., congestion control and queue management), we have chosen to give a joint presentation of the related research results; in such cases, a research work most probably concerns both building blocks. Therefore, we will survey the building blocks in the following order and grouping.

- Admission control
- Shaping and policing
- Signaling and resource management
- Queuing and scheduling
- Congestion control and queue management
- QoS routing
- QoS policy management
- QoS pricing

We precede the survey of the building blocks by a brief overview of the main traffic modeling results for packet-switched networks. Although not a building block, traffic modeling influences QoS research since it tries to express the properties of network traffic in an analytic manner.

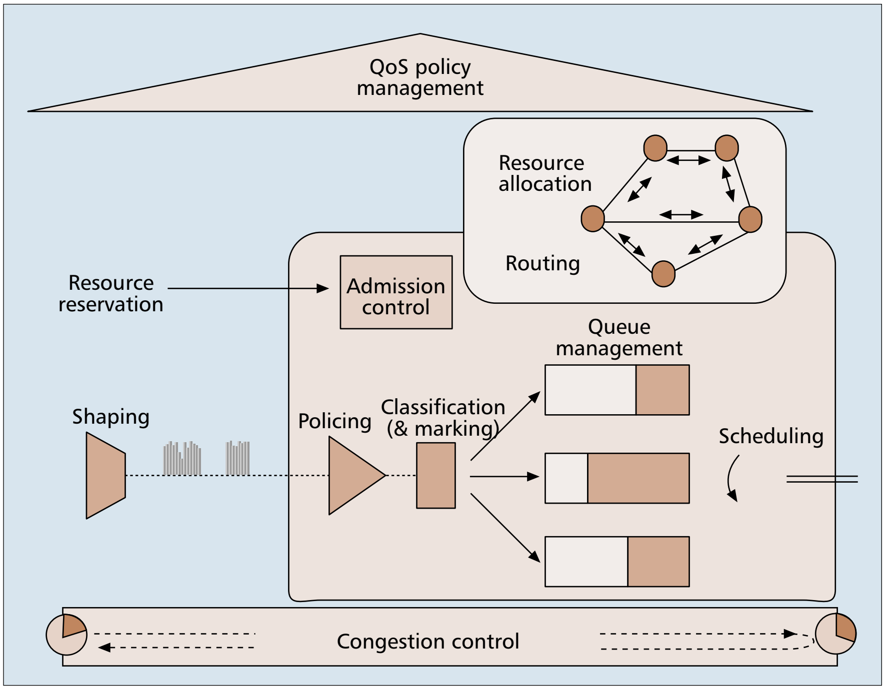
Insights on the Building Blocks of QoS

Traffic Modeling

Traffic modeling is not a building block per se, yet it is essential to designing and engineering the majority of the QoS building blocks. A traffic model is a set of rules (most usually mathematical) governing packet generation. Based on such mathematical models, researchers and engineers attempt to explain traffic performance relations linking bandwidth, demand, and performance. These relations can then serve as a basis for applying traffic control functions, as well as for estimating network performance. Given that traffic demand is statistical, performance is likely to be expressed in a probabilistic manner.

Traffic modeling research for packet networks focused initially on packet-level models. Packet-level models were expressed in the form of stochastic processes modeling packet inter-arrival times, as well as the packet sizes. Based on the assumption that traffic was stationary, several stationary models were devised using known stochastic processes (e.g., Poisson, Bernoulli, Markov Modulated Poisson Processes) each one concentrating on a particular application/service (i.e. video, voice, data services) [14]. Apart from packet-level models, stationary models at the burst level (i.e. considering batch transmission of groups of packets), as well as fluid flow models (e.g., [10]) transferring the modeling from the discrete to the continuous space, were devised. Burst-level and fluid flow stationary models are in general less accurate than packet-level models, leading, however, to simpler and more tractable control problems.

While traffic modeling using conventional stochastic processes (i.e., short-range traffic models) has the distinct advantage of allowing well known calculus to be applied to traffic control problems, extensive traffic studies have demonstrated that the actual traffic does not fulfill the stationary assumption. Moreover, traffic experiments have confirmed that packet traffic exposes strong non-stationary, uncertainty, and



■ **Figure 2.** *QoS building blocks.*

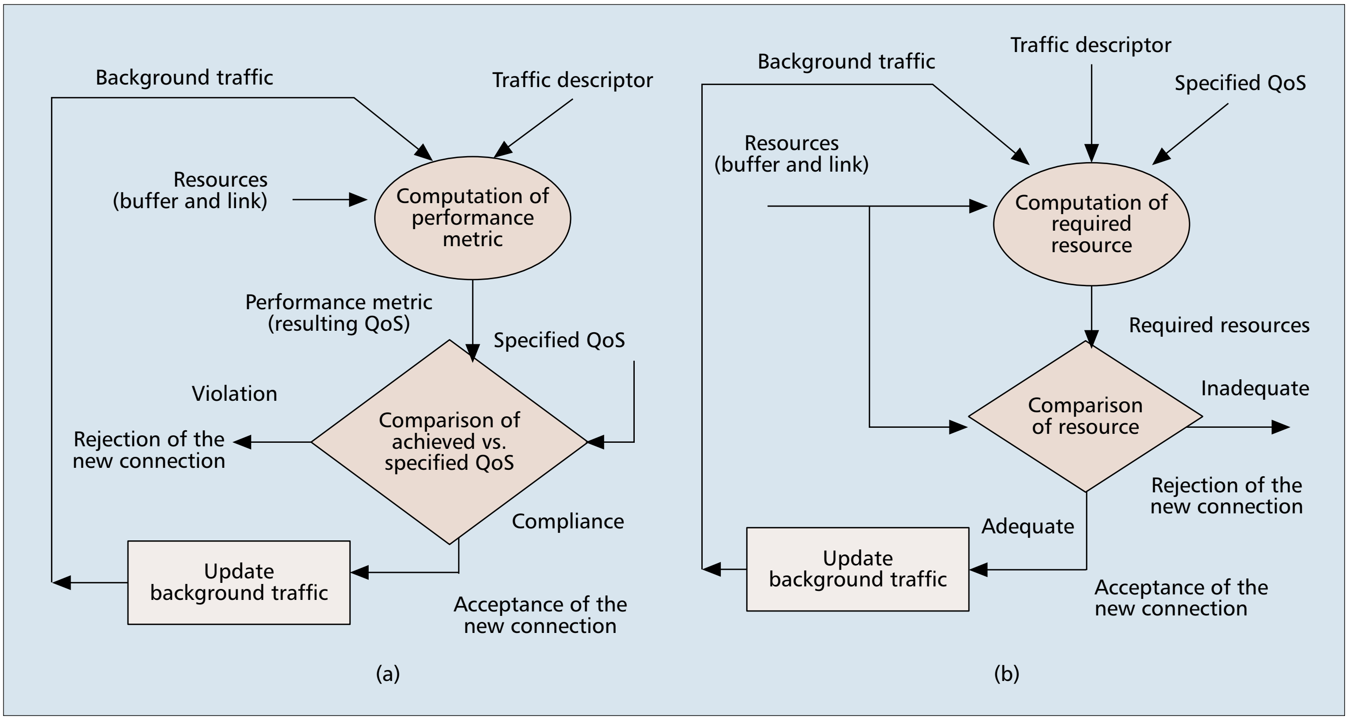


Figure 3. *Generic CAC procedure: a) based on a performance metric; b) based on an acceptance region boundary.*

nonlinear properties. In particular, both LAN/Ethernet [15] and WAN/Internet backbone [16] traffic expose long-range dependence and self-similarity. Self-similar and long-range dependent models are more complex to handle and therefore make traffic control problems more complex. The practical effect of self-similarity is that traffic exposes burstiness at all timescales. As a result, the buffering capacity at traffic multiplexing points (e.g., routers, switches) must be much larger than that derived by traditional queuing analysis and simulations. Large buffers alleviate packet losses at the expense of increasing delay. This delay increase can have a negative impact on the performance of real-time and/or streaming applications. Therefore, a bufferless multiplexing model is preferable for real-time and/or streaming flows. On the other hand, simple processor sharing models are probably better for handling elastic flows, since they could render average throughput performance largely insensitive to the detailed traffic characteristics of the flows [17].

From a pure research perspective self-similarity complicates the task of modeling aggregate traffic, as well as the network overall [18]. This complexity is primarily manifested in the Internet environment, due to additional challenges stemming from the scale, heterogeneity, and dynamics of this giant network. However, there are still cases where short-range dependent models can be applied, for example in the scope of un-congested backbone links [19]. Generally, the suitability of a conventional Poisson model for IP traffic depends heavily on the level of aggregation in the networks [20]. Moreover, traffic patterns in the scope of LANs may be significantly different from patterns at the core networks.

Regardless of the dynamics of the traffic, traffic shaping (illustrated in the sequel) can be employed to control the traffic profile at certain points of the network. As a characteristic example, in [21] we showed that packet shaping based on non-linear spacing laws can result in non self-similar aggregate traffic, giving rise to the application of conventional queuing theory. This kind of traffic shaping has been proven to be applicable in the Internet, provided that shaping parameters are carefully selected to account for negligible spacing delay.

Admission Control

Admission control determines whether a new traffic flow can be admitted to the network without violating the QoS enjoyed by already established traffic flows. As a result, admission control ensures that the network resources are sufficient to accommodate new traffic. Admission control concepts have their roots in ATM technology [22]. However, the notion was soon extended to IntServ and DiffServ networks and is nowadays considered an integral building block of QoS.

Admission control is a traffic control function applied at a flow time scale. It is a function associated with each network element (e.g., switch, router): establishing a traffic flow along a path of network elements requires that the flow is accepted from each one of the elements that comprise the path. The most crucial component of admission control is the algorithm employed to take the admission decision. As illustrated in Fig. 3, the admission control algorithm accepts as input information about existing traffic flows (i.e., traffic models, QoS requirements) and available network resources (i.e., link/path bandwidth, output port buffer), as well as information on the incoming traffic flow and its target QoS, and returns a *boolean* result relating to the acceptance or rejection of the new flow. As also shown in Fig. 3, the acceptance or rejection decision can be taken based on availability of resources to accommodate the target QoS, or alternatively based on whether the new flow violates the QoS of existing flows. Apart from ensuring QoS, an efficient admission control algorithm should provide a high degree of accuracy, thus achieving a high statistical multiplexing gain and an overall optimal utilization of network resources. Nevertheless, it should also be easily tractable, so as to be applied in real time.

The generic CAC process illustrated in Fig. 3 calculates all traffic parameters relating to the admission decision based on information about background traffic, available resources, as well as the traffic characteristic of the incoming flow. There is, however, another important class of admission control algorithms, which do not attempt an analytical calculation and estimation of all admission control parameters. Rather, they estimate (on-line, real-time) parameters that are essential to the admission control process. This class of algorithms is called "*measurement based algorithms*." Depending on the scheme, the parameters to be measured relate to the traffic model of the source (i.e., toward estimating parameters), or performance parameters such as the packet loss ratio. Measurement-based schemes have gained momentum given the difficulties in declaring traffic descriptors, the rapid emergence of new applications, as well as the potential traffic distortion as traffic travels through the network [23].

SHAPING AND POLICING

Closely related to admission control functionality are two other building blocks: shaping and policing. Traffic shaping refers to controlling the rate of outgoing traffic toward enforcing a particular traffic profile (e.g., [24]). Shaping is important in enforcing a particular traffic model at traffic sources or traffic multiplexing points, especially in cases of bursty traffic. In such cases, shaping retains excess packets in a queue and then schedules them for later transmission over increments of time (i.e., packet spacing is performed). Shaping is associated with the existence of a queue and of sufficient memory to buffer delayed packets. As a result, it is an outbound concept applied to packets going out of an interface. Moreover, it is usually associated with a scheduling function toward transmitting delayed packets. Packet shaping is a salient function for traffic flows crossing several multiplexing stages, since it provides a method to maintain their original traffic profile. This is accomplished through (re)shaping traffic at multiplexing points toward eliminating deviations from the original traffic profile resulting from variable delay experienced at multiplexing points.

