Integration of Call Signaling and Resource Management for IP Telephony

Pawan Goyal, Albert Greenberg, Charles R. Kalmanek, William T. Marshall, Partho Mishra, Doug Nortz, and K. K. Ramakrishnan
AT&T Laboratories

Abstract

IP telephony presents a tremendous opportunity to service providers to offer both traditional services as well as a range of creative new services. However, there are substantial challenges to be faced in supporting a resource management framework that is adequate for telephony, and in providing a signaling architecture that enables these services while preserving user privacy and preventing theft of service. This article describes the Distributed Open Signaling Architecture, a framework for call signaling and resource management that meets these needs. A key contribution of our work is a recognition of the need for coordination between call signaling, which controls access to telephony-specific services, and resource management, which controls access to network-layer resources. We evaluate one approach to resource management in the backbone, consistent with our architecture, using signaling for aggregates of flows. Using traces from calls on the AT&T long distance network, we show that the multiplexing gains achieved by such aggregation can achieve most of the benefits of per-flow signaling, while avoiding its overheads. We also evaluate scheduling algorithms in order to understand their effect on the end-to-end delay experienced by voice packets.

The advent of IP telephony gives service providers a tremendous opportunity to offer both traditional services as well as a range of creative new services that integrate voice communications with data applications and exploit the power of intelligent end terminals. While services will continue to be provided through conventional telephones, the potential interaction with the network will increase through enhanced user interfaces.

At the same time, IP telephony poses substantial challenges. The signaling architecture must support both traditional services and new services enabled by intelligent end-points, while preserving user privacy and preventing service theft. User expectations of quality impose stringent requirements on both network and signaling performance. Meeting the network performance requirements calls for a resource management framework that ensures adequate capacity for voice everywhere along the end-to-end path.

IP networks are evolving to provide multiple levels of service. The revenue models may be different for each service, and a provider may wish to derive additional revenue from an enhanced service such as telephony. Because users will have access to multiple communication capabilities, the provider needs to find incentives to use and pay for these enhanced services. Otherwise, users are likely to bypass them by using protocols and functionality implemented in the end systems.

A provider can offer at least three fundamental incentives to encourage the use of an enhanced service as opposed to a lower priced, best-effort data service:

- Dependable quality of service (QoS) assurances. For telephony, these assurances are primarily in the form of low delay and low loss, to ensure that speech quality meets user expectations.
- Services that truly belong in the network, especially those that rely on a trusted intermediary.
- Functionality that is much more cost-effective, and easy to provide and use in the network.

We believe that a difference in network-layer QoS is essential to robust IP telephony service. As a result, the network must support resource reservations and admission control, allowing the resources available to the higher quality service to be managed. Even overload conditions must not lead to call defects, such as the called party answering and finding no resources to support communication between the two endpoints. Since the network is providing enhanced QoS, the signaling architecture must ensure that the use of enhanced service is authorized and accounted for.

This article describes the Distributed Open Signaling Architecture (DOSA), which incorporates call signaling and resource management functions to meet the needs of telephony. A key contribution of our work is a recognition of

0890-8044/99/$10.00 © 1999 IEEE

the need for coordination between *call signaling*, which controls access to telephony-specific services, and *resource management*, which controls access to network-layer resources. This coordination provides critical functions to ensure that:

- Users are authenticated and authorized before receiving access to the enhanced QoS associated with the telephony service.
- Network resources are available end to end before ringing the destination phone.
- Resource usage is properly accounted for, consistent with the semantics of telephony in which charging occurs only after the called party picks up.

The explicit coordination between call signaling and resource management is a key distinction between DOSA and other proposed signaling architectures for IP telephony, including the H.323 series of Recommendations [1], the Media Gateway Control Protocol (MGCP) [2], and the Session Initiation Protocol (SIP) [3].

In this article, we present the application requirements that drive the DOSA design, followed by a description of the architecture itself. Then we outline resource management mechanisms, consistent with DOSA, for meeting the QoS requirements of telephony service. We also present simulation results quantifying the performance of these mechanisms.

Application Requirements

The Distributed Open Signaling Architecture was designed to meet "first-line" telephone service requirements, comparable to circuit-switched telephone service. Because of the large embedded base, the service must support existing customer premises equipment, including conventional telephones and legacy voiceband data equipment such as fax machines and modems. It must also support the broad range of service features currently available. These include basic telephony features such as emergency 911 service, operator services, and law enforcement wiretapping, as well as popular optional features such as call forwarding, caller ID, call waiting, and three-way calling.

In addition, first-line telephone service has stringent requirements for speech quality, signaling performance, reliability, and scalability. Users have come to expect highly reliable and available telephone service. To be acceptable by the mass market as a circuit-switched replacement, the service must be available 99.9+ percent of the time. While reliability must be built into the network elements, it must also be designed into the network and signaling architectures.

First-line telephony performance requirements include:

- *Low delay*. End-to-end packet delay must be small enough not to interfere with normal voice conversations.
- *Low packet loss*. Packet loss must not perceptibly impact either voice quality or the performance of fax and voiceband modems.
- *Short post-dial delay*. The delay between the user dialing the last digit and receiving positive confirmation from the network must not perceptibly differ from post-dial delay in the circuit-switched network.
- *Short post-pickup delay*. The delay between a user picking up a ringing phone and the voice path being cut through must be short enough that the initial "hello" is not clipped.

These performance requirements affect the design of both the signaling and network architectures. In this article we concentrate primarily on resource management mechanisms necessary to ensure that voice receives adequate performance from the network layer.

Delay Metrics

Recommendation G.114 of the International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) provides guidelines on the tolerable delay for a normal telephone conversation, based on extensive testing carried out over three decades. The maximum one-way delay acceptable for most user applications is given as 150 ms, but the Recommendation notes that some highly interactive voice and data applications may experience degradation even at this point. Therefore, it is critical to manage delay carefully.

The 150-ms one-way delay limit must be allocated among several different sources, producing a *delay budget*. The single biggest component of the delay budget is backbone transport. The longest path in the continental United States is between Seattle, Washington, and Orlando, Florida. However, if we consider the possibility of facility failure along this path, the restoration path may be routed from Orlando to Los Angeles, to Chicago, to San Francisco, to Seattle (when the Los Angeles-to-San Francisco facilities are out of service). This worst-case one-way delay has been measured at 95 ms. This leaves a remaining delay budget of only 55 ms (one-way) for all other sources.[1]

Delays encountered in the transmission path can be characterized as fixed or variable. *Fixed delays*, such as propagation time, will be the same for every data packet. *Variable delays* or "jitter," such as queuing and media access, may vary from one packet to the next. Therefore, a playout buffer is required at the receiving side to "dejitter" the packet stream.

