Providing End-to-End QoS for IP-based Latency-sensitive Applications

Dissertation Proposal

Chen-Nee Chuah
Department of Electrical Engineering and Computer Science
University of California at Berkeley

Committee: Professors S. McCanne (Chair), R. H. Katz, J. Walrand, G. Shanthikumar

Abstract

Advances in IP-technology, which include the real-time encoding and decoding of voice and video, router mechanisms, and real-time transport protocol (RTP) offer great potential for computer-telephony integration over IP networks. However, due to the unpredictable loss, delay, and delay jitter in packet switching network, the conventional Internet single best-effort class of service can adversely impact the end-to-end performance of real-time applications. This dissertation proposal describes a research effort in determining mechanisms to provide Quality of Service (QoS) in an efficient and scalable manner over the Internet. We will elaborate on an end-to-end service architecture that can support per session statistical quality of service guarantees, by building on the significant amount of previous research on network- and application-level QoS solutions. In our initial work, we have chosen Voice over IP (VoIP) traffic as a workload for evaluation, and use a subjective test to determine the effects of network centric parameters on human perceived voice quality.

First, we develop resource allocation techniques for VoIP traffic in virtual private networks (VPNs) to achieve the required QoS. Specifically, we will show how to use traffic statistics in predicting aggregate bandwidth usage, which is crucial for capacity planning and admission control decisions. Since these results may not scale to the publicly shared Internet, we propose to study performance of real-time applications within the Diff-Serv framework. We will describe a) the trade-offs between performance and network efficiency for VoIP traffic in the presence of other integrated computer-telephony applications, e.g., interactive web applications, and b) the interaction between real time flows transported using UDP, and congestion-controlled TCP connections. We expect to make modifications to the existing RTP protocol, so that real-time applications will adapt their rates to network load, and prevent potential starvation of TCP connections. Finally, as an example, we will investigate the role of adaptive audio applications by re-evaluating the play-out delay adjustment algorithms in the modern context. We have shown the existence of time correlation between packet delays, and propose to design and analyze a delay adjustment algorithm based on this conditional delay estimate. We will evaluate this algorithm using trace-based simulations.

1 Introduction & Motivation

1.1 Motivation

In just over a decade, the Internet has grown to over 40,000 sites and 3,000,000 hosts. The startling growth of Internet technology, coupled with the relatively low deployment cost of IP networks, have pushed for an integrating "IP-based core" - a single network for wireless, data, and voice access. However, the diverse service-requirements and novel traffic characteristics of the emerging Internet applications have posed many technical challenges that the Internet community must address in the near future. For instance, with the emerging multimedia applications, e.g., Netmeeting, NeVot, vic & vat, and CuSeeMe, packet audio/video begin to constitute an ever-increasing fraction of Internet traffic. High quality interactive voice and video applications can tolerate little delay variation and packet loss. Table 1 compares the traffic characteristics and service requirements between the traditional data applications and multimedia applications.

Table 1: Heterogeneous Traffic Behavior and QoS Requirements of Internet Applications

Applications	Traffic Behavior	QoS Requirements
Electronic mail (SMTP), file transfer (FTP), remote terminal (Telnet)	Small, batch file transfers.	Very tolerant of delay & loss. Bandwidth requirement: Low. Best effort.
HTML web browsing	A series of small, bursty file transfers.	Tolerant of moderate delay & loss. Bandwidth requirement: Low to moderate. Best effort.
Client-server Business Applications	Many small two-way transactions, "chatty".	Sensitive to loss & delay. Bandwidth requirement: Low to moderate. Best effort, must be stable & reliable.
IP-based voice (VoIP)	Constant bit rate, or bursty traffic (with silence suppression).	Very sensitive to delay & jitter. Sensitive to loss. Bandwidth requirement: Low & predictable. Requires predictable delay & loss.
IP-based Real-time Video	Constant or variable bit rate (depending on the codec).	Extremely sensitive to delay & jitter, and loss. Bandwidth requirement: High, e.g. 384 Kbps. Requires predictable delay & loss.

1.2 Research Issues

One of the main weaknesses of current Internet is the lack of differentiated service. Internet is originally designed for "best effort" datagram service. Since there is no dedicated end-to-end connection between the sender and receiver, packet loss, out-of-order delivery, delay jitter and latency are bound to occur when the shared network is congested. The performance of traditional data applications such as remote terminal (e.g. Telnet), file transfer (e.g. FTP, http), name service (e.g. DNS), and electronic mail (e.g. SMTP) degrade gracefully with respect to packet losses & delays, and therefore are well served by "best effort" service. However, the real-time applications are less elastic. This cause two problems:

- The perceived quality of real-time applications is adversely affected in times of network congestion.
- Current implementations of real-time applications do not back off in the presence of congestion, and therefore interfere with traditional data applications while contending for the network resources.

The decentralized control of IP-networks poses another technical challenge in QoS provisioning: scalability issue. Introduction of new protocols/functions are difficult since all the end and intermediate nodes must be upgraded. Effective QoS solutions must be scalable and able to operate over very heterogeneous collection of access technologies.

1.3 High-level Overview of Previous Work

There are currently two approaches to enhance QoS for real-time flows over the Internet. The first relies on application-level QoS mechanisms to improve QoS without making changes to the network infrastructure. This includes modifying the application implementations to make them more adaptive to variations in packet delays and losses. The main advantage of the application level mechanisms is that solutions can be deployed at end hosts and end routers without making any changes to the core network architecture or network interfaces. The second approach tries to modify the network implementations to provide variable grades of service with some performance guarantees to a heterogeneous mix of internet flows.

1.3.1 Application-Level QoS Mechanisms

In traditional implementations of real-time applications, incoming data are played out either immediately upon arrival or after a fixed delay [1]-[3]. Both methods lead to significant signal degradation under high delay variance conditions. To make real-time applications more tolerant of delays and delay jitter, adaptive techniques were introduced in [4], to dynamically adjust the play-back point. These techniques have been successfully deployed in Internet applications such as vat[1] and nevot [5]. In addition, researchers also study reconstruction methods at the receiver to compensate for packet loss in Internet audio, such as silence substitution, waveform substitution and Linear Predictive Coding (LPC) redundancy [6]. Other application-level QoS mechanisms include layered encoding, low-bit-rate coder [7], and forward error correction (FEC) [8]. Another research area that can potentially improve the end-to-end QoS for Internet audio is the transport layer. RTP/RTCP transport protocol [9] is designed for transporting real-time applications. Voice data is transmitted using UDP packets (RTP) to the receiver, while the RTCP packets provide feedback on delay, jitter and losses to the sender. The application then changes codec parameters such as sample sizes or output rate to adapt to the network condition based on the statistics in the RTCP packets.

1.3.2 Network-Level QoS Mechanisms

IETF proposed two models to provide Internet QoS: Integrated Services (Int-Serv) [10, 11] and Differentiated Services (Diff-Serv) [12, 13].

The philosophy behind Int-Serv is that routers must be able to reserve resources for individual flows[2] to provide QoS guarantees to end users. Int-Serv QoS control framework supports two additional classes of service besides “best effort”: a) Guaranteed service [14] and b) Controlled-load service [15]. Guaranteed service guarantees that datagrams will arrive no later than a certain time, i.e. maximum delay is bounded. This is useful for hard real-time applications that are intolerant of any datagram arriving after their playback time [16]. Controlled-load service is intended to support a broad class of applications which are highly sensitive to overloaded conditions by promising performance as good as in an “unloaded datagram network”. Many research contributions have been made to define the functionality and study the implementation issues of the following Int-Serv components: a) a method for applications to communicate their QoS needs (e.g. Unix RAPI, Winsock2 QoS API), b) a signaling protocol to set up and tear down reservations (e.g. RSVP [17]), and c) traffic control in the network (e.g. packet scheduling, policing).

Unfortunately, Int-Serv faces different challenges that make immediate deployment infeasible:

- The increase in per flow state information that needs to be kept at routers incurs huge storage and processing overhead. Thus, Int-Serv solution does not scale well in the Internet backbone.
- RSVP/IntServ Model needs to work over different data-links such as Ethernet, and ATM. This requires mechanisms that map integrated services onto specific shared media.

---

[1] Vat is an audio conferencing tool developed by Network Research Group of Lawrence Berkeley National Laboratory (http://www-nrg.ee.lbl.gov/vat/).