Traffic policing monitors traffic entering the network to ensure that it remains compliant to a predefined profile. Traffic policing drops (or marks) packets whenever the offered traffic goes out of the agreed profile (e.g., bursts above a configured maximum rate for more than an allowed burst interval). Policing is usually applied to inbound traffic of an interface. Note also that policing is commonly applied at ingress points, on traffic that has been previously shaped to conform to a particular profile. Therefore, it is considered as a dual function of traffic shaping.

Both functions feature a high penetration in the current networking infrastructures. The network engineer should choose to implement shaping and/or policing based on the objectives of the network design, as well as on their relative advantages and disadvantages. Shaping does not drop packets unless excess traffic is sustained at high rates, resulting in an overflow of the shaper's buffer. Therefore, it is appropriate for avoiding retransmissions due to dropped packets. However, it may introduce undesirable delays for delay-sensitive traffic. On the other hand, policing can also control the output rate through packet drops. As a result, it is mostly appropriate for avoiding queuing delays (e.g., in the scope of real-time applications).

SIGNALING AND RESOURCE MANAGEMENT

Signaling typically concerns network (re)configuration requested by users and achieved within a short time interval (milliseconds or seconds). When the reaction time for (re)configuration becomes measured in minutes or hours, it is often referred to as resource management, while even larger (re)configuration times constitute the notion of network provisioning. In all cases, the (re)configuring action involves establishing (or modifying) information used by routers or switches to control their forwarding actions, including forwarding (routing) information, classification rules, and/or queuing and scheduling parameters. Without the discussed (re)configuration practices, the networking elements (routers and switches) follow standardized behavior (e.g., FIFO best-effort forwarding) that is explicitly or implicitly defined by implementation agreements or specifications.

Today's Internet routing protocols, such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP), represent a form of free-running signaling, where the signaling topology changes and the respective routing information is forwarded among the routers under their care. Protocols such as Resource Reservation Protocol (RSVP) were developed expressly for the purpose of signaling additional QoS information along existing paths and associating it with specific classes of traffic. In the absence of RSVP-signaled QoS parameters, routers apply only provisioned or standardized CQS (Classify, Queue and Schedule) rules.

RSVP provides QoS guarantees by enabling applications to make requests to reserve ahead of time network resources for their explicit use. The network would respond to the reservation requests by explicitly admitting or rejecting the request. Once a reservation path for a request has been established, this reservation must be periodically refreshed through appropriate RSVP signaling messages for the whole duration of the transmission. The RSVP process provides a hard guarantee of whether an application request can be served or not. It also provides guarantees over the quality level of the service, meaning that if a reservation request is accepted then the application will receive the requested QoS.

RSVP constitutes the prevalent signaling protocol for the IntServ QoS framework. IntServ defines the models for expressing service types, for quantifying resource requirements, and for determining the availability of the requested resources at relevant network elements. In the RSVP/IntServ architecture, RSVP is responsible for signaling per-flow resource requirements to network elements using IntServ parameters. In turn, these network elements would then apply IntServ admission control to the reservation requests. Note that in the case of the reserved resources, these are only used by the requester and are isolated from other traffic. Generally speaking, the RSVP/IntServ architecture aims at transcending traditional protocol boundaries by defining a universal set of rules for enabling QoS to take place on any network topology, although as of yet it has only been clearly defined for IP-based networks.

A key lesson learned from RSVP is that in order for a QoS signaling protocol to be successful it must not add overhead burdens to the underlying networking elements. This requirement is addressed in the Diffserv QoS framework, which "flags" each packet that travels the network with a traffic class identifier, namely the differentiated services code point (DSCP). The network can then apply forwarding rules specific to that traffic class. In this context, service level agreements (SLAs) concern aggregate traffic of the same class and not individual flows.

Diffserv is promoted as a scalable QoS solution. However, despite the simple concept of packet labeling, the DiffServ inner workings are quite complex, featuring a series of traffic conditioners, shapers, and markers governing the process of differentiating traffic classes. A thorough understanding of this form of traffic engineering is essential for the purpose of designing seamless internetwork protocol mappings.

Another important issue regarding Diffserv concerns the resource management strategy in each underlying networking node (RSVP solves this issue for Intserv). This strategy constitutes the major task of bandwidth brokers (BBs) [24]. In general, a BB undertakes the following tasks: the negotiation of

SLAs with BBs of neighboring domains; the translation of SLAs into one or several traffic control agreements (TCAs) for edge devices; and the delivery of the TCAs to the edge routers of the administered domain.

QUEUING AND SCHEDULING

In packet-switched networks, packets contend for access to an outgoing transmission link, since the instantaneous rate at which packets arrive for transmission to that link may exceed the link's capacity. This is why packet-switching nodes use buffers, where packets arriving at a rate greater than the link capacity are queued and wait for transmission.

Packet queuing and scheduling is the mechanism that dictates which of the packets waiting in a link's buffer will be selected for transmission. Obviously, the simplest scheduling mechanism (or "queuing discipline") is first-in-first-out (FIFO) queuing, in which the packets are queued in a single queue in the order that they arrive and are transmitted in that order. This simple discipline guarantees ordering of packets belonging to the same flow, but it suffers from two weaknesses: it permits misbehaving senders to exhaust network resources, and it does not permit differentiation of performance levels.

In order to alleviate the above weaknesses of FIFO queuing, other scheduling disciplines have been developed.

Fair Queuing (FQ): Packets of each flow are queued in a separate queue. These queues are then serviced in a round-robin mode. Fair queuing extends round-robin queuing by taking into account the count of bytes serviced from each flow rather than the count of packets, so as to implement fairness among flows with large and small packet sizes [25].

Weighted Fair Queuing (WFQ): This scheme is based on the same principles as fair queuing, with the difference that each queue is assigned a weight and the number of bytes serviced from each queue during each round is proportional to this weight. Parekh and Gallager [26] proved that WFQ for token bucket-shaped flows with appropriate weight selection can provide absolute delay bound guarantees. The development of FQ/WFQ was followed by extensive research, which led to the development of several algorithms similar in concept with WFQ such as Delay-EDD [27], VirtualClock [28], and MARS [13]. Also, as we mention in the sequel, FQ/WFQ has formed the basis of other, more scalable queuing schemes such as CBWFQ, HPFQ [29], and CSFQ [30].

Non-Work-Conserving Scheduling: The above schemes are work-conserving scheduling algorithms, in that the link is never left idle if there are packets in the queue. In non-work-conserving scheduling, the packets are not allowed to leave early. Several non-work-conserving scheduling algorithms have been proposed: Stop-and-Go Queuing [31], Hierarchical Round Robin [32], and Jitter-EDD [33]. All these algorithms typically deliver higher average delays in return for lower jitter [34].

All the above scheduling schemes constituted the initial proposals for packet scheduling. They have been thoroughly studied; overviews and comparisons can be found in [34]. They were developed having in mind that each individual end-to-end flow is serviced by a separate queue, in order to maximize fairness among flows. However, such an approach leads to scalability problems in a core node, where a large number of flows are serviced. To deal with such problems, class-based queuing and hierarchical link-sharing schemes were introduced.

Class-Based Queuing (CBQ) and Hierarchical Link Sharing Schemes: According to these schemes a queue does not service an individual flow, but rather an entire class of flows, which require similar queuing treatment (i.e. have similar QoS requirements). To guarantee fairness among the flows of the same class, a per-flow scheduling scheme (e.g. FQ) may be applied to this class. This will eventually create a hierarchy of scheduling schemes that determine how a link's bandwidth is shared first among top-level classes, then among flows of the same class, or second-level (sub-)classes of the same top-level class, and so on (e.g., [35]). Several variants of the initial CBQ idea have been proposed, most of them combining concepts of FQ with CBQ, such as Hierarchical Packet Fair Queuing (HPFQ) [29], Hierarchical Fair Service Curve link sharing [36], and Class-Based WFQ (which is an implementation of WFQ for class aggregate flows adopted by some manufacturers, e.g. [37]). A different approach for a class-based scheduling scheme is the Waiting Time Priority scheduler (WTP) [38], which aims at keeping stable the ratio of the queuing delays experienced by different assured forwarding classes in a DiffServ architecture.

Core-Stateless Fair Queuing (CSFQ): Another approach for alleviating some of the scalability problems of the per-flow FQ scheme has been proposed in [39]. The idea is that the state associated with an end-to-end flow, which is required to preserve fairness (such as the offered traffic rate) is calculated by the edge nodes and inserted into each packet as a label ("dynamic packet state"). Core nodes do not need to maintain any per-flow state, they only need to compute what should be the fair share of resources that needs to be allocated to each flow. This results in a "stateless core" (SCORE) [40].

Joint Scheduling and Buffer Management (JoBS): A recent approach has been to combine packet scheduling with queue management, in order to achieve strong service guarantees with moderate implementation complexity [41].

From an implementation perspective, CBQ and variations are the most widely adopted by device manufacturers and network designers, since they allow for the greatest flexibility in configuring the scheduling discipline for a link. A common configuration for highly delay-sensitive traffic (such as VoIP or virtual-leased line service), establishes a strict priority queue, i.e., packets of other classes are transmitted only if this queue is empty, while for other classes a class-based WFQ scheme is used. However, this requires that strict-priority traffic has to be policed at the network entrance, so that it is kept at a low percentage of the total bandwidth (e.g., 20–40 percent, as suggested in [37]) in order to avoid starvation of other classes.

Multi-Stage Queuing and Scheduling — Most research efforts on scheduling assume that the network nodes (routers/switches) are based on an output queuing architecture; i.e., when a packet arrives at an input port, it is transferred as quickly as possible to the buffer of the corresponding output port. This means that for a node with N ports, output buffer memory should be accessible at a rate N times the maximum line rate, so as to avoid contention. It is evident that such an architecture, though simple and efficient, exhausts the capabilities of packet memory when the line rates reach the gigabit scale. To avoid such a scalability constraint, a node may queue the packets at the input ports only. The drawback of input queuing (assuming FIFO queues) is that, at a given moment, only one packet from one of the N input ports can be switched to a specific output port. This means that in the case where the first packets of two input queues are destined for the same output port, one of them will have to wait, thus possibly delaying the next packet in its queue, although this may be going to a different (and available at that moment) output port. This is called head-of line blocking (HOL).