It is important to note that simply adding fixed and variable delay does not calculate total delay. The jitter must actually be counted twice in the delay budget, once as potential delay in the transport network and once at the playout buffer in the receiver. When the first packet of a conversation reaches the playout buffer, it must be delayed by an amount equal to the allocated jitter in the network before starting to play out voice. If that packet had accumulated zero jitter in transport, the buffer is now set correctly to handle all further packets whether they are jittered or not. If that first packet was delayed by the maximum jitter in transport, the playing out of voice is actually delayed by twice the jitter value.

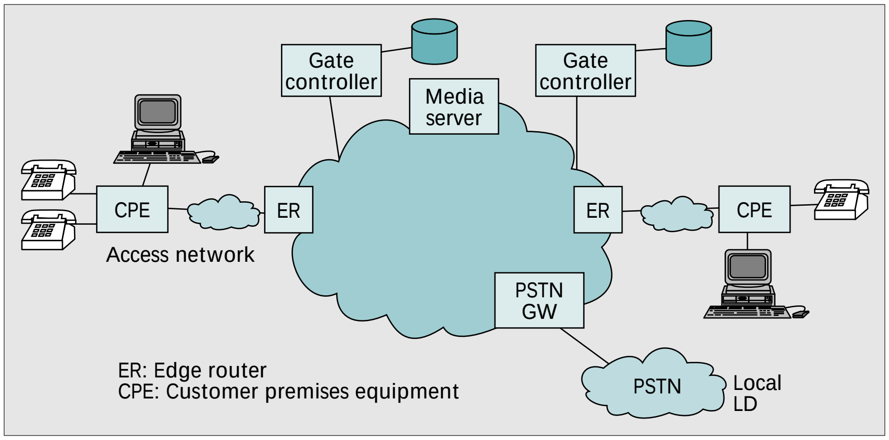
A representative delay budget can quantify the maximum queuing delay that can be introduced in the backbone. Consider the 150-ms one-way delay bound. Backbone propagation reduces the available delay to 55 ms. Allocate 10 ms to the speech coder, 10 ms to speech enhancement and silence suppression, and 5 ms to processing, propagation, and interleaving in the local access network. The remaining 30 ms is available for variable delay factors, which allows a 15-ms playout buffer at the receiver and a maximum 15-ms jitter tolerated in transit. Of this 15-ms jitter budget, we allocate 5 ms to the access network, leaving 10 ms for queuing delay in the backbone network. We return to the subject of backbone queuing delay later.

Packet Loss Metrics

Typically, the requirements for packet loss for encoded speech are 1 percent or less. While loss-concealment algorithms can be used to reproduce intelligible speech even with higher loss rates, the resulting performance is inadequate as a circuit-switched replacement.

Loss requirements for acceptable voiceband modem performance are even more stringent.

[1] *We have not considered international calls here, for which delay management is an even more important consideration.*



■ Figure 1. *Key components of the Distributed Open Signaling Architecture.*

Distributed Open Signaling Architecture

This section describes the key components of the DOSA architecture and uses a simple call flow to illustrate how these interact to support telephony service. While the DOSA signaling protocol supports service features implemented by the endpoints, the details of these feature flows are beyond the scope of this article.

Components

Figure 1 illustrates the key components in the architecture. Telephony clients, such as PCs or multimedia terminal adaptors, interface between a conventional telephone and a local area or access network. Telephony clients participate in end-to-end capability negotiation, call signaling, and resource reservation protocols. Edge routers (ERs) connect the access network to a managed IP backbone. In our architecture, ERs implement a function called a *gate*, which is a policy enforcement function that controls access to higher-quality service from the network layer. Gate controllers control the gates and provide service-specific functions, such as number translation, authorization, and feature support. Media servers are network-based components that play terminating announcements, perform audio bridging, and so on. Finally, public switched telephone network (PSTN) gateways interface to the PSTN.

For robust telephony service, the service provider must manage access to network-layer resources. Admission control is provided by trusted ERs at the boundary to the backbone network. The ERs support both resource-based and policy-based admission-control functions, allowing the IP network to ensure the bounded loss and delay required by telephony service.

ER-implemented gates support packet classification and policing to ensure that only IP flows authorized by the gate controller are granted access to enhanced QoS in the access and backbone networks. Gates allow a flow of packets from a specific IP source address and port to a specific IP destination address and port to receive enhanced QoS. Gates are *opened* for individual calls. Opening a gate involves an admission-control check, which is performed when a reservation request is received from the customer premises equipment (CPE) for an individual call. Opening a gate may involve reservation of resources in the network for the call, where and when necessary. Once a gate is opened, the ER participates in providing enhanced QoS service, by either marking the packets or queuing them as a particular QoS-assured flow.

ERs require authorization from a gate controller on a call-by-call basis before opening a gate. Thus, the ER can ensure that enhanced QoS is only provided for authorized end-to-end flows that have usage accounting mechanisms in place. Since ERs know about the resource usage associated with individual IP flows, they generate the accounting events that allow a user to be charged for service. Edge routers can also implement network address translation to support IP address privacy for both the called and calling parties.

Gate controllers provide the policy function that determines whether a gate should be opened. DOSA signaling sets up a gate in advance of a reservation request. This allows the gate controller to be "stateless" in that it does not need to know the state of calls already in progress. Because the gate is set up in advance, the gate controller provides an upper bound or *envelope* of acceptable traffic parameters to the ER, in addition to the source and destination addresses, port pairs, and protocol identifier. When the CPE makes a resource reservation request by signaling to the network, the ER admits the reservation if it is within the envelope specified in the gate setup.

Gate controllers implement a set of service-specific control functions required to support the telephony service:

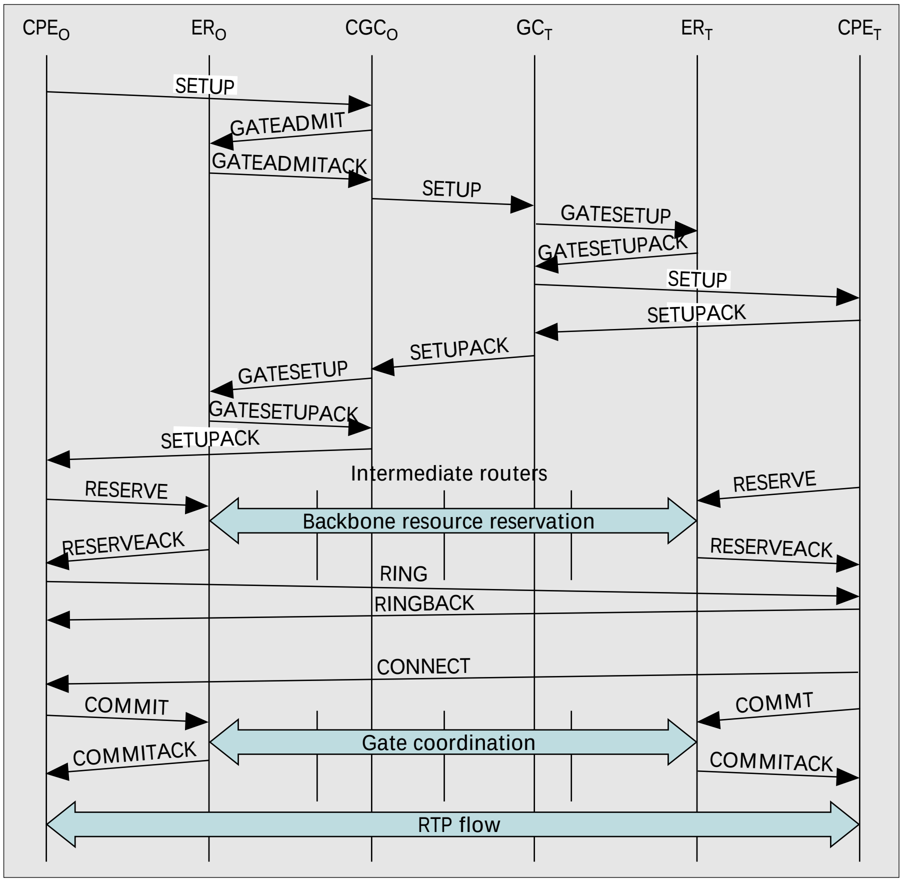
- *Authentication and authorization.* Gate controllers authenticate signaling messages and authorize requests for service on a call-by-call basis.
- *Number translation and call routing.* Gate controllers translate dialed E.164 numbers to a terminating IP address based on call routing logic to support a wide range of call features.
- *Service-specific admission control.* Gate controllers can implement a broad range of admission-control policies for the telephony service. For example, gate controllers may provide precedence for particular calls, such as 911 calls. Admission control can also be used to implement overload control mechanisms, for example, to restrict the number of calls to a particular location or to restrict the frequency of call setup to avoid signaling overload.
- *Signaling and service feature support.* Many service features are implemented in the CPE, but the gate controller also plays a support role. DOSA signaling provides a set of service primitives to endpoints that are mediated by the gate controller. The gate controller is involved in implementing service features that depend on the privacy of calling information, such as caller ID blocking). It also plays a role in supporting service features that require users to receive a consistent view of feature operation even when CPE is down.

The gate controller is designed as a stateless transaction server so that failure of a gate controller does not affect stable calls. Since the interactions with the gate controllers are stateless, there is no need for consecutive calls to be processed by the same gate controller. This design places the management of a call's state where it belongs: at the CPE. Only the CPE and the network elements along the data path are required in providing service for ongoing calls. We believe that this design makes the gate controller efficient and highly scalable, and the network architecture more reliable.

Simple Call Flow

Figure 2 shows the message exchanges associated with basic call setup, illustrating the relationship between the call signaling, resource management, and policy functions.

When a user goes off-hook and dials a telephone number, the originating CPE (CPE$_O$) collects the dialed digits and sends a SETUP message to the originating gate controller (GC$_O$). Note that providing service to untrusted endpoints requires call-by-call authorization as well as coordination with a policy enforcement function that controls access to the higher quality service. As a result, GC$_O$ verifies that CPE$_O$ is a valid subscriber of the telephony service (using authentication information in the SETUP message). GC$_O$ may perform service-specific admission control. Since only ER$_O$ knows the resources currently reserved by CPE$_O$, GC$_O$ queries it using a GATEADMIT request to ensure that the aggregate resources being used by CPE$_O$ do not exceed a provisioned limit. GC$_O$ does number translation and call routing, augments the SETUP message with accounting information associated with the caller, and forwards it to the terminating gate controller (GC$_T$).



■ Figure 2. *DOSA call flow.*

The GATESETUP message from GC$_T$ provides policy information to the terminating ER (ER$_T$) indicating that it has permission to open a gate for the IP flow associated with this phone call. This policy information is kept by ER$_T$ for use when it receives a resource management request from CPE$_T$. GC$_T$ then forwards the SETUP message to CPE$_T$ to notify it about the incoming call. CPE$_T$ sends a SETUPACK to GC$_T$, which forwards it to GC$_O$. GC$_O$ informs the originating ER (ER$_O$) that it has permission to open a gate for the IP flow associated with this phone call, and then GC$_O$ forwards SETUPACK to CPE$_O$. At this point, the role of the gate controllers in this setup is done, and the CPE and ERs proceed to the resource management phase.

DOSA partitions resource management into separate reserve and commit phases. At the end of the first phase, resources are reserved but not yet available to the CPE. At the end of the second phase (after the called party is rung and answers), resources are committed and made available to the CPE and recording is started so that the user can be billed for usage. The two-phase protocol is essential to the service requirements associated with telephony. It ensures that resources are available before ringing the far-end telephone, while ensuring that usage recording does not start until the far-end user picks up the phone. Backbone resources are reserved and allocated in the first phase to limit the impact on post-pickup delay.

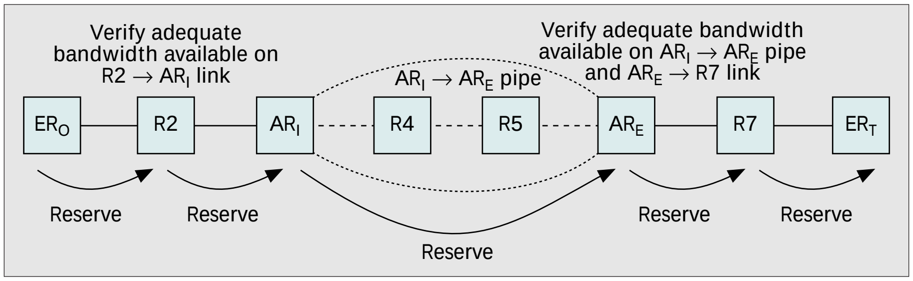
First, CPE$_O$ and CPE$_T$ issue RESERVE messages to ER$_O$ and ER$_T$, respectively. ER$_O$ and ER$_T$ perform admission control and, if successful, send a RESERVE_ACK to the respective CPE. As soon as CPE$_O$ knows that resources are available, it sends a RING message to CPE$_T$ instructing it to start ringing the phone. CPE$_T$ sends a RINGBACK message to CPE$_O$ indicating both that resources are available and that the RING message was received.

When the called party picks up the phone, CPE$_T$ sends a CONNECT message to CPE$_O$ and a COMMIT message to ER$_T$. CPE$_O$ sends a COMMIT message to ER$_O$ when it receives the CONNECT message. On receiving a COMMIT message, the ERs open their respective gates, making resources available for the call. In constrained-bandwidth access networks, this may involve an interaction with the layer-2 or layer-3 scheduling mechanisms. The COMMIT message also starts accounting for resource usage. To prevent some service theft scenarios, the ERs coordinate the opening of gates by exchanging gate coordination messages to ensure that both gates are opened at roughly the same time.