[2] A flow is a sequence of packets from the sender to the receiver that do not necessarily follow the same route.

Diff-Serv, on the other hand, supports the needs of various applications by using a simple classification scheme. The basic idea behind Diff-Serv model is to carry the QoS information in band within the packet in the Type of Service (TOS) field in IPv4 header or Differentiated Service (DS) field in IPv6 [18]. TOS or DS field is used to indicate the need for low-delay, high-throughput, or low-loss-rate service. The backbone routers provide per-hop differential treatments to different service classes as defined by Per Hop Behaviors (PHBs) [19-20]. Two service models have been proposed: *assured service* [21] and *premium service* [22]. *Assured service* is intended for customers to specific amount of bandwidth needed from service providers, while *premium Service* provides low-delay and low-jitter service, and is suitable for Internet audio/video applications and for IP-tunnels for Virtual Private Networks (VPNs) [23].

Diff-Serv approach has several advantages over Int-Serv:

- Diff-Serv is simpler than Int-Serv and does not require end-to-end signaling (no RVSP).
- Diff-Serv is efficient since classification and PHBs are based on per-class rather than per-flow information. Since the number of service classes are limited by the size of TOS field, and the amount of state information is proportional to the number of classes instead of number of flows, Diff-Serv approach is more scalable than Int-Serv.
- Diff-Serv requires minimum change to the current network infrastructure.

1.4 Our Research Effort

In the previous section, we described several approaches that help provide quality of service for Internet real-time applications, both at the network level and the end application level. The main contribution of this dissertation study will be the design of an end-to-end service architecture, based on Diff-Serv framework and adaptive application concept, that can satisfy QoS requirements of IP-based real time applications in an efficient manner. We approach this problem by studying QoS provisioning mechanisms and their interactions across multiple levels: network, transport and application layers. For our initial work, we consider VoIP as a workload for evaluation because of their traffic characteristics are relatively well-known. We use a subjective test to determine the performance measures in terms of human perceived voice quality in the presence of packet delays and losses. This study will be based on a combination of analysis, trace collection, and simulation. We will address the following problems:

- **Resource allocation for real-time applications in VPNs** VPN services are potential solutions to provide performance assurances close to that of private leased lines over the Internet. However, the existing Service Level Agreements (SLAs) [24] between the network provider and the customers remain vague. Further development is still needed in proper resource management in VPNs before QoS of delay sensitive applications can be accommodated. We will describe how to use traffic statistics to estimate the aggregate bandwidth usage. The results are useful to network managers to determine the amount of bandwidth required for a shared link, and the maximum number of connections that can be supported. We evaluate the proposed solution using VoIP as a workload in a simulation study. We will further validate our results with trace-driven experiments.
- **Design an end-to-end service architecture for real-time traffic under Diff-Serv framework.** We choose to base our study on Diff-Serv because of its scalability and relatively ease of deployment. Previous studies have evaluated the performance of different Diff-Serv service models in a single hop, using several analytical traffic models (e.g., Poisson, on-off source), but no specific conclusions can be drawn from these studies to help determine the best combination of router mechanisms to provide QoS for real-time traffic. First, we propose to evaluate the end-to-end performance of VoIP in the presence of interfering traffic, using Diff-Serv *assured* and *premium* service models, for multiple-hop topology. In the process, we will develop new workload models for integrated computer-telephony applications, such as VoIP and interactive web traffic. We will describe the trade-offs between performance and network efficiency for a given traffic load and mixture. Since the VoIP traffic is transported using UDP flows which do not back off in times of congestion, it can potentially impair

the throughput of TCP connections, and cause network inefficiency. We expect to make some modifications to RTP transport protocol, so that the real-time applications will adapt their transmission rates to network load and become more "TCP-friendly".

- **An example of application-level QoS: improve perceived audio quality by exploiting time correlation of network delays in adaptive playout delay algorithms** The playout buffer for packet audio applications tracks the packet delay and jitter to dynamically adjust the talkspurts playout delay to minimize degradation of perceived quality. Previous approach relies on the marginal distribution of delay statistics, and we propose to exploit additional information about the *correlation between packet delays* in designing new adaptive algorithms/schemes that are more robust. We define "robustness" as the ability of the algorithm to perform well without being affected by the choice of voice traces, or the nature of network delays.

1.5 Proposal Organization

The goal of this research effort is to design cross-level QoS mechanisms that are crucial to provide QoS to IP-based real-time traffic. We use VoIP as a workload for evaluation. Figure 1 shows a protocol stack for the end-to-end IP-based service architecture that we consider. The shaded layers denote the primary focus of this research effort.

The remainder of this research proposal is organized as follows:

- Section 2 gives a more detailed discussion of the related work.
- Section 3 describes VoIP, its performance requirements, and its equivalent mathematical model that we use as a workload for evaluation purposes for all our subsequent simulation study.
- Section 4 discusses how we use prior knowledge about the traffic statistics in capacity planning and resource allocation for VoIP over VPNs. We will discuss our simulation results.
- Section 5 highlights current and future work in designing and evaluating an end-to-end service architecture to provide QoS for real-time applications under Diff-Serv framework.
- Section 6 discusses how we use delay correlation information to improve the efficiency and robustness of the adaptive playout algorithms of packet audio applications.
- Section 7 presents the specific research agenda and development plan for this project.

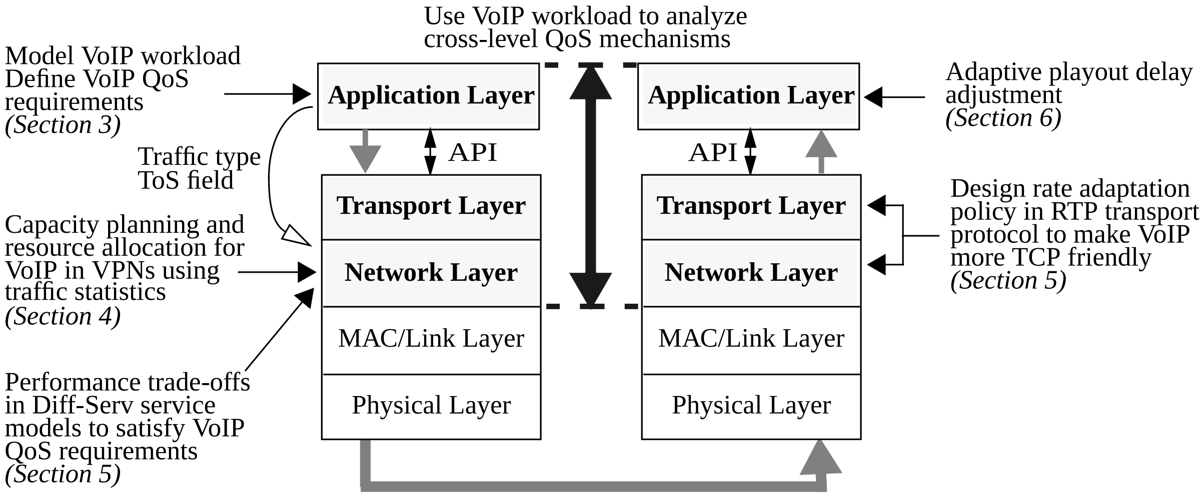

Figure 1. Network architecture and scope. The shaded layers denote the primary focus of this research effort.

2 Related Work

2.1 Providing QoS via Diff-Serv Framework

Besides the previous studies mentioned in Section 1.3, there have been several recent research efforts devoted to understand the delay and/or loss behavior of Diff-Serv service model, via both analysis and experimentation [25-33]. In [25], May et. al. developed analytical models to derive the expected delay and packet loss of Poisson arrivals, for the two Diff-Serv service models: *assured Service* and *premium Service*. Naser et al. [26] quantify the expected delay of packets for constant bit rate and On-Off source traffic using a two-bit architecture which combines *assured* and *premium* services, for a given traffic load and buffer size. S. Sahu et. al. [27] compare the performance of two router mechanisms: threshold dropping and priority scheduling, coupled with edge marking and edge discarding techniques. They observe that priority scheduling provides lower expected delays to preferred traffic than threshold dropping. They also observe that there is little difference in the loss incurred by preferred traffic under both router mechanisms, except when the sources are extremely bursty, in which case threshold dropping performs better. Yeom and Narasimha Reddy discussed techniques for achieving desired throughput guarantees for a Diff-Serv architecture in [28]. They pointed out that the drop precedences by themselves cannot achieve the desired target rates because of the strong interaction of the transport protocol with packet drops, and propose the following techniques to solve the problem: (i) modify the transport protocol at the sender, (ii) modify the marking strategies, and (iii) modify the dropping policies at the router. A few other researchers address the routing issues, and scalable signaling protocols [29, 30]. Adiseshu et. al. [30] proposed a lightweight protocol called SSP (state setup protocol) which is designed to disseminate and manage state information that associates data flows with QoS classes in the routers.