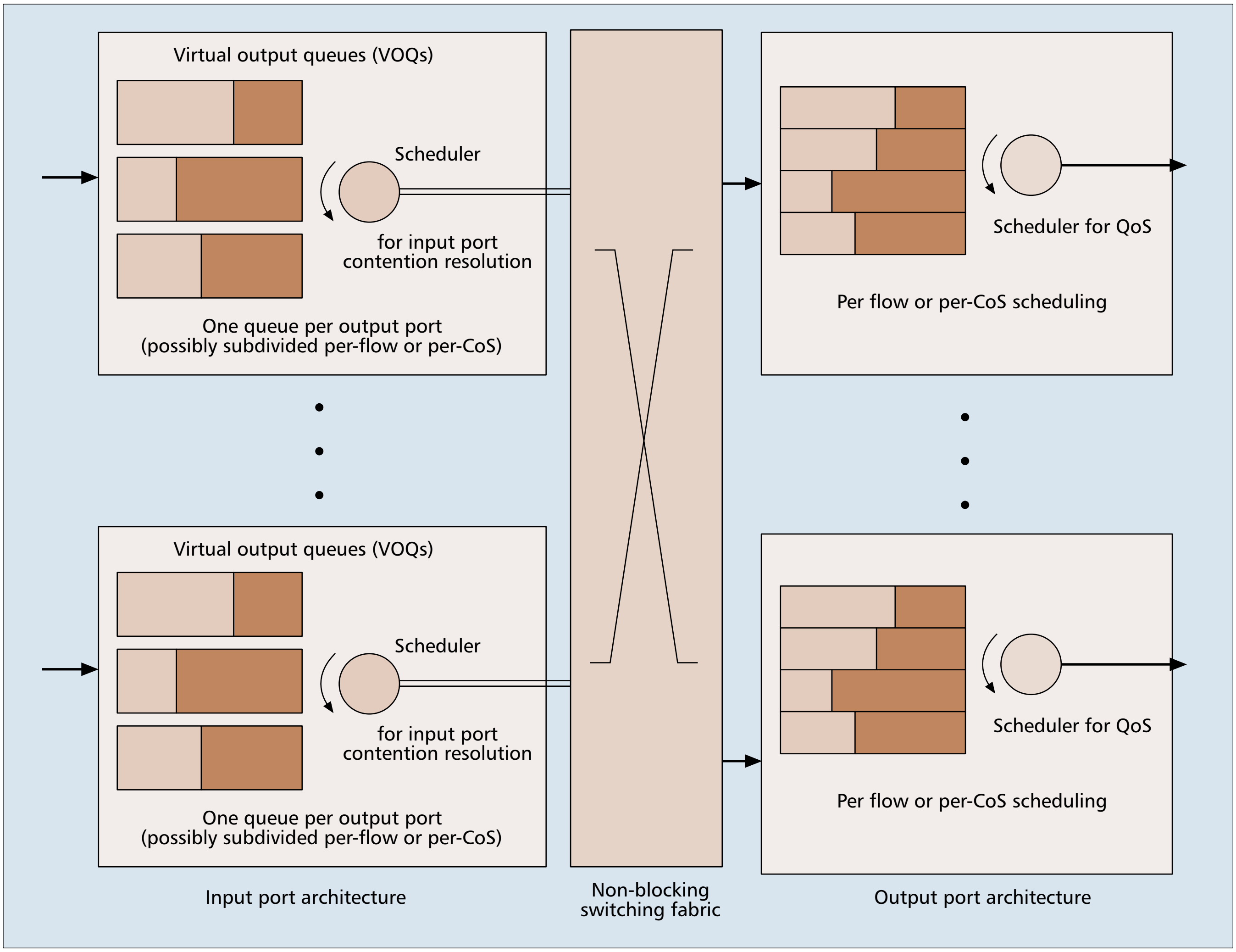


Figure 4. *Combined VOQ with output queuing architecture.*

To limit the HOL effect of input queuing, the Virtual Output Queuing (VOQ) architecture was proposed. In this technique, each input port maintains a separate queue for each output port. One key factor in achieving high performance using VOQ is the scheduling algorithm, which is responsible for the selection of packets to be transmitted in each time unit from the input queues (or VOQs) to the output lines. This algorithm has to retrieve the state of all N^2 input queues, compute a (pseudo-)optimum matching, and perform the switching accordingly, all within one cycle. In addition, the scheme must arbitrate fairly among inputs and outputs and not cause starvation of any queue. Several algorithms, such as Parallel Iterative Matching (PIM), iSLIP, Oldest Queue First (OQF), and Longest Queue First (LQF), have been proposed in the literature (see [42, 43]). It was shown that with as few as four iterations of the above iterative scheduling algorithms, the throughput of the switch exceeds 99 percent.

In view of the limitations of the two "pure" architectures, most recent proposals strive to combine the performance of output queuing with the scalability of VOQ-based input queuing. Such an architecture distributes the scheduling process to the input-port schedulers, which need only resolve contention at an input port (and not across input ports as in a VOQ-only node) and to the output port schedulers, which perform classical output queuing schemes. This is depicted in Fig. 4.

An important issue in such multi-stage queuing nodes is the application of scheduling schemes for QoS support. The difference between guaranteeing the QoS in a multi-stage queuing node and doing so in an output-queuing node is mainly due to the in the latter case, scheduling of packets enqueued in different outputs can be isolated from one another. However, in a multi-stage queuing node some packets may not be promptly scheduled across the switching fabric by the VOQ scheduler. Consequently, they may lose their chances of being serviced in time, which will result in violating their QoS.

Therefore, the key point for providing QoS guarantees in a VOQ node is to design a scheduling algorithm that can guarantee that queued packets are transmitted across the switch fabric promptly according to their QoS requirements. If the delays of queuing packets can be guaranteed, the employed scheduling algorithm will not lead to "starvation" for queued packets at any port. A number of algorithms using different methods to solve this problem have been proposed (e.g., [42]). Basically, these schemes are based on dividing each VOQ to different sub-queues per flow or per class of service (CoS). Their implementation difficulty has probably been one of the main reasons to move from IntServ-like guarantees to more qualitative guarantees. At the output queues, classical scheduling schemes can be applied.

In fact, many current high-capacity routers have adopted such architectures combining VOQ and output queuing with a non-blocking switch-fabric. For example, the Cisco GSR router uses a Modified Deficit Round Robin scheme for servicing the per-CoS VOQs as well as the per-CoS output queues.[1]

Congestion Control and Queue Management

Congestion control is an end-to-end mechanism designed to adapt traffic sources to the varying load conditions of a packet-switched network. When the network load increases to the point that congestion occurs or is imminent, congestion control should somehow "signal" the source to throttle back in order to avoid or reduce congestion. In IP-based networks, congestion control is mainly performed by the transport-layer TCP protocol (IETF RFC 2793). TCP controls congestion by adhering to a principle of "conservation of packets," i.e. the amount of information transmitted by the sender should be equal to the amount of information received and acknowledged by the receiver plus a "congestion window" which expresses, at any moment, the amount of information that is allowed to be offered to the network without receiving any Acknowledgment. The congestion window (Cwnd) size is the main parameter that controls the network load generated by a TCP session. At the initiation of a TCP session, Cwnd is set to one packet (or "segment" in TCP terms) and follows a "slow start" phase, during which it increases exponentially, until congestion occurs.[2] For TCP, no explicit signaling is required from the network's side to notify sources of congestion. Rather, a packet drop in the network (sign of congestion) will result in the expiration of an Acknowledgment timer at the source, which will then multiplicatively decrease its congestion window size by a factor of 2. After the first packet drop, the TCP sender enters the "congestion avoidance" phase, during which the Cwnd size increases linearly.[3]

Network nodes can play an active role in TCP congestion control, even without having any explicit signaling to the sources. The nodes can yet signal the sources by dropping packets from their queues, when they judge that their load is excessive. This will result in an Acknowledgment timeout, hence in a source throttling. Such a mechanism is called queue management (or buffer management).

The most obvious queue management scheme, called "tail-drop," is to drop packets when a queue has reached its maximum size. This scheme has two drawbacks. First, it tends to maintain long queues for significant periods of time, which leads to large average end-to-end delays and cannot effectively control larger periods of congestion. Second, it results in many sources throttling back simultaneously (i.e., at the time the queue fills up) and then ramping up together again, which means that eventually they will have to throttle back again when they fill up the queue. This synchronization leads to an oscillation effect (in and out of the tail-drop phase) and prevents the network from operating in a stable manner.

To prevent this from happening, network nodes can be more active in managing their queues. Active Queue Management (AQM) can be performed by letting the nodes randomly drop a packet before the queue is entirely full, with a probability that is a function of the current (and possibly also the recent past) queue size. Random Early Detection (RED) [44] was the first AQM technique proposed. RED calculates an exponentially weighted moving average (EWMA) of the queue length. After the EWMA exceeds a certain queue threshold, the router begins to randomly drop newly arriving packets with a probability that is proportional to the EWMA. This prevents concurrent throttling of all the flows and thus reduces synchronization problems.

However, the rather simple control law of RED proved hard to tune and inadequate to achieve high throughput, short average queue size, and stability under a wide range of traffic conditions [45], its main disadvantage being its slow response to sudden traffic bursts. This is why research on AQM schemes remains an open issue. In [46] the TCP/AQM system has been studied as a closed loop control system and new AQM techniques were suggested using a control-theoretic design approach. An extended control-theoretic study and the design of more robust AQM schemes are given in [47].

The above studies have made clear that congestion control is based on an end-to-end transmission protocol complemented by a queue management scheme. Apart from investigating the queue management techniques, there are several approaches that attempt to improve the source protocols. Some of these approaches were oriented toward improving TCP operation.

For instance, the Fast Retransmit algorithm was proposed to increase TCP throughput [48]. Fast Retransmit enables the sender to avoid waiting for the retransmit timeouts to expire before retransmitting packets. To achieve this, a TCP receiver should send an immediate duplicate Acknowledgment when it receives a segment out-of-order. This Acknowledgment will convey to the sender the sequence number of the last segment received in order. If a segment is dropped in the network, all subsequent segments will trigger the receiver to send duplicate Acknowledgments, with the same sequence number. Fast Retransmit dictates that the sender immediately retransmits the first unacknowledged segment, after receiving three such duplicate Acknowledgments. Also, Cwnd is set to one segment and the "slow start" phase begins again.

In current TCP implementations (such as Reno and NewReno), Fast Retransmit is complemented by Fast Recovery (IETF RFC 2581). According to Fast Recovery, after three duplicate Acknowledgments, the Cwnd size is decreased multiplicatively by a factor of 2 and the sender remains in the "congestion avoidance" phase, instead of commencing the "slow start" again. This results in a more stable behavior, since Cwnd varies around a size, which is optimal for the current network load. NewReno (IETF RFC 3782) achieves improved performance over Reno by changing the behavior of a sender, when an Acknowledgment is received for some but not all of the segments that were "outstanding" when Fast Recovery began. In such a case, NewReno will consider that the first unacknowledged segment was dropped and will immediately resend it and will stay in Fast Recovery phase, whereas Reno will wait for three duplicate acknowledgments before resending the segment.

The Reno and NewReno improvements to TCP involve modifications to the sender side only. Since the most dominant TCP senders in the Internet are the server applications (Web, FTP, e-mail, etc.), these improvements smoothly became the most common case in the Internet, as most major operating systems have adopted the NewReno mechanisms.

However, other proposed modifications to TCP involve both the sender and the receiver. For instance, to avoid dropping packets (and hence retransmitting them) in order to signal congestion, the Explicit Congestion Notification (ECN) capability was proposed and added to TCP (IETF RFC 3168). ECN allows the router to mark a packet on its way to the des-

[1] see the URLs:
http://www.cisco.com/en/US/products/hw/routers/ps167/products_white_paper09186a0080091fdf.shtml,
http://www.cisco.com/en/US/products/hw/routers/ps167/products_white_paper09186a0080091fdc.shtml

[2] Actually Cwnd increases by one for each acknowledged segment in that phase, hence the term "slow start"; however, this results in Cwnd doubling, when all segments within the current window are acknowledged in time.

[3] Now, Cwnd increases by one when all segments within the current window are acknowledged in time.

tination instead of dropping it. A TCP receiver receiving an ECN-marked packet would convey this information back to the sender on a subsequent response. A TCP sender should react to ECN in the same manner is it reacts to a packet drop. Avoiding packet drops just to signal congestion results in improved throughput. All recent AQM studies take into account the effect of ECN. Several research efforts also focused on the addition and exploitation of a Selective Acknowledgment (SACK) option to the TCP header (IETF RFC 2018) [49]. The SACK option can be sent by a receiver to acknowledge reception of non-contiguous blocks of data. Thus, in the face of multiple dropped segments, the receiver can inform the sender about all segments that have arrived successfully, so the latter need only retransmit the segments that have actually been lost. Another notable approach is Fast TCP, which proposed and evaluated several modifications in the TCP sending operation [50]. Other proposals focused on the issues that must be dealt with in modern high-bandwidth and long-distance networks, such as the Explicit Control Protocol (XCP) [51].

To sum up, although some years ago it was deemed that the ratio of TCP to non-TCP traffic in the Internet was going to decrease, several factors such as the incessant growth of the Web, the interest in Grid computing, and the introduction of distributed computing frameworks based on HTTP (Web services) led to a revived interest in TCP-related research.

Although modifying both the sending and receiving operation of TCP yields promising results, a major issue with such approaches is that they require modifying TCP/IP stacks of a considerable number of end-systems. On the other hand, most routers currently run a version of RED; introducing new AQM schemes involves much fewer devices and could be done incrementally, e.g., by first upgrading router software in specific portions of the network.