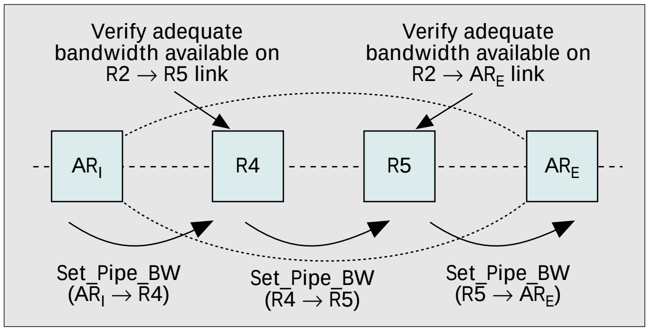
Backbone Resource Management

DOSA assumes that admission control and scheduling will be used in the backbone network to meet the QoS requirements for voice flows. Admission control is needed to ensure adequate capacity on the end-to-end path of each admitted voice call. Packet scheduling is required to ensure bounded end-to-end queuing delay for voice packets when the network capacity is shared with other classes of traffic having different traffic characteristics and QoS requirements. To support large-scale service, these techniques must manage resources on an aggregate basis. For example, it may not be feasible to implement per-flow queuing and scheduling for hundreds of thousands of active flows. In this section we describe admission-control and scheduling mechanisms, in concert with DOSA signaling, that do not require per-flow state in the core of the network. We also present simulation results that examine the performance of these mechanisms as measured by their ability to meet the delay requirements of voice flows, and the efficiency with which they utilize backbone network capacity.

The proposed admission-control framework requires per-flow signaling in the access network. The access network includes a small subset of routers starting from the ER, where the DOSA gates reside, up to an aggregation router (AR) that interfaces to a high-capacity backbone network. Aggregation routers deal with per-flow requests and determine whether they can be satisfied. The resource management in the backbone is for aggregate capacity required between these aggregation routers.



■ Figure 3. *Resource reservation and admission control with pipes.*



■ Figure 4. *Pipe bandwidth management.*

To avoid the need for per-flow admission-control checks in the backbone, we introduce the abstraction of a *pipe*. A pipe is a logical link of a given bandwidth between noncontiguous aggregation routers. Admission-control decisions for a voice flow are made at the ARs on either end of a pipe, without involving any of the backbone routers. Per-flow hop-by-hop signaling in the access network and per-pipe hop-by-hop signaling in the backbone network are used only for admission control, and do not instantiate any packet classifiers or scheduling state at intermediate routers. Thus, it may be possible to use pipes in combination with scheduling policies like those proposed for differentiated services (diff-serv)[4].

Figure 3 illustrates the operation of the proposed admission-control mechanism in conjunction with DOSA signaling. An originating ER forwards the DOSA RESERVE message from the originating CPE toward the destination CPE. This message carries the flow identifier tuple <IP_src_addr, IP_dst_addr, src_port, dst_port, protocol id> and the bandwidth required for the flow. The RESERVE is forwarded hop by hop along the end-to-end path until it reaches an ingress aggregation router AR_I. Each of the intermediate routers in the path to AR_I (for example, R2) performs an admission-control test to verify that adequate capacity is available for this flow on the next-hop link. Each router also marks itself as the previous hop before forwarding the RESERVE message. At AR_I, the format of the RESERVE message is modified before it is forwarded to *hide* these messages from the backbone routers.[2] This causes the RESERVE message to be forwarded through the backbone routers without being interpreted, until it reaches the egress aggregation router AR_E. AR_E determines the identity of AR_I using the previous-hop field, and executes an admission-control test to check if the new flow can be accommodated on the pipe between AR_I and AR_E.

If AR_E were to receive a RESERVE message for which it could not find a corresponding pipe, it would need to trigger the creation of a pipe by notifying AR_I. This requires AR_E to keep track of the available capacity (and per-flow state) for all the pipes that terminate on it. AR_E also modifies the format of the RESERVE message to no longer hide this message from the intermediate routers. The RESERVE is then forwarded hop by hop toward the destination, and is processed by the routers in the path, including ER_T.

There are alternative models that establish an explicit association between the ingress and egress ARs. For example, the information needed about AR_E can be integrated with the routing system [5, 6]. In this case, AR_I has sufficient information about the pipe to perform the admission-control test when it receives a RESERVE message. In either case, if the admission-control check fails, the call is blocked, and the appropriate DOSA RESERVE_ACK message is generated.

Aggregate QoS signaling may be used between the ingress and egress ARs to dynamically modify a pipe's capacity. This aggregate QoS signaling is processed hop by hop through the backbone, as shown in Fig. 4, and the backbone routers perform admission control and resource allocation for the aggregate capacity requested. The reservation messages for setting the pipe bandwidth in the backbone have to flow on the same path as the data.

The capacity needed on a pipe between ARs may be determined over multiple timescales. For example, the size of a pipe might be provisioned based on traffic forecasts. However, resizing a pipe's capacity dynamically allows better adaptation to the traffic fluctuations resulting from day-of-week or time-of-day effects. This can potentially reduce the aggregate capacity required in the backbone if all pipes do not see peak traffic at exactly the same time. Resizing the pipe even more frequently — on the order of minutes or even seconds — saves additional capacity through multiplexing gains across pipes on the backbone links. In a later section we present results illustrating that there are major capacity savings to be realized by resizing pipes frequently.

Scheduling

We used simulations to evaluate the effect of scheduling on the end-to-end delay behavior of voice flows. The scheduling framework evaluated in this article assumes that all voice flows are aggregated into a single FIFO queue, with priority or WF^2Q scheduling being used to partition bandwidth between the voice queue and other classes of traffic. This is similar to the diff-serv approach to IP QoS [4].

The topology used is shown in Fig. 5. In these simulations each *server* models the contention for transmission-link capacity at a router among the queued packets for active voice and data flows. At each server, all the voice flows are assigned to a single queue, while each individual data flow is assigned its own queue. Half of the active voice flows traverse the whole network, while the other half traverse only a single hop. All the data flows traverse only a single hop. Each server has a capacity of 10 Mb/s. (Nichols *et al.* [8] have carried out related experiments.)

Each voice flow is assumed to generate packets at a constant rate corresponding to a mean bandwidth of 64 kb/s. The number of active voice flows at each server is picked to ensure that the aggregate bandwidth of these flows is a fixed fraction *u* of the server capacity. The data sources are on-off sources with exponentially distributed on-off periods. The on and off periods were chosen to be 10 and 100 ms, respectively. We assume that all the data flows are statistically identical and

[2] *Several proposals have been made for mechanisms that can be used to hide reservation messages from being processed by backbone routers, in the context of supporting RSVP-based aggregated QoS signaling [7].*

determine the on rate such that the utilization due to data flows is 90 percent of the residual capacity: 0.9 (1 – u). We assume that there are four active data flows at each server.