It appears that there has been no published results of end-to-end performance of realistic workload such as VoIP, using Diff-Serv service models for multiple-hop topology. There is also a lack of study on architectural design of admission control for real-time applications over Diff-Serv.

2.2 Playout Delay Adjustment Algorithms in Internet Audio Applications

Previous studies [34-36] have indicated the presence of “spikes” in end-to-end Internet delays. The delay spike is found to have an initially steep rise and linearly monotonic decrease. Bolot’s conjecture [34] attributes this phenomenon to “probe compression” - the accumulation of audio packets (from the flow being examined) behind a large number of packets from other sources in a router queue. Ramjee et. al. show in [36] that it is advantageous to quickly react to the delay spike if the spike spans multiple talkspurts, and present a “delay spike” detection algorithm. They also study the trade-off between average playout delay and loss due to late packet arrivals incurred by different delay adjustment algorithms. S. Moon et. al. revisited this problem in [37] and compute the upper and lower bounds on the optimum (minimum) average playout delay for a given number of packet losses. They also presented a new delay adjustment algorithm that tracks the network delay of recently received packets.

Although there is a significant amount of previous work on playout delay adjustment techniques, most of the results are trace-specific, and based on estimators derived from marginal distribution of packet delays. We need to re-evaluate the performance of these algorithms in the presence of delay correlation, as we will discussed in Section 6.

3 Modeling VoIP

Before we address the various mechanisms used to provide QoS to IP-based real-time applications, we first explain why we choose VoIP as a workload for evaluation, and how we determine its performance requirements and equivalent mathematical model in this section.

VoIP refers to real-time delivery of packet voice across networks using the Internet protocols. The rapid growth of IP-based packet switched networks and the overall bandwidth efficiency of an integrated IP network make it an attractive candidate to transport voice connections. In fact, multiplexing data and voice results in a better bandwidth utilization than the traditional circuit-switched voice-or-nothing backbone in the PSTN (Public Switched Telephone Networks), which consists of over-engineered voice trunks. This justifies looking at VoIP as a workload for future Internet packet networks.

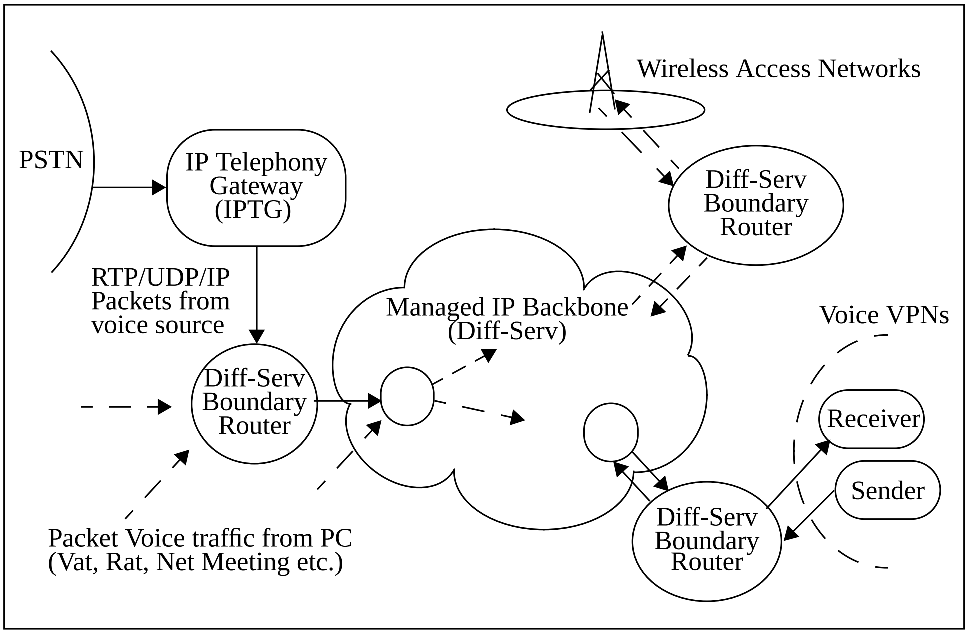

Figure 2. An end-to-end service architecture for Voice over IP using Differentiated Services.

We consider a very simple service architecture, as shown in Figure 2, which includes two important functional blocks: IP Telephony Gateways (IPTG) and Diff-Serv Boundary Routers (DSBR). IPTGs perform the necessary conversion between the transmission format of the input voice traffic and RTP/UDP/IP format that are carried over Diff-Serv networks at the ingress and egress points. Further details about Diff-Serv components and VoIP are described in Appendix A and Appendix B, respectively.

3.1 VoIP Performance Requirements

In this section, we quantify the performance requirements of VoIP, by mapping the human perceived voice quality to the more tangible network centric parameters: packet loss and packet delay.

3.1.1 Packet Loss Rate

In the fall of 1998, we used vat to run a simple subjective test to *map the packet loss rate to perceived voice quality* for the following case: PCM codec with silence suppression, 8 kHz sampling rate, 8 bits per sample (contributing to 64kbps when the source is active), and 20 ms of voice samples per packet.

The sound files of three sentences (about 6 seconds each) from the movie, “A Few Good Men” were downloaded and converted to PCM format with 8 kHz sampling rate[1]. The voice samples were packetized into RTP packets with 12-byte RTP Header and sent through a simple network emulation that introduced uniformly distributed packet losses according to different loss rates. The perceived voice quality was scored on a numeric 0 to 5 scales with the following definitions: 5 = crystal clear, 4 = comprehensible but

---

[1] Since these sound files are in WAV format, we used sndrfmt program from ICSI to resample the voice at 8KHz and convert the format to PCM and saved as μ-law bytes.

less clear; 3 = choppy speech; 2 = harder to comprehend sentences due to noise; 1 = can comprehend less than 50% of the sentence; 0 = gibberish noise. The result is plotted in Figure 3. Results show that the tolerable loss rates are within **1-3%** and the quality becomes intolerable when more than 3% of the voice packets are lost.

Note that packet voice using Forward Error Correction (FEC) is more resilient to losses and therefore we would expect the curve to shift to the right in this case. On the other hand, the quality of voice connection using compressed speech is more sensitive to lost voice samples, and we expect the curve to shift to the left. The impact of packet loss on voice quality depends on the codec used, burstiness of losses, and frame sizes per packet, but this is out of scope of this project. We have considered packet losses that are caused by buffer overflows in routers as well as discarding of delayed packets that miss the playout time at the receiver.

3.1.2 Maximum Tolerable Delay

ITU-T Recommendation G. 114 [38] specifies that one-way transmission time for connections with adequately controlled echo should be in 0-150 ms range to be acceptable for most user applications. As mentioned earlier the end-to-end delay for VoIP depends on various components of the packet network. In this project, we assume PCM transcoding introduces almost negligible delay if implemented in hardware (0.75 ms). We also assume that propagation delay is relatively constant and can be easily estimated. From [38], Public Land Mobile Systems contribute around 80 - 110 ms to one-way propagation time. Satellite systems introduce 12 ms at 1400 km altitude, and 110 ms at 14,000 km altitude. Optical fibre cable system contributes around 50-60 ms from coast to coast in United States. Assuming it takes 100 ms propagation delay for voice packets to be transported across United States, the total queueing delay should be kept within 50 ms (150ms - propagation delay). Since queueing delay is the only variable part in our model, we need to budget the per hop queuing delay. From traceroute, we found out that there were typically around 8-12 hops between a machine on the west coast and the east coast. Assuming that queueing delay is almost the same for each hop, we require the per hop queueing delay to be *at most 5 ms* when we design our resource allocation schemes.