Congestion control has also been applied to several kinds of non-TCP traffic. For instance, media playback applications can monitor network congestion and revert to a less bandwidth demanding encoding. This is especially applied in the context of layered media encoding schemes such as MPEG-2 and MPEG-4 [52].

QoS Routing

The task of routing in IP networks is to perform hop-by-hop forwarding of a packet from the source to its destination. Each router forwards the packet either to another connected router or to its destination according to its routing table and forwarding information base. Conventional IP routing is based on a best-effort model: selecting the path with the minimum cost, which is usually inversely proportional to the capacity of the links. However, the emergence of applications with QoS requirements has given rise to routing based on additional constraints, such as delay or bandwidth constraints. Routing based on additional constraints is a key requirement to providing QoS guarantees. Therefore, the associated routing mechanisms and protocols are characterized as constraint-based routing or QoS routing. Apart from supporting QoS requirements, constraint-based routing is also important for creating virtual IP paths, which are the cornerstone of VPN services. Furthermore, it allows avoidance of congested paths.

QoS routing constitutes a rather complex topic with several facets. Complexity arises from the multitude and diversity of QoS requirements imposed by IP applications, which are likely to target a variety of constraints including delay, delay jitter, bandwidth, etc. Problems with multiple constraints are usually not easily tractable, and in some cases impossible to solve. At the same time QoS routing has to deal with-best effort traffic that aggravates the computational complexity. Furthermore, as QoS routing algorithms need to take into account real-time data (e.g., changing paths, traffic conditions) there are a host of issues affecting the scalability and deployment of QoS routing mechanisms.

QoS routing algorithms can be classified into various categories. One possible classification is between intra-domain and inter-domain routing algorithms. Most research has focused on the intra-domain problem, i.e. building routing tables in the scope of a single administrative domain. Several solutions have been proposed toward satisfying bandwidth [53], delay [54], and in several cases multiple constraints [55]. These routing solutions deviate from the conventional link-state protocols (e.g., OSPF) and distance vector protocols (e.g., RIP) toward taking into account additional constraints.

Contrary to intra-domain routing, inter-domain routing is designed to route packets across administrative domains. In the inter-domain routing problem, routing information is aggregated based on the hierarchical structure of the IP network (e.g., the Internet) toward reducing the global state of the problem, and therefore achieving scalability. The Border Gateway Protocol (BGP) is the dominant protocol allowing exchange and dissemination of inter-domain routing information. Appropriate filtering of BGP requests has been proven capable of optimizing the use of network resources across domains, toward providing QoS [56].

QoS routing can also be classified under two different paradigms. The first paradigm pertains to the shortest-path routing model, where the aim is to calculate the shortest path that satisfies specific QoS requirements. Most of the relevant approaches have strived to appropriately extend link-state protocols (e.g., IETF RFC 1584, [57], [58]) or alternatively distance vector algorithms (e.g., [59]). The majority of these solutions address bandwidth constraints. However, there are also schemes that address delay constraints (e.g., IETF RFC 1584) or combinations of bandwidth and delay constraints (e.g., [60]).

Except for shortest-path QoS routing, there is also another paradigm aimed at pinning routing paths that handle aggregate traffic flows that match certain QoS or source-destination criteria. This mode of operation is aimed at changing routes toward capturing long-term variation of the traffic load and is characterized as traffic engineering (IETF RFC 3272). Traffic engineering is performed by computing paths that are able to accommodate slowly changing aggregate traffic patterns, according to multiple constraints. This can be accomplished through extending conventional routing protocols to monitoring network performance and accordingly directing traffic to pre-computed paths. Traffic engineering is acknowledged as an important control mechanism, which is constantly gaining momentum. It is no accident that one of the major advantages attributed to MPLS technology is that it supports the deployment of traffic engineering mechanisms (characteristic examples can be found in [61, 62]).

Another distinction of QoS routing schemes is between unicast and multicast. Unicast QoS routing solutions relate to establishing a path between a given source and a destination subject to QoS constraints, while multicast routing (e.g., [63, 64]) addresses the discovery of a tree covering a source and multiple destination nodes subject to a set of constraints.

Concerning the strategy for computing the routing path, we can distinguish source, distributed, and hierarchical routing algorithms. In the scope of source routing algorithms, (e.g., [55, 58]) each node maintains global state allowing computation of the end-to-end path at the source. Thus, a distributed problem is transformed into a centralized one. In distributed routing algorithms (e.g., [65]) the path computation is conducted in a distributed manner by the intermediate nodes comprising an end-to-end path. Finally, hierarchical routing algorithms (e.g., [66]) allow computation of routing paths in the scope of multilevel hierarchies constructed based on appropriate groups of nodes.

QoS Policy Management

Toward provisioning QoS over heterogeneous networks, all the building blocks discussed above have to work together according to appropriate policies. The necessity for combined blocks operation imposes the QoS policy management building block. The most widely adopted specification for QoS service offerings is the Service Level Agreement (SLA) and Service Level Specification (SLS). SLAs describe the characteristics of the service offering and the responsibilities of the parties involved for using and providing the offered service. The SLSs give the technical characteristics of the service offered in the context of a SLA. The service technical characteristics refer to the provisioning aspects of the service, e.g. request, activation, and delivery from the network perspective. For example, a SLS can include the following attributes: SLS identification, scope of SLS (ingress and egress points), flow identification (e.g., DSCP), traffic conformance and characteristics, traffic excess treatment, performance guarantees to be given to the traffic (in terms of delay, jitter, loss, bandwidth), etc. Non-technical service provisioning aspects, such as billing, security, authentication, etc., are part of an overall SLA and not part of a specific SLS. Generally speaking, SLSs are an integral part of a SLA, and conversely a SLA includes various SLSs.

There is a clear distinction between SLA negotiation protocols (such as SNAP [5]) and QoS-signaling or reservation or QoS-enabled session control protocols (e.g. RSVP, SIP (Session Initiation Protocol)). Specifically, SLA negotiation protocols are used for agreeing on QoS policies provided by the SLA under negotiation, whereas QoS-signaling or reservation or session control protocols are used for signaling/requesting the level of QoS that users require, considering that the respective SLAs with the network or service providers have already been agreed. SLA negotiation protocols operate at service subscription epochs, where users subscribe to the desired services offered by the providers, while QoS-signaling or reservation or session control protocols operate at service invocation epochs, where the users call for the services to which they have been subscribed. The distinction between service subscription and invocation is essential; usually a AAA (authentication, authorization, and accounting) function will check conformance of user service invocation requests against service profiles agreed during service subscription.

Prior to sending user traffic across multiple domains, it is necessary that SLAs between service providers (SPs) and network providers (NPs) or between NPs are also negotiated and put in place. Network resources are usually allocated in an aggregated level through different means after the subscriptions have taken place and during network provisioning cycles. After the allocation of resources, the service may be invoked for sending user traffic. In this context, the provider-level SLAs do not handle individual user streams but aggregated flows, which requires certain end-to-end QoS guarantees.

Usually a network operator applies its own policy-based management rules through a set of a PDPs (policy decision points) and PEPs (policy enforcement points) [67]. PDPs are likely to use an LDAP-based (Lightweight Directory Access Protocol) directory service for storage and retrieval of policy information. The harmonized functionality among heterogeneous network domains can be achieved at the PDP level through a distributed management system coordinating all the involved PDPs. In this way, the underlying network management system of each network domain remains unaffected. A protocol, such as COPS (Common Open Policy Service) (IETF RFC 2748), conveys self-identifying objects for relaying policy decision and reporting. This communication protocol is appropriate for PEP-PDP exchange messages. Appropriate COPS extensions can be used for the QoS management of PEP/PDP nodes of both IP and non-IP networks.

Pricing

The provision of services that feature better than best-effort quality in the scope of IP-based networks is directly associated with the availability of pricing schemes [68]. Service providers offering quality services must charge for them toward maximizing their revenues, while also avoiding unnecessary overuse and over-allocation of network resources. Thus, in a QoS environment pricing schemes serve as a vehicle to guiding users to use the level of service best matching their requirements, while also enforcing some sort of social fairness [69, 70]. One may argue that pricing for network services is probably impacted more from the service providers and network operator marketing and business strategies, rather than engineering facts. While this is true to a great extent, pricing policies are also influenced by engineering issues.

The most widespread pricing schemes fall into the class of flat pricing strategies. Flat pricing schemes charge users a fixed amount for a specified time period, independent of usage. The major advantage of flat pricing is its simplicity, as well as its independence from particular network technology, protocols, or QoS mechanisms. The major disadvantage of flat pricing is that it provides no pricing differentiation for users enjoying different quality levels. As a result, flat pricing pertains to over-provisioning mechanisms and is therefore hardly combined with traffic management and congestion schemes. In fact, the prevalence of flat pricing is debated as one of the factors combating QoS deployment, since it does not support service providers in recovering additional cost for deploying QoS.

A direct extension of flat pricing, called "Paris Metro" pricing [71], applies different flat rates to different logical divisions of a global network, resulting in lower network utilization and therefore better quality in higher priced logical networks. While this approach improves the quality enjoyed by several user groups, it is not appropriate to support service differentiation resulting from traffic control functionality, such as this illustrated in the scope of previous paragraphs.

Priority pricing [72] is a scheme associated with service differentiation, since it allows users to mark the relevant priority of their packets and be charged accordingly. Several priorities based on different criteria (such as delay and loss) can be defined in the scope of priority pricing. Priority pricing acknowledges that network resources are precious and therefore is appropriate for use with traffic management schemes. Relevant priority criteria for charging are also used in the scope of smart-market pricing, which adopts an auction/market-based approach toward charging for usage in periods of congestion.

Pricing based on effective bandwidth (see also the subsection on "Admission Control") constitutes the most representative form of usage-based pricing. Effective bandwidth pricing is based on the traffic profile of the charged source (e.g., the values for the mean and the peak rate). Assuming that these traffic parameters can be specified, the user is charged according to a linear function, placed tangent to the effective bandwidth curve of the source [73].

Another pricing scheme that is perfectly tailored to ATM/IntServ mechanisms is edge pricing, which calculates charges based on expected values of congestion and routes. The decoupling of pricing from usage and its association with congestion at particular time instants following an admission control function illustrates its pertinence to ATM and IntServ. Also, edge pricing is receiver-oriented (i.e. the end user is charged as a receiver), which is conceptually in-line with several content distribution applications [74]. Other Internet pricing schemes that can be used with IntServ are presented in [75].

Most of the above schemes fall into the broad area of static pricing policies, where the pricing function is independent from the network utilization in the course of the transmission/session charged. Contrary to static pricing schemes, dynamic pricing schemes calculate prices taking into account the ongoing state of the networks. Thus, dynamic strategies can react to traffic fluctuations, through keeping track of the network traffic and accordingly providing users with the optimal price for a particular time instant. A characteristic example of a dynamic pricing scheme is responsive pricing. Responsive pricing calculates prices based on the network state and accordingly sends price signals to end users. End users are then likely to adapt their behavior, thus achieving overall network and economic efficiency [76]. Response pricing, as well as other dynamic pricing mechanisms, are very often criticized as being computationally complex and overall impractical. Also, it seems quite hard for users to understand and accept them.

No matter which pricing scheme is adopted, pricing can have a direct impact on the functionality of the building blocks discussed so far. In particular, pricing can be employed as a means to supporting congestion avoidance, given that high prices discourage users from injecting additional traffic into the network. Moreover, the price of a network service/connection may also impact the admission control process, since it can act as an additional criterion. Dynamic pricing strategies may also have an effect on the volume of traffic offered by users and its distribution over time, which are valuable parameters in performing resource management, provisioning, and network planning. Being a factor that can help engineering a network, pricing can also be used in conjunction with other traffic control functionality, such as the already discussed routing and policing functions.

Combining Building Blocks in Network QoS Design

In the previous section we surveyed research efforts toward devising efficient mechanisms for the implementation of the packet-level QoS building blocks that we identified earlier. Most standardized QoS frameworks, such as these described earlier, generally determine a fixed combination and placement of these QoS building blocks in a network. As already stated, however, it is not common to find an IP network where a standardized QoS framework has been deployed exactly as specified. This is chiefly due to the diversity and heterogeneity of IP networks in terms of their architecture, the capabilities of network devices, as well as the QoS requirements of applications and users.