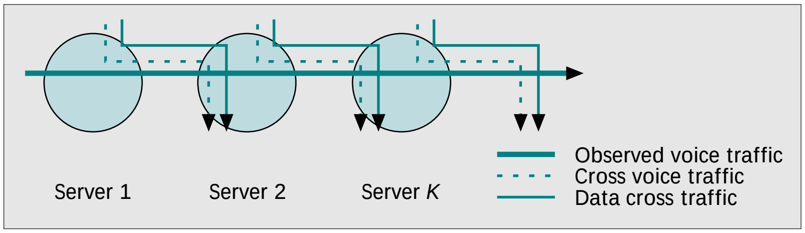
Since the voice sources generate packets at a constant rate, the phases of these sources (i.e., the time instants at which they generate packets) is important. In the simulations we consider two extreme scenarios:

- *Synchronized*: All voice sources generate packets periodically at exactly the same time instants.
- *Randomized*: Each voice source generates packets periodically, but at time instants that are randomized with respect to other voice sources.

In practice, the degree of synchronization between CBR voice sources is likely to lie between these two extremes.

We use two scheduling algorithms:

- *Priority scheduling*: In this case, the voice queue is given nonpreemptive priority over all the data queues. The data queues are scheduled using the WF^{2}Q+ fair scheduling algorithm [9].
- *Fair scheduling*: In this case, all the queues (including the voice queue) are scheduled using WF^{2}Q+. We look at three different alternatives for assigning weights to the voice queue. Specifically, if the voice queue utilization was u, we assign it a weight such that this queue is guaranteed a minimum bandwidth of uC, 1.5 uC, and 2.0 uC where C is the capacity of the server.



■ Figure 5. *The network topology for simulation of scheduling policies.*

We have conducted experiments with several different packet sizes for voice (64 bytes and 100 bytes) and data (64, 512, and 1024 bytes). The results presented in this article assume that the voice packets are 100 bytes and data packets 512 bytes.

For voice flows, the key metric of performance is the end-to-end delay behavior. Figure 6 plots the variation in the 99.9th percentile of the end-to-end delay as a function of the number of hops for the two scheduling algorithms, when the phase differences between the voice sources are randomized. We make the following observations:

- The bounds for queuing delay in the backbone (10 ms) for voice packets can be met as long as there are less than 10 hops on the end-to- end path.
- The delay increases linearly with the number of hops regardless of the scheduling algorithm.
- Priority scheduling provides the smallest delay.
- As voice utilization increases, the distinction between priority and WF^{2}Q+ reduces.
- Increasing the weight assigned to the voice queue for a particular voice utilization level causes the delay behavior obtained with WF^{2}Q+ to approach that provided by priority scheduling.

Figure 7 plots the variation in the 99.9th percentile of end-to-end delay with number of hops for the two scheduling algorithms when the voice sources are synchronized. Figure 7a (at 20 percent voice utilization) illustrates that, as in the previous case, priority scheduling provides the smallest delay, and the delay behavior of priority scheduling can be approached by assigning a large weight for the voice class. However, unlike the previous case, the delay does not increase linearly with the number of hops. This may be explained as follows.

First, consider the case where the weight is assigned such that the bandwidth of the voice queue is equal to the aggregate incoming rate (i.e., weight = 1.0 uC). For this case, the interpacket spacing for the voice flow is 100*8/(64 kb/s) = 12.2 ms, and all the flows generate a packet at the same time instant. This causes an accumulation of voice packets at each of the servers in the network. This built-up queue will take 12.2 ms to drain. Consequently, the packets experience close to 12.2 ms of queuing delay, most of which occurs at the first two hops. By that time, the queues built up at the subsequent hops have drained. Thus, the packets will experience little additional queuing delay in subsequent hops. When the number of hops gets large enough, the end-to-end delay grows beyond the 12.2 ms. At this time, a new burst of packets are generated at the sources. These new bursts of packets cause yet another standing queue through which packets will have to flow, thus increasing the delay of all the packets again by approximately half the queue drain time on average. A further increase in the number of hops causes the queuing delay to go up by the same amount again, as seen by the third step in Fig. 7a.

Now, consider the case where a weight of 1.5 uC is assigned to the voice queue for the WF^{2}Q+ scheduler. In this case, the queue drain time reduces to 12.2/1.5 = 8.1 ms. Hence, the packets build up their end-to-end delay to the queue drain time of 8.1 ms quickly. However, since the interpacket spacing continues to remain at 12.2 ms, and the queuing delay remains close to 8.1 ms, packets do not see a queue buildup due to new burst. Thus, the delay increases only linearly for networks larger than two hops. The nonlinear behavior for a weight of 2.0 uC and priority scheduling can be similarly explained. At higher utilization the behavior is qualitatively similar, as shown in Fig. 7b.

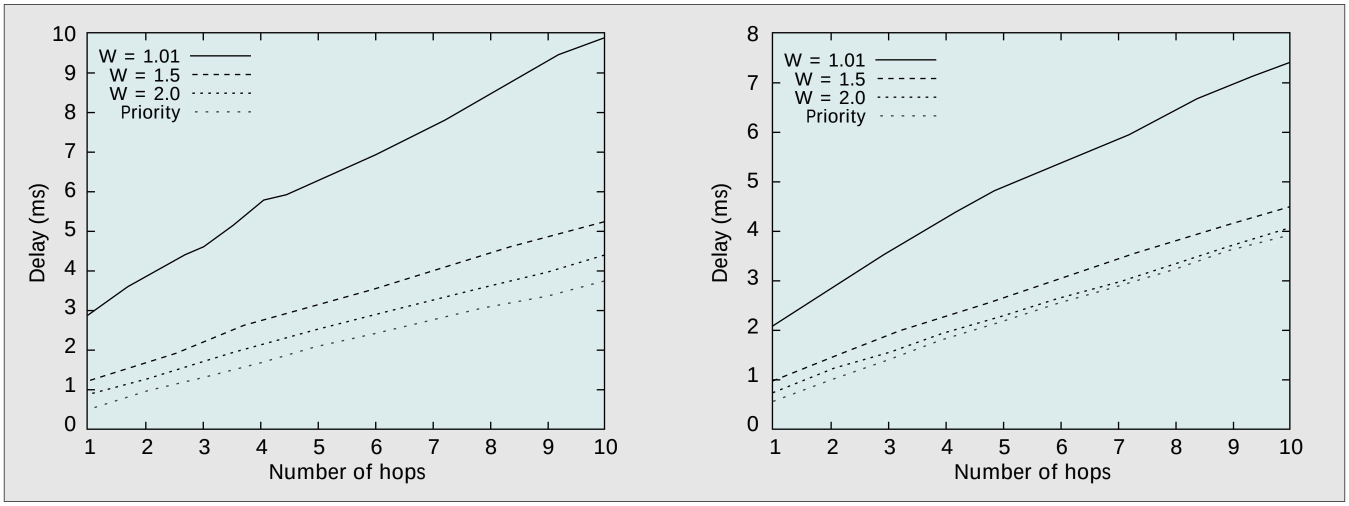
Observe that synchronization has less impact in the case of priority scheduling or fair scheduling with a large weight. We conclude that it is preferable to either have priority scheduling or give a large weight for the voice class, to avoid the effects of synchronization. Our observations were similar when the number of queues was increased to 20 and different packet sizes were used.