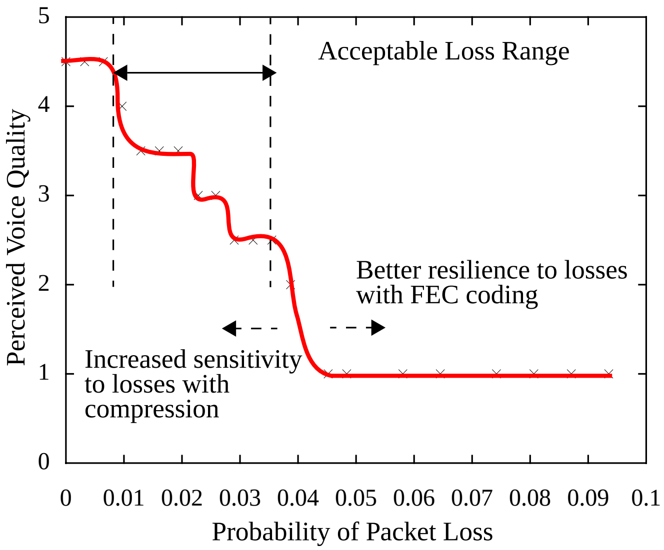

Figure 3. Perceived voice quality when different packets are dropped randomly according to different loss rates.

3.2 Analytical Model

With silence suppression, each VoIP source can be modeled as an on-off Markov process. The alternating periods of activity and silence are exponentially distributed with average durations of $1/\mu$ and

$1/\lambda$, respectively. The fraction of time that the voice source is "on" is $\frac{\lambda}{\lambda+\mu}$. When the source is in the "on" state, fixed-size packets are generated at a constant interval. No packets are transmitted when the source is "off". The size of the packet and the rate at which the packets are sent depends on the voice codecs and compression scheme. Let $X_i(t)$ be the instantaneous rate of voice connection *i*, then,

$$X_i(t) = \begin{cases} \mathrm{R} & \text{when the voice source is active} \\ 0 & \text{when the voice source is silent,} \end{cases} \tag{1}$$

where R is the voice bit rate (i.e. packet size/packet interval). The rate of transition from state '0 kbps' to state 'R kbps' is λ while the reverse transition happens at the rate of μ.

4 Capacity Planning and Resource Allocation for VoIP in VPNs

Latency and packet losses in IP-networks have adverse impact on the perceived voice quality (as shown in Section 3.1), and therefore need to be bounded. One potential solution is to have a fully managed networks like VPNs, which provide more predictable performance assurances. In this section, we show how to efficiently allocate resources in VPNs to satisfy QoS requirements of VoIP connections.

4.1 Background

VPNs represent the latest alternative for constructing a secured wide area network (WAN) to connect private corporate networks or individual users over the shared public Internet. The substantial progress in IP security [39] and the relatively low deployment cost of IP-based networks have resulted in the dramatic growth of VPNs. Some Internet Service Providers (ISPs) and other network providers offer VPNs as fully managed services, which provide performance assurances close to that of private leased lines. Users planning to adopt Internet VPN require, at a minimum, a predictable performance over the VPNs' secure tunnels, which is usually specified in the Service Level Agreements (SLAs). An SLA [24] is a contract between the service provider and the customer that specifies delay, throughput and loss characteristics needed to meet the application requirements. It can also include the maximum packet transmission rate as promised by the customer. Packets transmitted in excess of the specified rate may not get the desired service level. Traditional ISP SLAs have focused on backbone performance guarantees, such as backbone/network availability, maximum or average delay, and mean time to notify customers of an network outage. However, further development is needed to extend the SLA to customer on an end-to-end basis. It is important to understand the detailed technical specifications of the QoS attributes in the SLA related to delay, loss and jitter.

4.2 Research Issues

From the application performance point of view, VPNs become natural course grained solutions to provide better QoS for IP-based real-time applications such as VoIP. However, there are relatively few studies on resource management issues related to VPNs, which is important for supporting QoS. Besides security and privacy issues, other work on IP-based VPNs has mainly focused on group membership, routing protocols and tunneling [40].

In this section, we investigate how a prior knowledge about the first and second order statistics of the VoIP traffic can be used to predict the aggregate bandwidth requirement. VPN service providers can make appropriate pre-allocation of bandwidth to different classes of traffic, e.g., delay sensitive vs. best effort, based on this bandwidth predictor. It is reasonable to assume that the number of possible voice sources, *N*, is known, and off-line measurements of traffic statistics is relatively easy in VPNs. Specifically, we demonstrate how to allocate limited resources to meet QoS constraints for VoIP in VPNs. The following are two fundamental network design questions that are of interest:

- Given that N VoIP flows arrive at a node i, how much bandwidth, C, and buffer space, B, should be allocated to the VoIP service class to limit packet loss to at most 3% (Section 3.1.1) and delay to at most 5 ms (Section 3.1.2),
- Given a fixed link capacity, C, what is the maximum number of VoIP flows, $n_{\max}$, we can admit into the queue before the QoS requirements of VoIP are violated?

In the fall of 1998, we addressed the questions mentioned using both analytical and simulation techniques, and submitted the results to ICNP'99 [41] (attached as Appendix D). Although our study is based on VoIP workload, the results can be extended to other delay sensitive applications. The next two sections briefly describe our findings.

4.3 Aggregate Bandwidth Predictor

We consider N potential incoming VoIP traffic served at C_v kbps. Each VoIP source is modeled as an on-off Markov process as described in Section 3.2. Let X_i be the instantaneous arrival rate of flow i, and $Y = \sum_{i=1}^{N} X_i$ in the equivalent discrete time model (Appendix C). Assume that X_i's are i. i. d. with mean m and variance σ^2. When N gets large, Y tends to have a normal (Gaussian) distribution with mean Nm and variance $N\sigma^2$ under the *Central Limit Theorem* [42], $Y \sim N(Nm, N\sigma^2)$. Assume that losses occur when $Y > C_v$, and the maximum tolerable loss is $P(Y > C_v) = \delta_{\max}$. Using the well-known Q-function, we can estimate the required bandwidth, C_v^o as the following (detailed derivation can be found in Appendix C),

$$C_v^o \geq Nm + Q^{-1}(\delta_{\max}) \cdot \sqrt{N} \cdot \sigma \qquad (2)$$

In practice, the voice traffic arrives in IP-packet format, and there are buffers in the system. The ideal fluid model in Appendix C does not hold, but the results in Eq. (2) can still serve as a first order approximation of the actual bandwidth required. From Figure 3 in Section 3.1.1, the packet loss rate should not exceed 3% to preserve satisfactory voice quality. Therefore, we chose $\delta_{\max} = 0.03$. With $Q^{-1}(0.03) = 1.88$, the total bandwidth needs to be at least:

$$C_v^o = N \cdot m + 1.88 \cdot \sqrt{N} \cdot \sigma \qquad (2a)$$

to satisfy the QoS requirements of VoIP. As a numerical example, let $\lambda = 0.4$, $\mu = 0.6$, R = 80 kbps. From Eq. (2), the mean $m = 0.4*80 = 32$ kbps, while $\sigma = \sqrt{E[|X-m|^2]} \approx 39.2$ kbps. For every value of N, one can then estimate the capacity needed by substituting the numerical values into Eq. (2a).

The statistical gain from multiplexing multiple flows together results in a lower "effective" bandwidth per flow. Therefore, the actual bandwidth that is needed to meet the performance requirement of VoIP can be less than the predicted value of C_v from Eq. (2a), which can be viewed as a upper bound. For comparison purposes, we denote the required bandwidth computed from analytical model as C_v^o, and compare it with the actual bandwidth determined from simulation, denoted as C_v^{ns}, in the next section.