It is thus more practical for a network designer or engineer, instead of having to entirely adopt a specific standardized QoS framework, to select among a pool of building blocks and associated mechanisms that instantiate them and subsequently craft the QoS architecture that best suits the particular aspects of each network deployment. This selection can be based upon a number of factors, such as:

- The architecture, resources, and technologies of the network, e.g., whether it is an enterprise or service-provider network, a campus network, or a WAN, whether it has high or limited bandwidth resources, etc.
- The requirements of the network users and applications, as well as the operational/business policies and requirements of the organization running the network.
- The capabilities of the network devices and the mechanisms that they support in order to instantiate QoS building blocks.
- The complexity that the designer/engineer is willing to deal with, in order to support QoS. Usually, the more complex the QoS architecture and the more sophisticated the related mechanisms, the better and more fine-grained is the resulting QoS.

Based on the above factors, the network designer can select which blocks to combine, where to place them, and which will be the specific mechanisms and algorithms for the realization of each building block. Furthermore, depending on the underlying link-layer technologies, some additional technology-specific mechanisms may be needed, so as to ensure that the packet-level QoS achieved by the building blocks is maintained across some special network segments. Examples of such segments, in particular broadband access, mobile and optical networks, and their associated tools for maintaining packet-level QoS, are discussed in the following section.

The approach of combining building blocks in order to devise a QoS design is commonly followed [77], and it is not entirely new. DiffServ can be considered as an intermediate step toward this approach, since it specified a targeted combination of blocks and their placement, but left enough open space to select to which extent the full or a partial combination will be used and which will be the specific mechanisms. The forthcoming Recommendation Y.1291 of ITU-T SG13 is a standards effort that adopts a building blocks approach in devising a QoS architectural framework, as described in [6].

In the sequel, we illustrate the rationale that a network designer may follow in order to devise a QoS architecture using building blocks through some simple but indicative examples. Through these examples we also stress another important issue: when combining building blocks and selecting specific mechanisms that instantiate them, these no longer work in isolation, but rather the operation and performance of each block is influenced by the selections made for another block.

As a first simple example, we consider the interoperation of congestion control with queue management. The vast majority of relevant research results (e.g., [44, 45, 47]) conclude that, regardless of the particular topology and resources of a network, enabling a queue management scheme at the network nodes (even not at all of them) will boost the throughput achieved by congestion control mechanisms of TCP. The designer here needs to be aware that certain queue management schemes are combined more efficiently with specific TCP implementations. Thus, knowledge of the dominant TCP implementation in a network can result in a more optimal choice of the mechanism to be realized in the queue management building block, provided of course that the network devices can support it.

As a second example, we consider an organization running an enterprise-wide IP network, which is used to support both the mission-critical business applications of the organization, as well as the Internet access needs, e.g. for Web browsing. The organization has decided that the business applications should be allocated a guaranteed portion of the network resources and that the Internet traffic should be treated in a best-effort manner. To meet such a requirement, a scheduling building block should be deployed in all network devices. This block should be implemented with a class-based queuing scheme, so that business application traffic is classified and serviced by a separate queue to which the scheduling algorithm assigns the desired priority and guarantees resources.

We can further complicate this example by assuming that the organization wants to put a per-user quota to the bandwidth available to each user for Web browsing. In such a case, a QoS policy management block should be deployed in the network, in combination with a policing block that needs to be deployed at the Internet access links. The QoS policy management block will control the policing block by enforcing the parameters according to which the latter will monitor and forward or drop Web traffic destined to each user.

Next we consider that this organization also adds VoIP as an application. Since VoIP traffic demands stringent QoS guarantees, the architecture of the QoS building blocks deployed in this enterprise network will need considerable revision. VoIP traffic is not elastic to queuing time fluctuations, thus it needs an amount of network resources exclusively allocated to it. The class-based queuing implementation should be modified by adding a separate queue for this traffic. Because of the tight delay requirements of VoIP traffic, it is usual to allocate a guaranteed amount of bandwidth to this queue and service it with strict priority [77, 78]. However, this amount of bandwidth will be able to accommodate a fixed number of VoIP flows. Having more VoIP flows on a link will result in violating their strict QoS requirements. Moreover, if a strict priority queue is used for VoIP traffic, then the VoIP traffic volume has to be kept at the allocated level, otherwise, queues servicing the other traffic types may starve. Therefore, a signaling and resource management block needs to be deployed in combination with an admission control block, so that the end stations can signal requests for VoIP call establishment that will subsequently be admitted or rejected, depending on the availability of bandwidth. The implementation of these blocks is pertinent to the protocols and devices selected for implementation of IP telephony (SIP, H.323, gatekeeper, etc.). The routing building block can also play an assisting role in such a case, by pinning specific routes for the VoIP traffic, which are kept clear of the fluctuating traffic of data applications.

Now the co-existence of traffic responsive to congestion-control (i.e., TCP) with non-responsive traffic (i.e., VoIP) in a class-based queuing block demands attention to the interoperation of queue management and queuing. Queue management achieves the desired results only when applied to traffic responsive to congestion control. On the contrary, applying queue management to non-responsive traffic may seriously degrade the QoS of such traffic. Thus, queue management should not be applied to the queue servicing VoIP traffic.

Lastly we consider an example involving interworking of IP with a special link-layer technology: the deployment of wireless LAN hot-spots as an access segment to an IP-based infrastructure. WLAN deployments are usually based on two general scenarios. The first scenario refers to stand-alone hot-spots that simply provide access to a fixed IP backbone. The QoS-effective design of this scenario requires building blocks that can guarantee efficient network provisioning and can properly satisfy micro-mobility issues, when a mobile user moves from one cell to another adjacent cell of the hot spot. The first requirement is covered by a QoS policy management block that enforces the appropriate policies between the exploiter of the hot spot and the operator of the IP backbone, while the second one is dealt with as an appropriate instantiation of a QoS routing block. Note that in such settings routing can be limited in the fixed IP backbone domain.

The second scenario refers to the case where the hot spots constitute an alternative access network for the subscribers of a cellular operator. In this case the QoS-effective design requires building blocks that, besides ensuring the efficient network provisioning, can properly satisfy macro-mobility issues, when a mobile user moves from one access network (e.g. WLAN) to another access network (e.g. GPRS). For handling macro-mobility, appropriate instantiations of building blocks for inter-domain routing, resource management, and admission control must be used. Apparently, the pricing building block concerns both the above scenarios.

QoS in Broadband Access, Mobile and Optical Network Segments

Large-scale inter-networks consist of various network segments of different technologies, as illustrated in Fig. 1. Specifically, networks encompass an access part, which can be based on either wired or wireless access technologies. Roaming users connect to IP intranets of the Internet through mobile/wireless networking infrastructures. Moreover, backbone networks are increasingly relying on optical network technologies. In the scope of highly heterogeneous large-scale inter-networks (such as the Internet), IP is used as the main internetworking protocol.

Providing QoS in such large-scale inter-networks requires more than a combination of the building blocks analyzed in the previous section. because packet-level QoS control mechanisms are not sufficient to support quality services on certain network segments. As a result, additional non-packet-level QoS mechanisms have been devised and developed for these segments, with a view to maintain or enhance the QoS provided by packet-level mechanisms. Network operators can make use of these additional mechanisms with a view to deploying QoS solutions spanning more than one segment and technology. In this way, service quality is significantly improved.

Even though such mechanisms are not directly related to the building blocks outlined above, they have an impact on research work related to advancing several building blocks. This is because their deployment (e.g., in access, wireless, mobile, and optical networks) has to deal with the interworking with IP QoS mechanisms outlined in previous sections. Such an interworking is a prerequisite to achieving seamless QoS provisioning across network segments and technologies. Moreover, research is required to achieve a longer term objective, which is to automate provisioning processes across heterogeneous network segments.

In the sequel, we highlight key technologies targeting QoS issues in the scope of such network segments, while also discussing their interworking with conventional IP QoS frameworks. As far as access networks are concerned, we focus on broadband access technologies. With respect to mobile networks, we report on the QoS efforts undertaken in the scope of third generation (3G) mobile systems, while also providing information on QoS support for composite radio infrastructures. A more thorough survey of QoS in the mobile computing environment is given in [79]. With respect to optical network solutions we focus on recent efforts toward integrating a control plane into optical network elements. This control plane aims at offering traffic control capabilities, smart QoS services, and optimizing interworking with neighboring IP networks.

BROADBAND ACCESS NETWORKS

The bandwidth abundance in the network core, along with the deployment of QoS mechanisms, render the access network segment (i.e., the "last mile") an important bottleneck, especially for residential users. QoS in the last mile can also be controlled at the IP layer by employing some of the building blocks described earlier in this article, such as: scheduling, policing, and queue management. However, due to the nature of some access technologies, it is important to employ QoS mechanisms specific to each technology and moreover to employ them in a way that interworks with the mechanisms deployed at the IP layer.

For instance, in DSL access networks, traffic passes through at least one ATM switch (the DSLAM) and sometimes through an entire ATM network, before reaching the first IP hop. It is possible that all traffic is transmitted on a single ATM VC, which means that the ATM-based access network will treat all IP traffic types with the same service level, regardless of the service level they have been assigned at the IP layer. Consider, for example, a user receiving VoIP and Web traffic. Even though these two traffic types may be differentiated at the IP hop, they share the same ATM VC through the DSL access network. This means that in case congestion occurs in the ATM switches, it cannot be guaranteed that VoIP traffic will maintain the desired QoS. Mapping traffic of different IP service classes to different ATM VCs of an appropriate ATM service class can permit guaranteeing QoS end-to-end. In our example, VoIP could be mapped to a CBR VC.

Likewise, in a cable access network several issues stem from the fact that users may share the bandwidth of a coaxial segment of the cable infrastructure. On the one hand, it should be guaranteed that each user has access to a fair and committed share of the coaxial segment's bandwidth, and on the other hand, that flows belonging to different IP service levels are prioritized accordingly, when transmitted over the cable network. Such tasks should be undertaken by the MAC protocol of the cable network. The DOCSIS Radio Frequency Interface Specification (v. 1.1), by CableLabs, Inc., specifies an enhanced MAC protocol for data-over-cable networks that addresses such QoS issues. The basic concept for DOCSIS v. 1.1 QoS is the service flow, a sequence of packets that is assigned a certain priority for accessing the shared medium through the DOCSIS Medium Access Control scheduling algorithm. Higher-layer protocol flows can be mapped to MAC service flows according to various criteria (IP addresses, ToS byte/DSCP, application port numbers, etc.) so that they are assigned the appropriate resources by the MAC layer both in the upstream as well as in the downstream direction. In this manner the QoS of IP flows can be maintained over a cable access network.

Similar concepts also need to be applied in the case of a Wireless LAN access network, such as IEEE 802.11a/b/g. Here again, users are contending for access to a shared medium, the 802.11a/b/g frequency band. Fair access to this medium is governed by the Distributed Coordination Function (DCF), the most widely adopted 802.11 MAC algorithm. The IEEE 802.11 working group has defined, within the 802.11e specification, an enhancement to DCF that allows different flows to have different priorities for accessing the radio links by altering the inter-frame space and collision backoff timer of the DCF.

MOBILE NETWORKS

Current state of the art mobile networks are based on third generation (3G) systems. 3G was originally conceived by ITU as a single worldwide standard called FLMTS (Future Land Mobile Telecommunication System). After the ITU phase ended in about 1998, two bodies — 3GPP and 3GPP2 — completed the first standardization of the two flavors of 3G, known as UMTS [80] (developed and promoted by Europe and Japan) and cdma2000 [81] (developed and promoted by North America), respectively. Several of the building blocks presented in the previous sections work together in providing QoS in current releases of UMTS and cdma2000 systems.