To summarize, our results on scheduling show that aggregation does not adversely affect the performance of voice transport. Handling all voice packets in a single FIFO queue meets the stringent delay requirements, as long as the scheduling policy is properly configured; that is, either the voice queue is given priority, or, when using weighted fair allocation, the voice queue is assigned a sufficiently large weight. Furthermore, the effect of synchronization of voice sources is limited.

Bandwidth Management

In this section we present experimental results that provide insight into the use of pipes for bandwidth management and capacity planning in the backbone network. In the simulations described here we only simulate the events corresponding to the arrival and departure of voice flows at every router, not the individual packet arrivals for each active voice flow.

We use configuration and usage data derived from AT&T networks in our simulations to ensure that our results are obtained in a realistic setting. We use the IP backbone for AT&T WorldNet (Fig. 8) to model the network topology. We use trace-driven simulations to model the arrival and departure processes for voice flows. These traces are derived from the call detail records (CDRs) corresponding to all of the telephone calls offered to the AT&T switched network in the



■ Figure 6. *99.9th percentile of end-to-end delay with randomized phasing between voice sources at a) 20% utilization; b) 40% utilization.*

domestic United States in the time period from August to December 1998. The CDRs enumerate the originating, dialed, and terminating numbers, along with the origination time and duration for each call.

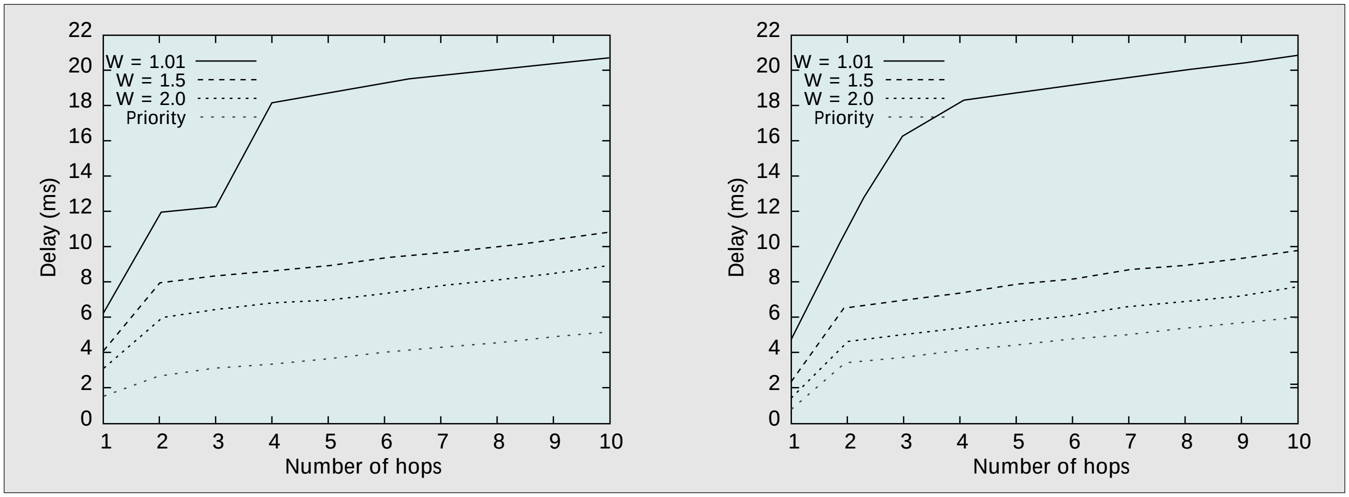
To determine the routing of each of these voice calls over the network topology shown in Fig. 8, we use the following rules. A given 10-digit telephone number is associated with an area code (based on the first three digits of the phone number). All of the traffic originating in an area code is assumed to be funneled through an access link to one of the 14 backbone routers geographically closest to the centroid of the geographical region corresponding to that area code. For each pair of area codes, two unidirectional pipes are established between the backbone routers into which each area code feeds its traffic. (If two area codes feed their traffic into the same router, this traffic is considered local and not simulated.) The pipes map to shortest path routes in the network topology, following normal IP intradomain routing.

Each individual simulation run corresponds to a trace-driven simulation for all calls placed on the AT&T switched network in a single day. Each call is simulated in chronological order. To simulate a call, we:

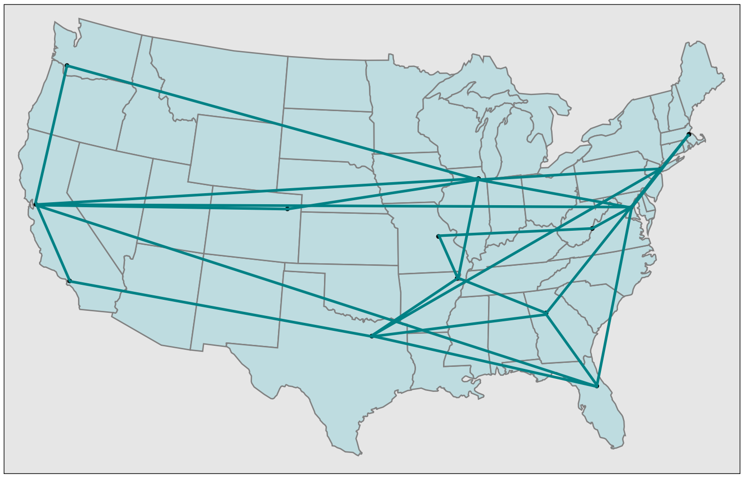
- Map the call to the corresponding pipe in the network topology
- At the call's arrival time, increment the count of active voice flows at each link in the network on the route for this pipe
- At the call's departure time, decrement the count of active voice flows for each of these links

During the course of a simulation, each pipe is resized periodically, with a period of *t* min. In practice, pipe resizing needs to be driven by a prediction of the number of voice flows expected to be carried on the pipe in the future. Specifically, the pipe size requested for the time interval [*kt*, (*k* + 1)*t*) for *k* = 0, 1, …, must be chosen as a function of the number of calls in progress at time *kt*, and of recent measurements of the call arrival and departure process for this pipe. In the experiments reported here, we assume perfect prediction; that is, the capacity requested for a pipe in [*kt*, (*k* + 1)t] is determined based on a-priori knowledge of the maximum number of voice flows which will be simultaneously active on that pipe during this interval. In related work, we have evaluated the use of realistic predictors (driven by measurements) and determined that the use of such predictors does not significantly modify the insights presented in this article.