4.4 Numerical Results & Discussions

We used simulation to determine the minimum bandwidth, C_v^{ns}, needed to support a specific number of voice users N. We used ns simulator[1] to model a simple one-hop topology where N VoIP flows share the same link served at C_v kbps. The VoIP sources were simulated according to Section 3.2 with voice activity cycle of 40% (λ=0.4). Assume that 8 KHz 8 bits/sample PCM codec was used with 20 ms frame per packet. The voice data packets were 160 Bytes. With 12 byte RTP header, 8 byte UDP header and 20 byte IP header, the size of each packet = 200 Bytes. With these header overheads, the effective rate of a single voice connection when it was active was (200*8)/20 =80 kbps (25% overhead). Buffer size *B* was chosen to limit the maximum possible delay (for packet at the tail of the queue) to at most 5 ms:

$$(B \times 200 \cdot 8) / C_v^{ns} \le 5 \text{ ms.} \qquad (3)$$

N was varied from 0 to 200. For each *N*, C_v was increased until the worst per flow loss rate decreased to 3%, and the corresponding value $C_v^{ns}(N)$ was recorded. Figure 4 shows how $C_v^{ns}(N)$ was determined for *N*=20 and *N*=30. Result were plotted in Figure 5, together with C_v^o that was predicted using Eq. (2a). Our main findings were as follows:

- We found that $C_v^{ns}(N)$ increased linearly with *N* at a slope of approximately 33 kbps for each additional voice connection (per unit of *N*). *The slope was roughly 40% of the peak rate of 80 kbps, which was very close to the mean rate of 32 kbps.*
- C_v^o served as an upper bound for the actual bandwidth usage, and tracked C_v^{ns} fairly well for moderate *N*. This was a pleasant discovery since it implied that if we know the mean and variance of each individual flow, we can estimate the aggregate bandwidth required. By allocating C_v^o kbps to the whole VoIP service class, VPNs can give statistical guarantees (maximum packet loss & delay) to each individual VoIP flow.

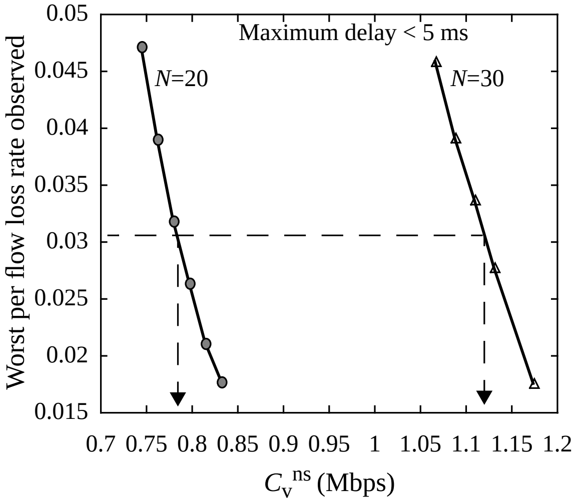

Figure 4. The worst case per flow packet loss rate is plotted against the available bandwidth, C_v^{ns} for *N*=20 and *N*=30. We can determine the required C_v^{ns} to achieve 3% loss rate for each *N* from a family of curves like these, as shown by the dotted lines.

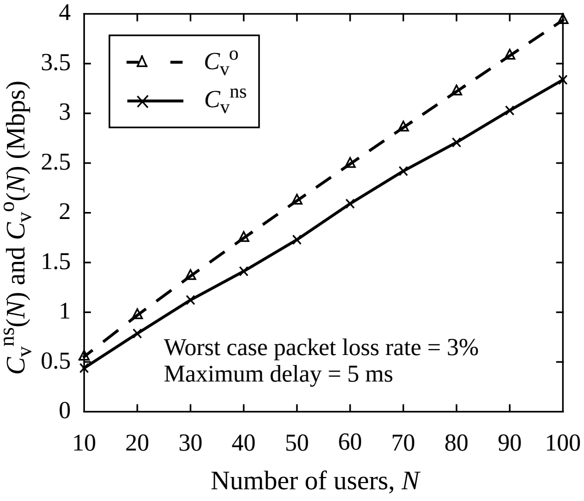

Figure 5. This figure demonstrates that proper bandwidth provisioning is sufficient to support performance requirements of VoIP. C_v^o is the required bandwidth predicted from Eq. (2a) while C_v^{ns} is the results obtained from ns simulations.

The result shown in Figure 5 can also be interpreted in a different way. We defined N_{max} as the maximum number of users that could be supported so that loss rate was below 3% and delay was less than

---

[1] ns is a discrete event simulator derived from REAL simulator [43]. The ns development effort is now part of the ongoing VINT project.ns source code and documentation can be downloaded from http://www-mash.cs.berkeley.edu/ns/.

5 ms. Using the same experimental settings, the value of $N_{\max}$ for each corresponding value of available bandwidth, C_v, was obtained from Figure 5 for: a) ns simulation, and b) analytical prediction. $N_{\max}$ was plotted against C_v, for these two cases in Figure 6 (solid and dashed lines). For comparison, we considered a third case where peak rate of 80 kbps was allocated to each flow (assuming all sources became active simultaneously). $N_{\max}$ in this case was the largest integer such as $N_{\max} \cdot 80 \le C_v$ kbps (dotted line in Figure 6). This line lay much lower than the previous two cases. This implied that the peak rate allocation was over-conservative and resources were under-utilized. For example, at 6.4 Mbps link bandwidth, one could only support 80 users with peak rate allocation. But simulation showed that one could actually support up to 196 users, 2.4 times as many (solid line in Figure 6). If we allocated bandwidth based on Eq. (2a), we could support $N_{\max} = 170$, which was still more than double the $N_{\max}$ with peak rate allocation. Further details of this work can be found in [41].

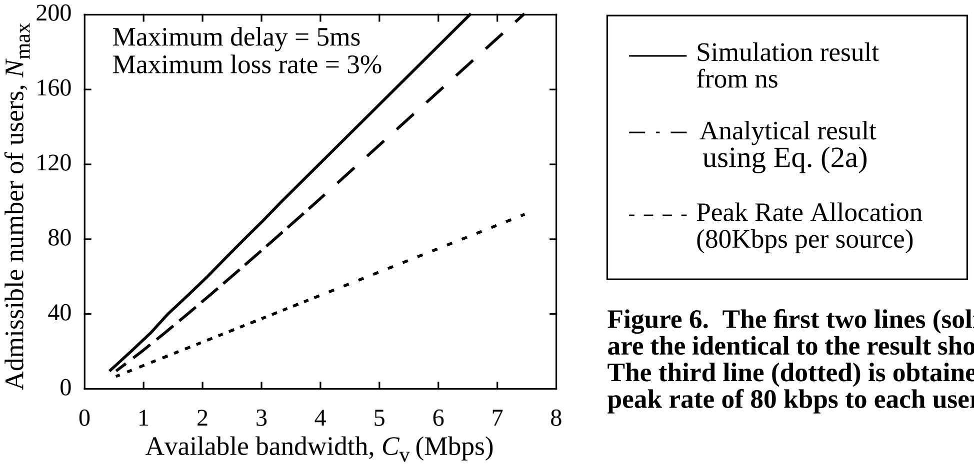

Figure 6. The first two lines (solid and dashed) are the identical to the result shown in Figure 5. The third line (dotted) is obtained by allocating peak rate of 80 kbps to each user.

4.5 Future Work

We will further validate the model we use to describe VoIP source in Section 3.2 by performing a number of trace-driven experiments. We plan to collect the following two sets of traces:

- audio packets from two way voice conversations conducted via Microsoft Netmeeting[1] running on two NT machines (connected by a LAN) at Berkeley. In the recording of voice traces, one can specify the encoding schemes (e.g. 64 Kbps/ 8-bit PCM μ-law or A-law, 13 Kbps GSM) in Netmeeting.
- H.323 packets between a telephone on a circuit switched network (PSTN) and a PC/Laptop running Netmeeting, that go through the H.323 [44] gateway in the Iceberg[2] test-bed in Lab 420, Soda Hall. Figure 7 shows the components in Iceberg testbed that are directly involved in our trace collections. The H.323 gateway provides services such as translations between different transmission formats, communications procedures, and audio codecs to H.323 clients so that they can communicate with non-H.323 entities.

We will use these traces as a workload to evaluate the resource allocation technique described in Section 4.3, and compare the results to that obtained in Section 4.4.

---

[1] NetMeeting is a Window based Internet real-time conferencing tool that supports multi-point data conferencing, text chat, and point-to-point audio and video (http://www.microsoft.com/windows/netmeeting).

[2] The Iceberg project seeks to construct a scalable service architecture for computer-telephony (both PSTN and wireless cellular infrastructure) integration based on Internet Protocol (IP) technology (http://iceberg.cs.berkeley.edu).