3G systems can be divided into three network parts: the air interface, the radio access network (RAN), and the core network. The air interface concerns the technology of the radio hop from the terminal to the base station. The RAN is the glue that links the core network to the base stations and deals with most of the consequences of the terminal's mobility. In well designed networks, the air interface will be the major bottleneck where most congestion and QoS violations will take place. Appropriate functions, mechanisms, and schemes activated by the corresponding building blocks distributed between the terminal and the RAN undertake to guarantee QoS provisioning over the radio link.

For the UMTS RAN (UTRAN), the corresponding standards [82] foresee functionality pertinent to the IP QoS building blocks. These include admission control, resource management (i.e. power and code management), packet scheduling, and mobility management. Note that as the standards do not specify exactly how QoS is implemented in UTRAN, but only how QoS is signaled across the provided interfaces, the different operators implement their own schemes for the realization of the mentioned QoS building blocks. Ultimately, QoS provisioning in UMTS is achieved through the concept of "bearers." A bearer is a service providing a particular QoS level between two defined points invoking the appropriate schemes for either the creation of QoS guaranteed circuits, or the enforcement of special QoS treatments for specific packets. The selection of bearers with the appropriate characteristics constitutes the basis for UMTS QoS provisioning. Each UMTS bearer is characterized by a number of quality and performance factors. The most important factors are:

- The bearer's traffic class. Four traffic classes have been defined in the scope of the UMTS framework (i.e., conversational, streaming, interactive, and background).
- Whether or not the requested class is negotiable.
- The bearer's maximum bit rate and guaranteed bit rate.

As already mentioned, an alternative method for the development of the 3G systems is the U.S. cdma2000 flavor, which unfortunately is not compatible with UMTS. Targeting an all-IP solution based on existing Internet protocols and standards, 3GPP2 and the U.S. standards group TIA (Telecommunications Industry Association) promote for the cdma2000 systems a packet core network (PCN) that can deliver services using IP protocols from end to end. Due to the all-IP goal, the work in 3GPP2 is closely tied to work in the IETF (Internet Engineering Task Force). The major difference between the PCN and UMTS is in the maner in which mobility management is handled. In UMTS, this is handled in the HLR (home location register) and uses SS7 signaling, while in the PCN it is based on mobile IP, an Internet mobility concept. PCN and UMTS evolutions might be expected to converge, although backward compatibility with their earlier releases is one stumbling block.

According to [83], end to end QoS support in the cdma2000 systems is going to be provided via one or more

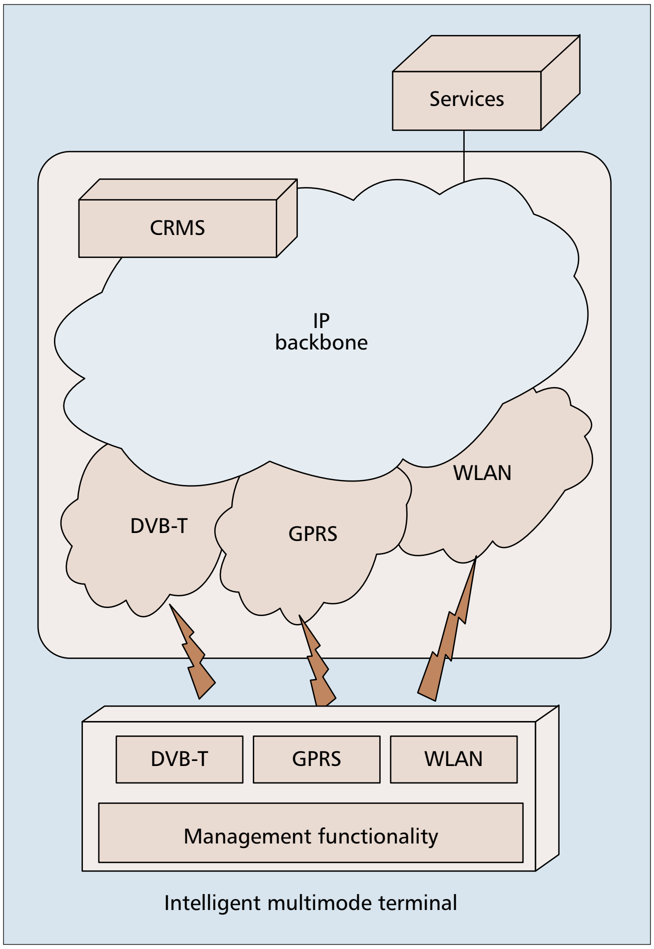


Figure 5. *Example composite radio infrastructure.*

instances of a packet data service. Two types of packet data service instances have been specified: main service and auxiliary service. The main service instance is set up during the initial establishment of a packet data service. This packet data service instance normally has default QoS characteristics that are based on the subscriber's AAA (authorization, authentication, accounting) profile and local policy. The auxiliary service instance is set up on-demand to support a required QoS greater than the default QoS characteristics that are configured for the main service instance. This packet data service instance has QoS characteristics that are based on the request of the user, limited by the subscriber's QoS profile and local policy. The subscriber's QoS profile contains limits on the resources that can be authorized for use by a subscriber. These limits may include maximum allowed bandwidth, minimum delay, or minimum packet loss rate, and may contain default values greater than best-effort, when no specific QoS request is signaled. One or more auxiliary service instances may be established by a mobile user based on the number of applications in use, each requiring different QoS attributes. Note that as in the UMTS case, with respect to the QoS attributes four traffic classes (i.e., conversational, streaming, interactive, and background) are defined. In the cdma2000 systems, the radio resources are allocated per service instance to ensure that the requested QoS requirements for a user's application are satisfied. If the necessary resources are not available, an attempt should be made to negotiate a lower QoS.

Similar to the UMTS case, the cdma200 standards [84] do not specify exactly which schemes are implemented for the realization of the QoS building blocks. When a mobile station establishes a packet data service, it originates a main service instance and may open one or more auxiliary service instances, to carry traffic that is not suitable for the main service instance. For example, a mobile station may have a main service instance for TCP and an auxiliary service instance to carry an RTP video stream. Various IETF-based traffic mapping and processing techniques can be applied to the flows carried by the service instances. The mobile station defines these techniques in such a way that downlink packets are routed to the service instance that matches the characteristics of the receiving application. Furthermore, in accordance with the differentiated services standard, a mobile station can mark its packet according to the corresponding user profile. The PCN may re-mark the packets according to local policies in cases where the type of marking is not authorized. Moreover, the PCN can reject service requests through appropriate admission control schemes, in order to preserve the QoS of the requests being served.

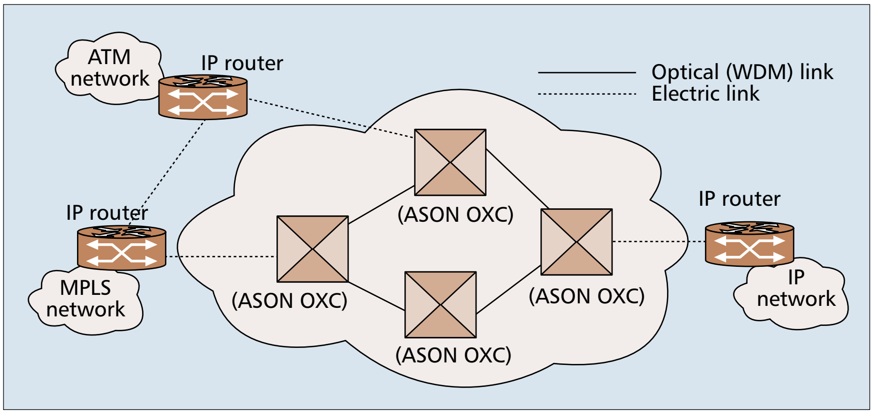
Besides UMTS and cdma2000 3G systems, various other technologies have contributed to the current success of mobile/wireless communication, including the family of broadband radio access networks (such as IEEE 802.11 and HIPERLAN) and the wireless broadcasting technologies, such as digital video broadcasting (DVB-satellite and terrestrial). Apart from the evolutionary paths of individual technologies, considerable interest has been recently directed toward the additional benefits that may arise from their joint exploitation. Interoperability in the network among various air interfaces (and other access media) become more and more important as wireless service providers expand the scope of their telecommunications businesses through partnerships and increased service offerings. In this context, it seems that the 4G systems will be networks of networks with good synergy that will be beneficial to global operators and to the industry as a whole.

In addition to 3G/4G systems, the concept of composite radio networks gains an important place in this emerging mobile/wireless setting. A composite radio system is not simply a system where terminals switch to alternative access networks through a vertical handover upon loss of coverage, but rather as a system where its constituent components coordinate intelligently, toward exploiting the increased potential for optimization that becomes possible when these constituents are jointly operated. Operation at this level of intelligence presupposes the existence of appropriate management functionality; such functionality is assumed to be available at both the composite network and the wireless terminals. The management functionality possessed by the terminals is intended to capture a localized "view" of the system, in accordance with the conditions pertaining to each particular terminal. Consequently, relevant optimizations (intelligent selection of the appropriate radio segment) are of a distributed character; furthermore, they are scalable and may operate at a near real-time fashion.

Figure 5 illustrates a composite radio environment consisting of three radio segments: WLAN (IEEE802.11b), DVB-T, and GPRS.

The management functionality possessed by the terminals is intended to capture a localized "view" of the system, in accordance with the conditions pertaining to each particular terminal. Consequently, relevant optimizations (intelligent selection of the appropriate radio segment) are of a distributed character; furthermore, they are scalable and may operate at a near real-time fashion.

On a more global scale, optimal joint utilization of the heterogeneous resources in the composite environment is achieved through a composite network process, which should be capable of:



■ **Figure 6.** *Client IP-based networks connected to an ASON optical backbone.*

- Monitoring and analyzing the performance and the service QoS levels that may be achieved over the various segments of the composite radio infrastructure.
- Performing dynamic reconfigurations of the overall managed infrastructure and/or appropriately redistributing the traffic load to the radio segments, as a result of resource management strategies for handling new environment conditions (i.e., traffic load, mobility levels, etc.) in an efficient manner.

OPTICAL NETWORKS

Most optical deployments to date have focused on providing high-capacity point-to-point bandwidth pipes between adjacent client equipment (e.g., SONET/SDH Digital Cross-connects (DXC) or IP routers). The introduction of Wavelength Division Multiplexing (WDM) made it possible to transmit vast amounts of traffic over a single optical fiber. This high bandwidth availability came with a trend toward removing processing layers between optical transport and IP, which is expressed as a proliferation of IP over WDM networks. This reduction of layers provides opportunities for more cost effective operation and management of the optical backbone network. QoS is mainly addressed based on bandwidth abundance and over-provisioning.

However, more sophisticated QoS features are provided in emerging optical architectures, as operators seek ways to reduce their capital and operational expenses through automating provisioning and management of lightpaths [85]. It is envisaged that automated provisioning will provide flexibility into deploying new revenue generating services (e.g., Storage Area Networks (SANs), Optical VPNs, and Grid Networking Services). Automating the provisioning and management of lightpaths requires the introduction of a control plane on optical networking elements (i.e., OXC — Optical Cross Connects). Based on this control plane, it is also possible to control QoS characteristics of the lighpaths.

Much effort has been allocated toward this control plane, both from standard bodies and the industry. These efforts resulted in protocols such as the Multi-Protocol Lambda Switching (MPLambdaS), the Generalized Multi-Protocol Label Switching (GMPLS), the Optical Network-to-Network interface (O-NNI), and the Optical Signaling and Routing Protocol (OSRP). Following these initiatives, the ITU-T defined the Automatic Switched Optical Network (ASON) [86], as a more specific case of another model, namely the Automatically Switched Transport Network (ASTN). ASON and ASTN are protocol-independent, since ITU-T has focused on defining the frameworks and then specifying the protocols. The IETF has focused on the Generalized Multiprotcol Label Switching (GMPLS) [87] toward specifying control algorithms and protocols for the optical control plane. GMPLS extends label switching to lambda and fiber switching, and accordingly provides extended versions of the LDP (IETF RFC 3212) and RSVP protocols (IETF RFC 3473). Apart from GMPLS, there are also alternative proposals exploiting other control protocols such as ATM's PNNI [66].