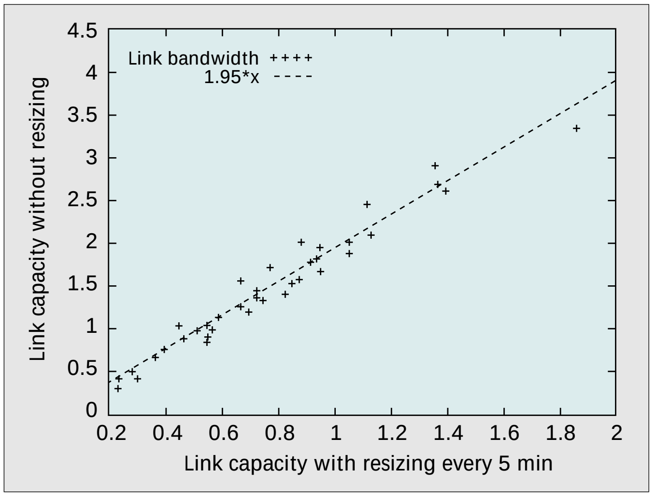
The key performance metric that is captured during each simulation run is the capacity requested for each pipe as a function of time. The aggregate capacity required on each network link (as a function of time) is then computed as the sum of the capacities required for all of the pipes routed over that link. We run multiple simulations to understand the effect of the pipe resizing intervals on the aggregate link capacity requirements. More precisely, let:



■ Figure 7. *99.9th percentile of end-to-end delay with synchronized phasing between voice souces at a) 20% utilization; b) 40% utilization.*



■ Figure 8. *AT&T WorldNet backbone topology.*



■ Figure 9. *Link capacities required with and without pipe resizing.*

- *Pipe-Size*(p, t, k) denote the size of a pipe p, in interval [kt, (k + 1)t), assuming a resizing period of t.
- *Link-Size*(l, t) denote the maximal bandwidth reserved on link l over the entire day, that is, $\max_k \Sigma_{p \text{ routed on } l}$ Pipe-Size(p, t, k).
- *Link-Size*(l, 24) denote the maximal bandwidth reserved on link l when the resizing period equal to the whole day (t = 24 hr). This is the capacity needed on the link if pipes were provisioned with a fixed capacity and never resized during a day.
- *Link-Size*(l, 0) denote the maximal bandwidth reserved on link l when the resizing period is extremely small. *Link-Size*(l, 0) is therefore just the maximal number of calls in progress on link l during the entire duration of the simulation.

Let *Gain*(l, t) = *Link-Size*(l, 24)/*Link-Size*(l, t), measure the capacity savings on link l due to pipe resizing with period t. Finally, define *Gain*(t) as Σ_l *Link-Size*(l, 24)/Σ_l *Link-Size*(l, t), to measure the networkwide capacity gain due to pipe resizing. *Gain(0)* represents the capacity savings that would be achieved if pipes were resized for every call. Thus, comparing *Gain(t)* to *Gain*(0) is interesting and useful.

Figure 9 plots the link capacity required without resizing (*Link-Size*(l, 24)) versus the link capacity required with resizing every 5 min (*Link-Size*(l, 5/1440)), for all links l, for a typical simulation run (September 28, 1998). Note that the gain is fairly uniformly about 2 across all links. Thus, resizing pipes once every 5 min saves a factor of about two in capacity on all links. In a large backbone network, this represents enormous capital savings on transmission facilities.

Figure 10 depicts the value of *Gain(t)* as a function of the resizing period t for five weekdays. The maximal gain, *Gain*(0), appears to be about 2.5. Again, for a resizing period of about 5 min, the gain is near 2 for all five days.

We searched the data set for days whose results contradict Fig. 10, but were unable to find any such results, or even any significant variation. For example, on August 31, 1998, the U.S. stock market dropped, and although the peak call volume increased by 20 percent, the results (i.e., the gains) did not change. Weekend data shows essentially the same results, although the variability is slightly greater and the usage volumes somewhat less.

There are two possible explanations for why pipe resizing provides such large capacity gains. One possibility is that deterministic *time-of-day* fluctuation in the calling loads in different geographical areas causes each of the pipes routed over a given link to hit its maximal load at different times during the day. The second possibility is that due to the normal statistical fluctuations in the call arrival process, the aggregate capacity required for a pipe varies over time. Pipe resizing can provide capacity gains due to either kind of traffic fluctuation since it allows better multiplexing of the link capacity between pipes. Analysis of the simulation results reveals that at particularly busy periods (e.g., 4:00 p.m. EST), *all* 42 links are very close to their daily peaks (this is not surprising since both coasts are in the midst of the workday at 4:00 PM EST). Thus, the dominant reason for the effectiveness of pipe resizing appears to be statistical fluctuations.

In our experiments there are 168 area codes, and therefore about 30,000 pipes are being multiplexed on 42 (unidirectional) links. In related work, we have examined the effectiveness of pipe resizing when the degree of traffic aggregation is much greater (i.e., there are a much small number of pipes). These results suggest that pipe resizing is still useful as the degree of traffic aggregation increases. However, the capacity savings are lower because there is less potential for statistical multiplexing across pipes.

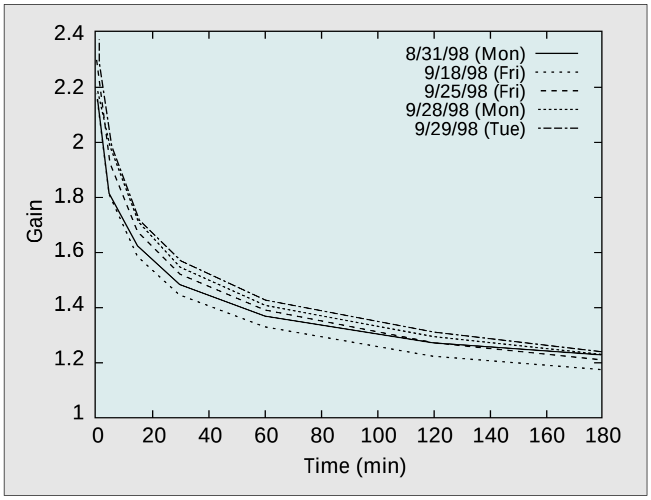
The use of pipe resizing requires backbone routers to interpret signaling messages. In our experiments the total signaling traffic for the entire network generated due to pipe resizing is on the order of a few hundred messages per second, assuming a resizing interval of 5 min. In contrast, in a busy hour in the AT&T switched network, calls arrive at a rate of about 10 million calls/hr, generating total signaling traffic on the order of a few thousand messages per second. We could reduce signaling load even further without sacrificing much in terms of efficiency by resizing pipes with larger capacities less frequently.