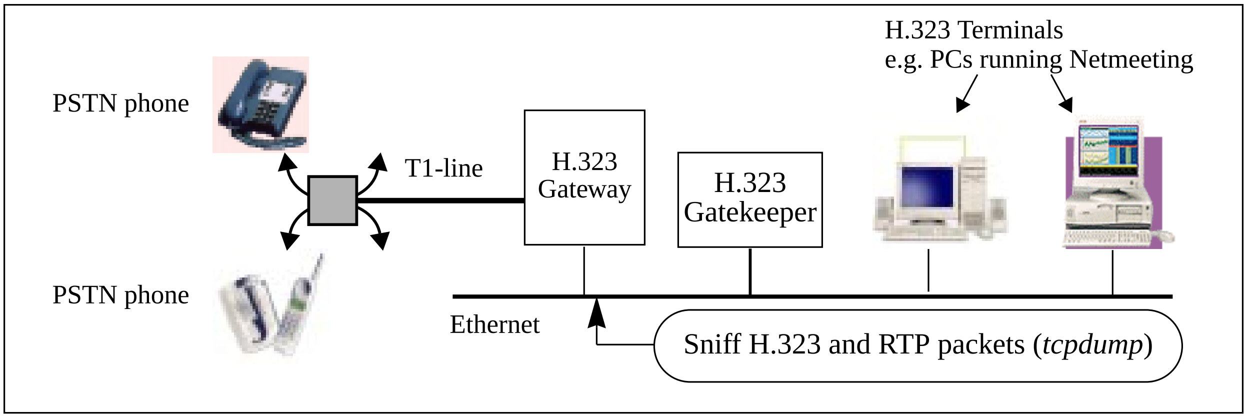

Figure 7. Testbed set up for trace collection, including the H.323 gateway, H.323 terminals running client applications (e.g., Netmeeting) and a T-1 line connecting to circuit-switched telephone lines.

5 End-to-end Service Architecture Design Based on Diff-Serv

5.1 Background

We have shown in Section 4 that it is possible to provide statistical guarantees to real-time traffic like VoIP in VPNs by proper resource allocation. We can predict the total required bandwidth for a given user population (N_{max}) and traffic characteristics (μ and σ). However, these solutions may not scale well to publicly shared IP-network which span multiple autonomous systems and loosely coupled networks belonging to different service providers, where end-to-end traffic monitoring is hard or impossible.

In this section, we investigate mechanisms to provide some service discrimination in the *general IP-networks* to achieve better QoS for real-time applications. We will consider Diff-Serv architecture (Appendix C), which has been proposed as a low-cost way to augment the single class Best-Effort service model for the current Internet to a flexible network architecture that can accommodate emerging Internet applications with diverse service requirements. However it remains uncertain how the concatenation of Diff-Serv routers affects the end-to-end QoS of real-time applications under high network load, both at the application level (e.g. perceived voice quality, jitter-free video etc.), and network level (e.g. controlled latency, dedicated bandwidth, low packet loss etc.).

5.2 Research Issues

Many previous research efforts have studied the performance of Diff-Serv service models by using various analytical traffic models. However, no specific conclusions can be drawn from previous studies as to which combination of router mechanisms can provide satisfactory QoS for real-time applications. In addition, most of these studies evaluate the performance of a specific router mechanism (scheduler, buffer management, or packet marking techniques) for *one hop* case. Since what matters the most to the end users is the *end-to-end QoS*, it is important to quantify the *end-to-end performance* observed at the end receiver, after the flow traverses *multiple nodes*. Below we describe a number of issues that we will address:

5.2.1 Performance Trade-offs for Novel Workloads

The first part of the study intends to compare the QoS requirements of VoIP (Section 3.1) using two Diff-Serv service models a) *premium* and b) *assured service*. For *premium service*, we consider a priority scheduler with separate FIFOs for different service class. For *assured service*, we consider weighted fair queueing (WFQ) [45-47] with separate queues running REDs for the various types of traffic that share the same path. Besides VoIP, we will consider the following applications as interfering traffic:

- interactive web traffic,
- TCP connections (telnet, ftp),
- novel integrated computer-telephony applications, e.g. stock quotes with text to speech conversion.

We will describe the trade-offs between delay and packet loss, and network efficiency for a variety of mixed workloads. We will address this problem in multiple-hop scenarios through a simulation study. We will implement both *premium Service* and *assured Service* in ns simulator. We will determine the aggregate bandwidth that is required using these two service models to achieve the QoS goal determined in Section 3.1 for VoIP traffic. The results will be compared to C_v^{ns} and C_v^{o} obtained in Section 4.4 for VPNs. In the process, we will develop appropriate models for novel integrated computer-telephony applications. One observation that is of interest is how a concatenation of priority scheduler shapes the behavior of the input traffic traversing the multiple-node path, and how this compared to the case when WFQ-REDs are used.

Besides real time video/audio and traditional data applications, web traffic constitutes an increasing fraction of the Internet traffic. Though not real time, some of the interactive web applications are delay sensitive and impose more latency constraints than best-effort traffic. It is important to understand whether it is necessary to separate this class of traffic from best-effort. We will also study the effect of background web traffic on throughput of TCP connections as well as the loss and delay behavior of RTP flows in the following two scenarios:

1. Using a service model that contains only two classes: real-time and best effort, and inserts packets from web applications to the best-effort queue.
2. Using a service model that contains three service classes: real time, latency sensitive and best-effort, and inserts packets from web applications to their own queue under “latency sensitive” category.

The results will be useful as a guideline for network managers to choose proper router mechanisms to meet the performance requirements of real-time applications and other workloads, while achieving high bandwidth efficiency.

5.2.2 Effect of Traffic Heterogeneity on QoS Performance

Most of the previous works on Diff-Serv examine the performance of homogeneous traffic, and often use easily tractable traffic models such as Poisson process. We need to relax this assumption, and study the case of heterogeneous traffic that are more realistic. One question of interest is how traffic heterogeneity affects the end-to-end performance of real-time applications, using same router mechanisms (as described in Section 5.2.1) with the same set of parameters. Next, we will determine the choice of parameters that are needed to achieve the same kind of performance with traffic heterogeneity.

We plan to use the traces collected in Section 4.5 in a simulation study to perform sensitivity analysis described above. We expect to modify our modeling methods of various workloads based on the results. The modified models will potentially be of interest to other network researchers.

5.2.3 TCP-friendly Transport Protocol for Real Time Applications

Although our main goal is to provide end-to-end QoS for real time applications, we would like to achieve this goal without penalizing the throughput of other traffic sharing the same link. Since TCP flow and congestion control algorithms have strong interaction with packet drops, and since traditional implementations of real time applications do not back off in times of congestion, the overall throughput of TCP connections can be impaired in an unfairly manner.

First, we will quantify the effect of UDP flows on the throughput of TCP connections and identify scenarios when starvation of TCP connections occur. Next, we will design a rate-adaptation protocol based on the statistics on packet delays and losses carried in the RTCP packets, which are assumed to be good indicators of network conditions. We expect to modify the existing RTP transport protocols, so that the real-time applications can adapt the transmission rates to network load and become more “TCP-friendly”. We will consider the following performance measures:

- Average throughput of TCP connections, and
- Human perceived disruption as the transmission rate is changed to adapt to the network conditions. We will run a network emulation to calibrate the human perceived disruption by inserting an active module to artificially force the change of transmission rates, e.g. change the codecs or compression algorithms used at the sender.

6 Application QoS Solution: Better Playout Delay Adjustment Algorithm

In Section 4 we have presented an example on how resource allocation techniques can be used to provide QoS to VoIP in well-managed VPNs. In Section 5, we investigate the performance trade-offs in DiffServ service architecture to meet QoS requirements of VoIP service class while maintaining overall bandwidth efficiency and data throughput of congestion controlled traffic such as TCP. However, Braslau and Shenker [48]-[49] show that the usefulness of a priority-based service differentiation at the network level depends heavily on the *adaptive capabilities of applications*. As an example, we evaluate the role of such application QoS solution for Internet audio in this section. We will design a better playout delay adjustment algorithm based on delay correlation that exists in current Internet.