Based on ASON functionality and its supporting control protocols, QoS features can also be implemented in the emerging optical backbone networks that are expected to serve several client (e.g., IP, MPLS, DiffServ) sub-networks. ASON networks support traffic control functionality at the lightpath level. Switched lightpaths can be established on demand subject to specific bandwidth constraints, as well as subject to particular routing policies. The amount of bandwidth as well as the required routing of the lightpath can be determined based on the demands of the client network. Traditional tele-traffic models (e.g., Poisson, Fredericks, Engset) can be considered toward modeling the arrival of requests for switched lightpaths, as well as their holding times [88].

The optical backbone network supports traffic engineering mechanisms toward alleviating potential traffic congestion resulting from shortest path routing. This is accomplished through exploiting the control plane for routing traffic based on the traffic load conditions of the network. A popular mechanism is "traffic grooming," which allows reuse of already established lightpaths as part of an end-to-end connection through the optical network [89]. Using existing connections for routing additional client traffic minimizes the cost function for any new connection (i.e. less signaling is required), while at the same time optimizing network resource utilization. In the scope of a dynamic routing scheme, using active lightpaths instead of establishing new lightpaths is accomplished through assigning a lower cost to active paths in the scope of the route calculation process.

Additional resilience mechanisms are another QoS feature supported by ASON networks. As far as resilience at the optical layer is concerned, ASON/GMPLS networks allow for establishing alternative backup paths through signaling, which is a much more cost effective option than conventional 1+1 protection used in SONET/SDH networks. Apart from cost effective single-layer network resilience, ASON networks can also provide multi-layer resilience by allowing traffic to be rerouted at the client subnetworks, in cases where recovery at the optical layer is not possible [90, 91]. This is accomplished since ASON networks operate as server backbone networks serving multiple client subnetworks (Fig. 6). As shown in Fig. 6, an ASON network is likely to carry traffic from multiple client networks (e.g., MPLS, ATM, IP). Note that some of these client networks are likely to implement QoS mechanisms based on the building blocks concept. ASON QoS mechanisms boost end-to-end provisioning of QoS services.

As a result, ASON networks and their supporting control mechanisms (e.g., GMPLS) provide several QoS mechanisms. The majority of these mechanisms are directly derived from traditional IP control algorithms and protocols. Moreover, most of these mechanisms are readily available for deployment. Toward a wider end-to-end scenario, interworking issues between client and optical backbone networks need to be studied. Solving interworking issues will be greatly facilitated by the pertinence of optical control protocols to IP (e.g., MPLS/GMPLS, RSVP/RSVP-TE) [92].

CONCLUSIONS

After more than a decade of active research on IP QoS mechanisms, there are still numerous issues to be addressed. Overprovisioning solutions employed in the backbone networks seem to have limitations when it comes to supporting the wave of emerging mission-critical IP applications, such as VoIP. At the same time the evolution of IP-based infrastructures into large-scale networks comprising numerous IP and non-IP segments poses new challenges. Addressing these challenges demands that additional non-packet-level QoS mechanisms are deployed in the respective segments. A prominent example is the convergence of 3G networks with the Internet, toward providing Internet access to roaming users. No single global end-to-end framework can provide QoS to large-scale heterogeneous IP networks, as envisioned in the early days of QoS research. An appropriate blending of packet-level, but also non-packet-level mechanisms, is required to offer QoS.

Packet-level QoS mechanisms can be applied in various ways in the scope of IP/ATM/MPLS-based segments. Conventional QoS frameworks, such as IntServ, DiffServ, and ATM, suggest ways for providing QoS in these segments. Most importantly, these frameworks identify the main building blocks of a QoS solution, namely admission control, shaping and policing, signaling and resource management, queuing and scheduling, congestion control and queue management, QoS routing, QoS policy management, and QoS pricing. Based on these building blocks, several other combinations (i.e. different from those suggested in standard frameworks) are possible toward customized QoS solutions. The nature and objectives of these solutions will in most cases drive the combination of building blocks. Specifically, in combining the building blocks, network engineers must take into account:

- The architecture, resources, and technologies of the network.
- The requirements of the network users and applications, as well as of the organizations running the network.
- The capabilities of the network elements of the network with respect to QoS support.
- The target complexity and cost of the solution.

Fortunately, there is a host of research results and mechanisms for each one of these building blocks, as we discussed earlier. These results constitute a sound basis for applying and combining the various building blocks in the scope of operational networks. Moreover, they provide a foundation for further research into improving the performance and efficiency of IP mechanisms, in line with the “building blocks” approach.

In segments where IP QoS control is not applicable, there is a need to deploy other QoS solutions, depending on the technology of the particular segment. 3GPP2/UMTS standards provide support to wireless/mobile users accessing the IP network. At the same time, backbone optical infrastructures are augmented with control planes toward enhancing QoS support in the scope of all optical backbone networks. It is therefore important that IP QoS research takes into account these non-packet-level mechanisms, so as to boost development of interworking solutions where appropriate. By and large, improving the performance of QoS mechanisms, while also solving scalability and deployment problems, seems to be a key prerequisite for their penetration in the ubiquitous IP networks.

REFERENCES

[1] V. Firoiu *et al.*, “Theories and Models for Internet Quality of Service,” *Proc. IEEE*, special issue on Internet Technology, Aug. 2002.

[2] C. Aurrecoechea, A. Campbell, and L. Hauw, “A Survey of QoS Architectures,” *Multimedia Systems Journal*, vol. 6, no. 3, pp. 138-151, May 1998.

[3] J. Ash *et al.*, “AT&T’s MPLS OAM Architecture, Experience, and Evolution,” *IEEE Commun. Mag.*, vol. 42, no. 10, Oct. 2004.

[4] C. Fraleigh, F. Tobagi, and C. Diot, “Provisioning IP Backbone Networks to Support Latency Sensitive Traffic,” *Proc. IEEE INFOCOM 2003*, Apr. 2003.

[5] K. Czajkowski *et al.*, “SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems,” D. G. Feitelson *et al.* (Eds.): JSSPP 2002, LNCS 2537, 2002, pp. 153–83.

[6] I. Faynberg and H.-L. Lu, “An Architectural Framework for Support of QoS in Packet Networks,” *IEEE Commun. Mag.*, vol. 41, no. 6, June 2003, pp. 98–105.

[7] S. Minzer, “Broadband ISDN and Asynchronous Transfer Mode (ATM),” *IEEE Commun. Mag.*, vol. 27, no. 9, Sept. 1989.

[8] J. Hui, “Resource Allocation for Broadband Networks,” *IEEE JSAC*, vol. 6, no. 9, 1988., pp. 1598–1608.

[9] R. Guerin, H. Ahmadi, and M. Naghshineh, “Equivalent Capacity and its Application to Bandwidth Allocation in High-speed Networks,” *IEEE JSAC*, vol. 9, no. 7, 1991, pp. 968–81.

[10] A. Elwalid and D. Mitra, “Effective Bandwidth of General Markovian Sources and Admission Control of High Speed Networks,” *IEEE/ACM Trans. Net.*, vol. 1, no. 3, 1993, pp. 329–343.

[11] A. Romanow and S. Floyd, “Dynamics of TCP Traffic over ATM Networks,” *IEEE JSAC*, vol. 13, no. 4, 1995, pp. 633–41.

[12] D. Clark, S. Shenker, and L. Zhang, “Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism,” *Proc. SIGCOMM’92*, Aug. 1992, pp. 14–26.

[13] J. Hyman, A. Lazar, and G. Pacifici, “Real-Time Scheduling with Quality of Service Constraints,” *IEEE JSAC*, vol. 9, no. 9, 1991, pp. 1052–63.

[14] J. Cosmas *et al.*, “A Review of Voice, Data and Video Traffic Models for ATM,” *European Trans. Telecommun.*, vol. 5, no. 2, pp. 139–54, Mar.–Apr. 1994.

[15] W. Leland *et al.*, “On the Self-Similar Nature of Ethernet Traffic (extended version),” *IEEE/ACM Trans. Net.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[16] V. Paxson and S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Trans. Net.*, vol. 3, no. 3, June 1995.

[17] J. Roberts, “Traffic Theory and the Internet,” *IEEE Commun. Mag.*, vol. 39, no. 1, Jan. 2001, pp. 94–99.

[18] S. Floyd and V. Paxson, “Difficulties in Simulating the Internet,” *IEEE/ACM Trans. Net.*, vol. 9, no. 4, Aug. 2001, pp. 392–403.

[19] C. Barakat *et al.*, “Modeling Internet Backbone Traffic at the Flow Level,” *IEEE Trans. Sig. Proc.* (Special Issue on Networking), Aug. 2003.

[20] P. Salvador, R. Valadas, and A. Pacheco, “Multiscale Fitting Procedure using Markov-modulated Poisson Processes,” *Telecommun. Sys. J.*, vol. 23, June 2003, pp. 123–48.

[21] J. Soldatos *et al.*, “Enforcing Effective Rates for Packet-Level QoS Control in IP Networks: Theory and Validation based on Real Traffic Data,” *Telecommun. Sys. J.*, Kluwer, ISSN 1018-4864, vol. 27, Sept. 2004, pp. 9–31.

[22] H. Perros and K. Elsayed “Call Admission Control Schemes: A Review,” *IEEE Commun. Mag.*, special issue on Congestion Control, Nov. 1996, pp. 82–91.

[23] K. Shiomoto, N. Yamanaka, and T. Takahashi, “Overview of Measurement-based CAC Methods in ATM Networks,” *IEEE Commun. Surveys*, 1st Quarter, 1999.

[24] Z. Duan *et al.*, “A Core Stateless Bandwidth Broker Architecture for Scalable Support of Guaranteed Services,” *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 2, Feb. 2004, pp. 167–82.

[25] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm,” *Proc. ACM SIGCOMM ’89*, pp. 3–12.

[26] A. Parekh and R. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case,” *IEEE/ACM Trans. Net.*, vol. 2, no. 2, 1994, pp. 137–50.

[27] D. Ferrari and D. Verma, “A Scheme for Real-Time Channel Establishment in Wide-Area Networks,” *IEEE JSAC*, vol. 8, no. 4, 1990, pp. 368–79.