Putting it all together, the pipe resizing technique, operating on the timescale of a few minutes, successfully exploits aggregation and statistical multiplexing, with attendant benefits in scalability and capacity savings:

- Capacity needs are comparable to those of networks that use per-flow resource management in the backbone.
- Signaling overheads are negligible.

Conclusions

Building a robust IP-based telephony service requires coordination of call signaling and resource management. The Distributed Open Signaling Architecture incorporates explicit coordination between the call-signaling and the resource-management protocols to ensure that users are authenticated and authorized



■ Figure 10. *Gain*(t) *for various values of* t.

before receiving access to the enhanced QoS associated with telephony service. This coordination also ensures that the semantics of traditional telephony service are met; specifically, the network verifies that resources are available end to end before the called user's phone is allowed to ring, and the charges for network-layer resources begin only when the called party picks up. DOSA allows services and features to be implemented by intelligent end terminals, while allowing a service provider to add value as a trusted intermediary or through other network-based functionality, such as conference bridging.

Providing a high-quality telephony service that meets stringent bounds on end-to-end packet delay, jitter, and loss requires adequate capacity along the end-to-end path for a voice flow. These functions have traditionally been implemented in the PSTN using per-flow hop-by-hop signaling. In this article we investigate a more scalable alternative involving per-flow signaling and admission control near the edge of the network, and signaling and admission control for aggregate flows in the core of the network. We analyze the performance of two scheduling algorithms, class-based priority and WF^2Q+, to quantify the ability of IP routers to provide bounded queuing delay when using only a single FIFO queue for all voice packets. Our results suggest the need either to use priority scheduling for the voice class or to give it a large weight when using WF^2Q+ scheduling.

Our performance results demonstrate that aggregation does not adversely effect the efficiency with which network capacity is utilized. Trace-driven call-by-call simulations using data from the AT&T switched network demonstrate that the pipe resizing technique proposed in this article, operating on the timescale of a few minutes, successfully exploits aggregation and statistical multiplexing, with attendant benefits in scalability and capacity savings.

In summary, we have presented an integrated framework for scalable call signaling and resource management to support IP telephony.

References

[1] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony," *Proc. NOSSDAV '98*.

[2] M. Arango *et al.*, "Media Gateway Control Protocol," Internet draft, <draft-huitema-MGCP-v0r1-01.txt>, Nov. 1998.

[3] M. Handley *et al.*, "SIP: session initiation protocol," IETF RFC 2543, Mar. 1999.

[4] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec.1998.

[5] R. Callon *et al.*, "A Framework for Multiprotocol Label Switching," Internet draft, < draft-ietf-mpls-framework-02.txt>, Nov. 1997; work in Progress.

[6] A. Lauck, C. Kalmanek, and K. K. Ramakrishnan, "SUBMARINE: An Architecture for IP Routing over Large NBMA Networks," to appear, *Proc. IEEE INFOCOM 1999*.

[7] R. Guerin, S. Blake, and S. Herzog, "Aggregating RSVP-based QOS requests," Internet draft, Feb. 1998.

[8] V. Jacobson, K. Nichols, and K. Poduri. "An Expedited Forwarding PHB," Internet draft, <draft-ietf-diffserv-phb-ef-02.txt>, Jan. 1999.

[9] J. C. R. Bennett and H. Zhang, "Hierarchical Packet Fair Queuing Algorithms," *Proc. SIGCOMM '96*, Aug. 1996, pp. 143–56.

Additional Reading

[1] G. Hjalmtysson and K. K. Ramakrishnan, "UNITE: An Architecture for Lightweight Signaling in ATM Networks," *Proc. INFOCOM 1998*.

Biographies

Pawan Goyal (goyal@ensim.com) received his Ph.D. in computer sciences from the University of Texas at Austin. His research interests are in computer networks, multimedia systems, operating systems, and distributed systems. He has received several awards for academic excellence, including the IBM Doctoral Fellowship. After working at AT&T Laboratories — Research for more than a year, he joined Ensim Corporation (a Silicon Valley startup).

Albert Greenberg's (albert@research.att.com) biography was unavailable when this issue went to press.

Charles R. Kalmanek (crk@research.att.com) received a B.S. degree in applied physics from Cornell University in 1980, an M.S. degree in electrical engineering from Columbia University in 1981, and an M.S. degree in computer science from New York University in 1988. He is currently division manager of networking research in AT&T Laboratories. His research interests include network architecture, protocol design, quality of service, and multimedia systems. Recently, he has been working on cable access networks and Internet telephony.

William T. Marshall (wtm@research.att.com) received a B.S. degree in mathematics from Carnegie Mellon University in 1974, and a Ph.D. degree in computer engineering from Case Western Reserve University in 1979. From 1978 to 1979 he was assistant professor of computer engineering at CWRU. He joined AT&T Bell Laboratories in 1979, and is now a technology consultant in the Networking and Distributed Systems Research Center of AT&T Laboratories. His activities and research interests include all aspects of data communication networks, analytic modeling and performance measurement, and operating system design.

Partho Mishra (partho@research.att.com) received a B.Tech. degree from the Indian Institute of Technology, Kharagpur, in 1988, and M.S. and Ph.D. degrees from the University of Maryland in 1991 and 1993, all in computer science. He is currently a principal member of technical staff in the Networking and Distributed System Research Center at AT&T Laboratories-Research. His research interests include traffic management, wireless networking, and packet telephony and video services. His home page is at http://www.research.att.com/info/partho.

Doug Nortz (dnortz@ems.att.com) is district manager of the Local IP Services Architecture group at AT&T. Since joining AT&T in 1997, he has been leading systems engineering teams investigating AT&T ventures into local telephone service over cable systems. In 1998 he began working on IP-based telephony over cable systems, developing service, architecture, and signaling definitions. Before joining AT&T, he worked for eight years at Bellcore consulting on broadband architecture and service development. He earned an M.S. in information networking from Carnegie Mellon University and a B.S. in electrical engineering from Cornell University.

K. K. Ramakrishnan (kkrama@research.att.com) is a technology leader in AT&T Laboratories-Research, Florham Park, New Jersey. He got his Ph.D. in computer science from the University of Maryland in 1983. He was with Digital Equipment Corporation until 1994, where he was a consulting engineer. His research interests are in the design and performance of protocols and algorithms for computer networks and distributed systems. He is an editor for *IEEE/ACM Transactions on Networking* and *IEEE Network*, and is on the editorial board for *Computer Communications Journal*. He is a member of the End-End Research Group as part of the Internet Research Task Force, and participates in the IETF and ATM Forum. He has worked and published papers in the areas of congestion control and avoidance, local area networks and ATM, distributed systems performance, packet telephony, and issues relating to network I/O.