6.1 Background

Although the network-level QoS mechanisms can help limit the queueing delay and packet loss experienced by IP-based audio traffic, delay variations still exists. The existence of delay spikes, even if rare, have adverse impact on human perceived audio quality. In order to compensate for the network delays, packets are usually buffered at the receiver in adaptive Internet audio applications. The periodic playout of these packets is delayed a certain amount of time beyond the reception of the first packet in the talkspurt. We refer to this additional "artificial" delay as the *playout delay.* There are currently two approaches for adaptive playout adjustment: per-talkspurt and per packet adjustment. The former approach uses the same playout delay throughout a talkspurt, but varies the playout delays from one talkspurt to another. Figure 8 shows the relationship between the various timings associated with the *i*-th packet in the *k*-th talkspurt when it is sent, received and eventually played out at the receiver. Per talkspurt adjustment results in artificially expanded or compressed silence periods, as indicated by s_k and s_k' in Figure 8. Fortunately, these changes are not noticeable in played out speech if the change is reasonably small [1]. The second approach is to vary the playout delay from packet to packet, but this introduces gaps inside talkspurts that has been reported as being damaging to audio quality [50].

It is clear from Figure 8 that the longer the playout delay is, the more likely it is that a packet will have arrived before its scheduled playout time. However, excessively long playout delays can damage the perceived quality of human conversation. On the other hand, if the playout delay is too small, it incurs some amount of packet loss due to late packet arrivals. Therefore, there is a critical trade-off between the choice of playout delay and packet losses, depending on the severity of their impact on audio quality. As a rule of thumb, delays less than 400 ms [38] and loss percentage up to 3 % (Section 3.1) are considered to be tolerable for interactive human conversations.

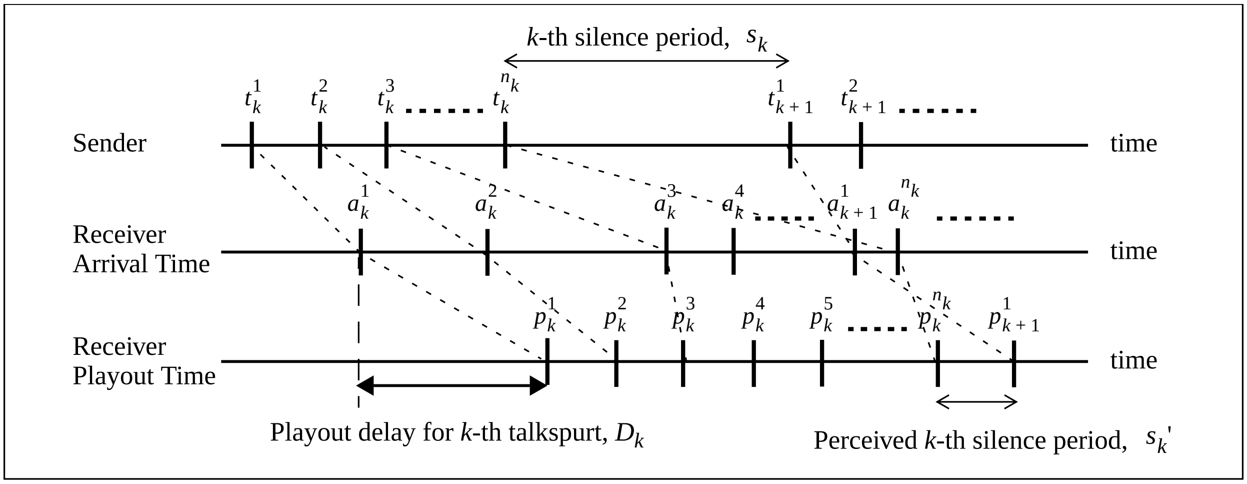

Figure 8. Timing information of audio packets at the sender, receiver, and output of playout buffer.

6.2 Research Issues

Playout delay adjustment has been widely used in many Internet audio/video applications. Unfortunately, due to the rapidly changing nature of traffic patterns and dynamics, and with the deployment of new router mechanisms, the network delays we experience today exhibit significantly different characteristics than the ones observed in 1993 [34]. These have made some of the delay adjustment algorithms obsolete, less robust or less efficient. It is time to reevaluate these algorithms in current context and address any new challenges that arise. The following issues remain to be investigated.

6.2.1 Time Correlation in Network Delays

All the previous work on playout delay ignore the time correlation of network delays and only look at their marginal distribution. It is important for us to understand a) if any significant correlation exists between packet delays, and b) if so, how can one exploit the correlation information to improve the performance of the existing playout delay adjustment algorithms.

We started by examining the conditional distribution of packet delays based on a more recent network traces collected by Jonathan Rosenberg in September, 1997 [51]. Packets are generated as if using G.723.1 at 6.3kbps, one frame per packet, plus RTP/UDP/IP packet headers, resulting in packet size of 36 bytes. The packet stream is transmitted from Columbia University, New York to University of Massachusetts, Amherst, and echoed back to the sender site. The send and receive times are noted, along with the sequence numbers. We computed the round trip time, RTT_m, for each packet m and plotted its empirical distribution in Figure 9(a). From the data, we found that the mean μ_{RTT} is 0.108 s, while the standard deviation σ_{RTT} is 0.053 s. The distribution shapes like an asymmetric Gaussian centered around 0.1 s. On the same graph, we plotted the conditional distribution of RTT_m given that the delay of previous packet is greater than 0.161 s (i.e. the mean + one standard deviation). Note that if there is no time correlation between the packet delays, the conditional distribution should look similar to the marginal distribution. However, results show that $f(RTT_m | RTT_m \geq \mu_{RTT} + \sigma_{RTT})$ is actually shifted to the right with a higher mean at 0.176 s, and a narrower peak around 0.15 s. The complementary cumulative distribution functions (CCDF) for RTT_m is plotted in Figure 9(b) along with the CCDF when conditioned on the event that previous delay sample, RTT_{m-1} is greater than 0.161 s.

This proves the existence of substantial time correlation in network delays. The next step is to design a better delay adjustment algorithm using a better delay estimator based on the knowledge of network correlation, which has the following advantages over the existing approach:

- In [36] and [37], the adaptive algorithm is based on estimating two statistics: the delay itself, and a variational measure of the observed delays. Both estimates are in the form of $u_k^i = \alpha u_k^{i-1} + (1-\alpha) d_k^i$, where u_k^i and u_k^{i-1} are the *i*-th and previous (*i*-1)-th estimates, while d_k^i is the *i*-th delay sample. The weight, α, is chosen based on off-line sensitivity analysis using simulation, without considering the presence of delay correlation. Clearly, the accuracy of these estimates can be greatly improved by dynamically choosing the values of α^i that based on the observed delay correlation.
- The previous approach [36] depends on "delay spike" detection algorithm to decide whether the playout delay should be adapted, based on whether the delay spike spans multiple talkspurts. If we design an algorithm based on network correlation, the decreasing correlation (as shown by α^i) will automatically detect the end of delay spike and adjust the playout delay accordingly. This can potentially improves efficiency of the delay adjustment algorithm.

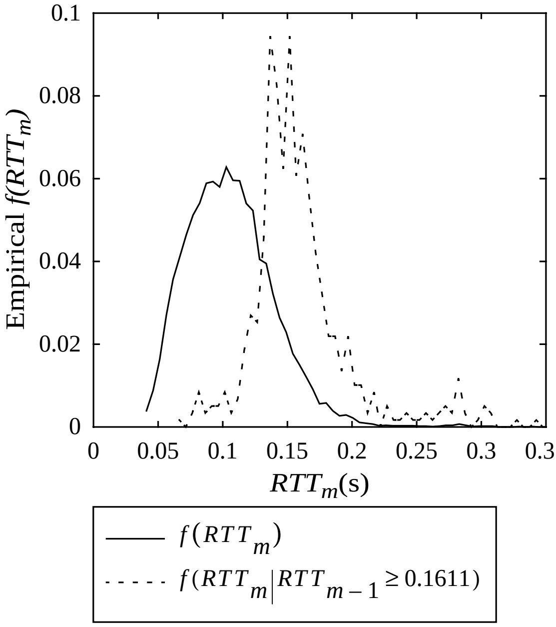

Figure 9(a). Empirical distribution of RTT_m and the conditional distribution given that the delay of RTT_{m-1} is greater than 0.161 s ($\mu_{RTT} + \sigma_{RTT}$).