[28] L. Zhang, "VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks," *ACM Trans. Comp. Sys.*, vol. 9, no. 2, pp 101-124, May 1991.
[29] J. Bennett and H. Zhang, "Hierarchical Packet Fair Queuing Algorithms," *IEEE/ACM Trans. Net.*, vol. 5, Oct. 1997, pp. 675–89.
[30] I. Stoica, S. Shenker, and H. Zhang, "Core-Stateless Fair Queuing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks," *Proc. ACM SIGCOMM'98*.
[31] S. J. Golestani, "A Stop and Go Queuing Framework for Congestion Management," *Proc. ACM SIGCOMM'90*, pp. 8–18, 1990.
[32] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate Controlled Servers for Very High-Speed Networks," *Proc. GLOBECOM'90*, 1990, pp 300.3.1–300.3.9.
[33] D. Verma, H. Zhang, and D. Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network," *Proc. TriCom '91*, 1991, pp 35–43.
[34] H. Zhang and S. Keshav, "Comparison of Rate-Based Service Disciplines," *Proc. SIGCOMM '91*, 1991, pp 113–21.
[35] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks," *IEEE/ACM Trans. Net.*, Aug. 1995, pp. 365–86.
[36] I. Stoika, H. Zhang, and T. Ng, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services," *Proc. ACM SIGCOMM'97*, Sept. 1997.
[37] S. Vegesna, *IP Quality of Service*, ISBN 1578701163, Cisco Press, 2001.
[38] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *Proc. ACM SIGCOMM'99*, Sept. 1999.
[39] I. Stoica, S. Shenker, and H. Zhang, "Core-Stateless Fair Queuing," *IEEE/ACM Trans. Net.*, vol. 11, no. 1, Feb. 2003, pp. 33–46.
[40] I. Stoica, "Stateless Core: A Scalable Approach for Quality of Service in the Internet," Ph.D. Thesis, Carnegie Mellon Univ., Pittsburgh, PA, 2000.
[41] N. Christin and J. Liebeherr, "Architecture for Quantitative Service Differentiation," *IEEE Commun. Mag.*, vol. 41, no. 6, June 2003, pp. 38–45.
[42] G. Nong and M. Hamdi, "On the Provision of Quality-of-Service Guarantees for Input Queued Switches," *IEEE Commun. Mag.*, vol. 38, no. 12, Dec. 2000, pp. 62–69.
[43] C. Minkenberg and T. Engbersen, "A Combined Input and Output Queued Packet-switched System based on PRIZMA Switch-on-a-chip Technology," *IEEE Commun. Mag.*, vol. 38, no. 12, Dec. 2000, pp. 70–77.
[44] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net.*, Aug. 1993.
[45] C. Hollot *et al.*, "A Control Theoretic Analysis of RED," *Proc. IEEE Infocom 2001*, Anchorage, Alaska, Apr. 2001, pp. 1510–19.
[46] C. Hollot *et al.*, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows," *Proc. IEEE Infocom 2001*, Anchorage, Alaska, Apr. 2001, pp. 1726–34.
[47] X. Deng *et al.*, "A Control Theoretic Approach for Designing Adaptive Active Queue Management Schemes," *Proc. IEEE GlOBECOM'03*, San Francisco, CA, Dec. 2003.
[48] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 18, no. 4, 1988, pp. 314–29.
[49] M. Mathis *et al.*, "TCP Selective Acknowledgment Options," IETF RFC 2018 Oct. 1996.
[50] C. Jin *et al.*, "FAST TCP: From Theory to Experiments," http://netlab.caltech.edu/FAST/, 2003.
[51] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-delay Product Networks," *Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, pp. 89–102.
[52] R. Rejaie, M. Handley, and D. Estrin, "Layered Quality Adaptation for Internet Video Streaming," *IEEE JSAC* vol. 18, no. 12, 2000, pp. 2530–43.
[53] Q. Ma and P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees," *Proc. IEEE Int'l. Conf. Network Protocols*, Atlanta, GA, Oct. 1997.
[54] Q. Zhu, M. Parsa, and J.J. Garcia-Luna-Aceves, "A Source-Based Algorithm for Delay-Constrained Minimum-Cost Multicasting," *Proc. IEEE INFOCOM '95*, Boston, MA, Apr. 1995.
[55] S. Chen and K. Nahrstedt, "On Finding Multi-Constrained Paths," *Proc. IEEE ICC '98*, June 1998.
[56] B. Quoitin *et al.*, "Interdomain Traffic Engineering with BGP," *IEEE Commun. Mag.*, vol. 41, no. 5, May 2003, pp. 122–128.
[57] G. Apostolopoulos *et al.*, "QoS Routing: A Performance Perspective," *Proc. SIGCOMM '98*, Vancouver, Ontario, Canada, Sept. 1998.
[58] Z. Wang, and J. Crowcroft, "QoS Routing for Supporting Resource Reservation," *IEEE JSAC*, Sept. 1996, vol. 14, no. 7, 1228–34.
[59] I. Cidon, R. Rom, and Y. Shavitt, "Multi-Path Routing Combined with Resource Reservation," *Proc. IEEE INFOCOM '97*, Kobe, Japan, Apr. 1997, pp. 92–100.
[60] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Two Distributed Algorithms for Multicast Multimedia Communication," *Proc. ICCCN '93*, San Diego, CA, June 1993, pp. 343–49.
[61] X. Xiao *et al.*, "Traffic Engineering with MPLS in the Internet," *IEEE Network*, vol. 14, no. 2, 2000, pp. 28–33.
[62] D. O. Awduche, "MPLS and Traffic Engineering in IP Networks," *IEEE Commun. Mag.*, vol. 37, no. 12, 1999, pp. 42–47.
[63] G. N. Rouskas and I. Baldine, "Multicast Routing with End-to-End Delay and Delay Variation Constraints," *IEEE JSAC*, vol. 15, no. 3, 1997, pp. 346–56.
[64] H. Salama, D. Reeves, and Y. Viniotis, "Evaluation of Multicast Routing Algorithms for Real-Time Communication on High-Speed Networks," *IEEE JSAC*, vol. 15, no. 3, 1997, pp. 332–45.
[65] K. G. Shin and C.-C. Chou, "A Distributed Route-Selection Scheme for Establishing Real-Time Channel," *Proc. 6th IFIP Int'l Conf. High Performance Networking (HPN'95)*, 1995, pp. 319–29.
[66] S. Sanchez-Lopez *et al.*, "PNNI-based Control Plane for Automatically Switched Optical Networks," *IEEE J. Lightwave Tech.*, vol. 21, issue 11, 2003, pp. 2673–82.
[67] J. Moffett and M. Sloman, "Policy Hierarchies for Distributed Systems Management," *IEEE JSAC*, vol. 11, no. 9, 1993, pp. 1404–41.
[68] L.Delgrossi and D. Ferrari, "Charging Schemes for Reservation-based Networks," *Telecommun. Sys.*, vol. 11, issue 1, 1999.
[69] M. Koutsopoulou *et al.*, "Charging, Accounting and Billing Management Schemes in Mobile Telecommunication Networks and the Internet," *IEEE Commun. Surveys and Tutorials*, 1st Quarter 2004, vol. 6, no. 1.
[70] L. A. DaSilva, "Pricing for QoS-Enabled Networks: A Survey," *IEEE Commun. Surveys*, 2nd Quarter 2000, vol. 3, no. 2.
[71] A. M. Odlyzko, "Paris Metro Pricing: The Minimalist Differentiated Services Solution," *Proc. 7th Int'l. Wksp. Quality of Service (IWQoS '99)*, IEEE, 1999, pp. 159–61.
[72] A. Gupta *et al.*, "Priority Pricing of Integrated Services Networks," *Internet Economics*, L. W. McKnight and J. P. Bailey, Eds., Cambridge, Massachusetts, 1997, MIT Press, pp. 323–52.
[73] F. Kelly, "On Tariffs, Policing, and Admission Control for Multiservice Networks," *Operations Research Letters 15*, 1994.
[74] M. Falkner, M. Devetsikiotis, and I. Lambadaris, "An Overview of Pricing Concepts for Broadband IP Networks," *IEEE Commun. Surveys*, 2nd Quarter 2000, vol. 3, no. 2.
[75] B. Stiller and P. Reichl, "Pricing and Cost Recovery for Internet Services: Practical Review, Classification, and Application of Relevant Models" *Netmonics: J. Economic Research and Elec. Net.*, vol. 3, issue 2, Sept. 2001.
[76] L. Murphy, J. Murphy, and J. MacKie-Mason, "Feedback and Efficiency in ATM Networks," *Proc. IEEE ICC*, Ilkley, England, July 1995, pp. 378–93.
[77] X. Xiao *et al.*, "A Practical Approach for Providing QoS in the Internet Backbone," *IEEE Commun. Mag.*, vol. 40, no. 12, 2002, pp. 56–62.
[78] J. Evans and C. Filsfils, "Deploying Diffserv at the Network Edge for Tight SLAs, Part 2," *IEEE Internet Computing*, Mar./Apr. 2004, pp. 2–10.
[79] D. Chalmers and M. Sloman, "A Survey of Quality of Service in Mobile Computing Environments," *IEEE Commun. Surveys and Tutorials*, 2nd Quarter 1999.

[80] H. Kaaranen *et al.*, *UMTS Networks (Architecture, Mobility and Services)*, Wiley, 2001.

[81] J. Langer and G. Larson, "CDMA2000 — A World View," Ericsson Review No. 03, 2001.

[82] TS25.401 v3.70 — R99, UTRAN Overall Description, available from www.3gpp.org

[83] S.R0079-0 v1.0, Support for End-to-End QoS — Stage 1 Requirements, 2004/06, available from www.3gpp2.org

[84] X.S0011-004-C v1.0, cdma2000 Wireless IP Network Standard: Quality of Service and Header Reduction, 2003/09, available from www.3gpp2.org

[85] R. Boutaba *et al.*, "Lightpaths on Demand: A Web-Services-Based Management System," *IEEE Commun. Mag.*, July 2004, vol. 42 no. 7, pp. 101–07.

[86] ITU-T Rec. 8080 "Architecture for the Automatic Switched Optical Network (ASON)."

[87] Eric Mannie (Editor), Generalized Multi-Protocol Label Switching (GMPLS) Architecture, draft-ietf-ccamp-gmpls-architecture-07.txt, Internet Draft, work in progress, May 2003.

[88] J. R. Boucher, "Traffic System Design Handbook," *Telecommunications Handbook Series*, IEEE Press, 1993.

[89] J. Comellas *et al.*, "Integrated IP/WDM Routing in GMPLS-based Optical Networks," *IEEE Network.*, vol. 17, no. 2, Mar./Apr. 2003, pp. 22–27.

[90] P. Demeester *et al.*, "Resilience in Multilayer Networks," *IEEE Commun. Mag.*, vol. 37, no. 8, 1999, pp. 70–76.

[91] G. Kylafas, J. Soldatos, and G. Pikramenos, "Multilayer Resilience Strategies for Automatic Switched Optical Networks," *Proc. Int'l. Conf. Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN'04)*, St. Petersburg, 2004.

[92] A. Banerjee *et al.*, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements," *IEEE Commun. Mag.*, Jan. 2001, pp.144–51.

Biographies

JOHN K. SOLDATOS (jsol@ait.edu.gr) was born in Athens, Greece in 1973. He obtained his Dipl-Eng. degree in 1996 and his Ph.D. in 2000, both from the electrical and computer engineering department of the National Technical University of Athens (NTUA). He has had an active role in several EU co-funded research projects. Specifically, he has actively participated in EXPERT AC-094, WATT AC-235, IMPACT AC-324, Chameleon EP 20597, CATCH-2004 IST-1999-11103, and LION IST-19990-11387 projects, and is now with Athens Information Technology (AIT), involved in the CHIL-FP6-506909 project. He has also consulted in many ICT projects for leading Greek enterprises (INTRACOM S.A, PEGASUS S.A, IBM Hellas S.A, OTE S.A, TEMAGON S.A). Dr. Soldatos has extensively lectured in NTUA and AIT, and he has given numerous invited lectures. As a result of his activities he has co-authored more than 45 papers published in international journals and conference proceedings. Since March 2003 he has been with AIT, where he is currently an assistant professor. His current research interests are in broadband traffic control, network management, grid computing, and ubiquitous and autonomic computing.

EVANGELOS K. VAYIAS [M'98] (evag@intracom.gr) received the Dipl.-Eng. and Ph.D. degrees, both from the department of electrical and computer engineering, National Technical University of Athens, Greece, in 1996 and 2000, respectively. He is now a project consultant with Intracom SA, network professional services department, specializing in next-generation IP-based network technologies. Prior to this, he was a research associate with the Telecommunications Laboratory of the National Technical University of Athens, where he worked in numerous research projects in the areas of resource management and performance evaluation of broadband networks, distributed object-oriented technologies for network management, interactive multimedia services, and grid computing. He also consulted for several public and private organizations in Greece. He is a member of the Technical Chamber of Greece and a Cisco Certified Internetwork Expert.

GEORGE KORMENTZAS (gkorm@aegean.gr) is currently a lecturer at the University of the Aegean, department of information and communication systems engineering. He received the Diploma in electrical and computer engineering and the Ph.D. in computer science, both from the National Technical University of Athens (NTUA), Greece, in 1995 and 2000, respectively. From 2000 to 2002 he was a research associate with the Institute of Informatics & Telecommunications of the Greek National Center for Scientific Research "Demokritos," working in a number of EU funded research projects in the area of high-speed wired and wireless networks. His research interests are in the fields of traffic analysis, network control, resource management, and quality of service in broadband networks. He has more than 40 publications in books, journals, and international conference proceedings in the above areas. He is a member of IEEE, IFIP, and of the Technical Chamber of Greece.