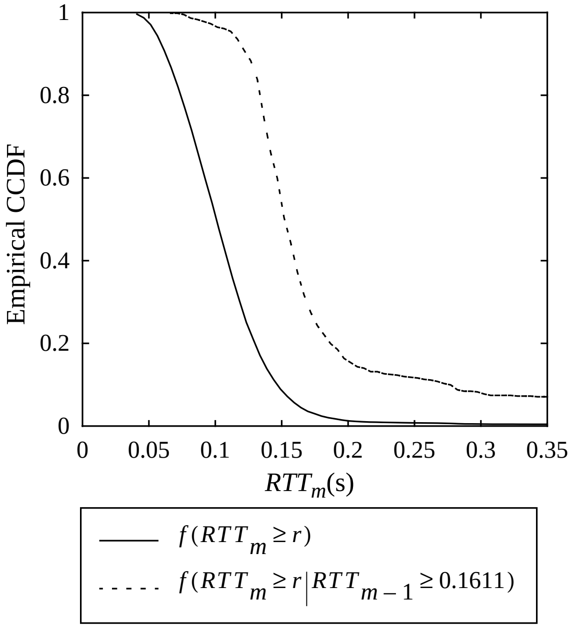

Figure 9(b). Empirical CCDFs of RTT_m and the conditional CCDFs of RTT_m given that RTT_{m-1} is greater than 0.161 s ($\mu_{RTT} + \sigma_{RTT}$).

6.2.2 Delay Jitter: Definitions & Bounds

Some of the existing delay adjustment algorithms estimate the absolute or relative delay of received packets (e.g. Nevot), and adjust the appropriate playout delay accordingly. Besides the end-to-end delays, another option is to estimate the delay jitter (e.g. vat), which is easier to estimate because of the following reasons:

- Delay jitter has less dynamic range than the actual queueing delay, and therefore the estimates are more accurate.
- Delay jitter is easier to measure because clock synchronization is less critical.

Jitters are variances in delay, which come from queueing delays. In [55-56], researchers have defined and derived different analytical definitions and bounds on delay jitter. From my discussions with Ramesh Nagarajan[1], we define the delay jitters as the following using the notations as shown in Figure 8:

- 1-jitter: $(a_k^{i+1} - t_k^{i+1}) - (a_k^i - t_k^i)$ (per packet basis),
- t-jitter: $(a_k^i - t_k^i) - (a_k^1 - t_k^1)$ (reference point is the first packet of each talkspurt),
- T-jitter: $(a_k^i - t_k^i) - \min_{j,\,l}(a_l^j - t_l^j)$ (the second term is minimized over all packets in all talkspurts).

We assume that out-of-order is taken care of. Note that T-jitter is the variable part in queueing delay since the minimum round-trip time is subtracted off. On the other hand, 1-jitter distribution has a much smaller variation than T-jitter, since it is just the delay variation between adjacent packets. The next challenge is to evaluate different types of jitter definition to see which is the most appropriate/effective candidate to be used as the estimates for the playout delay adjustment algorithms.

6.3 Research Plan

Below we describe two specific research efforts that we will pursue.

- We will use delay correlation information to get a better estimate of the future packet delay, and design a new playout delay adjustment algorithm based on this new estimate. Since a delay spike may or may not be completely contained within a talkspurt, the algorithm should be able to differentiate between the two cases and adjust the delays accordingly. One approach is to introduce additional threshold, β, such that if the delay correlation is greater than β, the algorithm will track the delay and adjust the playout delay accordingly to compensate for the network delay. Otherwise, we assume that the correlation is too small, and will not adapt to the changes.
- We will explore different playout delay algorithms using different jitter estimates as defined in Section 6.2.2. We will evaluate the performance of our algorithms based on simulations using the traces we collected in Section 4.5, and compare the results to the existing algorithms approach.
- We will perform sensitivity analysis on how the performance of the proposed algorithms are affected by the nature of network delays. We consider a) the current Internet and b) the proposed service architecture built on Diff-Serv framework (Section 5), since they will potentially have very different delay characteristics.

We are interested in the following performance measures: a) loss vs. average playout delay, b) number of gaps in a talkspurt, c) average length of gaps in a talkspurt, d) total length of gaps in a session, and e) mean opinion score.

7 Summary and Research Agenda

7.1 Summary of Research Contributions

The main contribution of this research will be the design of end-to-end Diff-Serv Service architecture, based on TCP/IP protocol suite, that provides high quality interactive Internet audio to end users while maintaining network efficiency (see Figure 1 for the high-level picture). We will base the design on a combination of analysis and simulation, in addition to trace collections and network emulations where feasible.

---

[1] Ramesh Nagarajan is a Distinguished Member of Technical Staff (DMTS) in the Performance Analysis Department, Lucent Technologies, Holmdel, NJ.

7.2 Research Plan

The previous sections (Section 4-6) have described a number of tasks which will move us forward toward our research goals, based on the concept of a) allocating resources based on predicted bandwidth requirements in VPNs, b) providing differential service in the transporting network through Diff-Serv framework and c) dynamically adjust playout delay in adaptive end applications. This section outlines the specific research agenda and a preliminary development plan.

7.2.1 Capacity Planning & Resource Allocation in VPNs

The following work has been completed:

- We have run a subjective test to map the packet loss to the perceived voice quality (0-5 scales).
- We have derived a bandwidth usage predictor for aggregate VoIP traffic using mathematical model.
- We have designed, implemented, and evaluated, through ns simulation, a specific instance of capacity planning and resource allocation for VoIP service class based on predicted bandwidth usage.

We will pursue the following:

- collect Internet audio packets (RTP and/or H.323 packets) using Netmeeting and H.323 gateway in Iceberg test-bed, and use these voice traces in simulation study to validate VoIP traffic model.

7.2.2 End-to-end Service Architecture Design

We will follow the iterative “design, simulate, refine” strategy to design a service architecture using Diff-Serv framework to support real-time applications.

Stage 1: Analysis and Design

We have initiated the following work:

- Assume there are two classes of traffic: delay sensitive vs. best-effort, and we will derive a performance bound for packet delays and losses in the multiple node case using Diff-Serv framework.
- We will develop workload models for new computer-telephony applications, including interactive web applications and text-to-speech applications.
- We will design rate-adaptation policy for RTP transport protocol to react to the network load.

Stage 2: Trace Based Simulation

We will pursue the following:

- implement *premium service* and *assured service* in ns simulator, including packet marking/discarding, scheduler and packet discarding policies that are of interest.
- incorporate our rate-adaptation policy for RTP in ns, and use simulations to analyze the effect of increasing VoIP workload on data throughput and voice quality, with and without rate adaptation.
- extend our work to include interactive web traffic as the third service class within the Diff-Serv framework.

Stage 3: Evaluation & Refinement

Based on the simulation results, we will propose new solutions or modification to the originally proposed service architecture. We expect there will be some overlap between Stage 2 and Stage 3. We plan to:

- apply the differentiated service concept on Iceberg test-bed. If available, we plan to use the Accelar 790 Server Switch from Nortel Networks which provides us with a Java-based software interface to perform policy-based load balancing, and policy-based resource reservations.
- re-design or change some portions of our original Diff-Serv service models, as we get more insights from our experience with the Iceberg test-bed, and Accelar software router.

7.2.3 Playout Delay Adjustment Algorithm

We have examined the time correlation of Internet delays, and are convinced that one can design a more robust delay adjustment algorithm by taking into account the delay correlation.

Stage 1: Analysis and Design

The following is a list of tasks that we have already initiated, and will continue to pursue:

- derive a better delay estimates based on delay correlation.
- design a new playout delay adjustment algorithm based on the estimator that we derive.
- evaluate how effective each definition is as an estimate for future jitter, and study how the different definitions of delay jitter can be used in playout delay adjustments.

Stage 2: Simulation Study

- We will verify the solution developed in Stage I via simulation study, first using simulated VoIP traffic based on an on-off Markov model, and secondly, using voice traces collected from Section 4.5.
- The results will be compared to the current algorithms used in vat and Nevot.

Stage 3: Evaluation & Refinement

- If necessary, the result of an adaptive algorithm will be fed to the audio device to let humans judge it.

7.3 Timeline

Figure 10 illustrates a proposed timeline for the work, broken down into quarters. We intend to complete the collection of network and VoIP traces using the Iceberg testbed before the end of 1999. In addition we will initiate the design effort for rate adaptation policy in RTP. During the first half of 2000, we will focus on the simulation study (Stage 2) of Diff-Serv performance, while concurrently working on the analysis and design (Stage 1) of playout delay adjustment algorithms. The rest of the year will be spent on refining the Diff-Serv framework based on the simulation results (Stage 3). At the same time, we will initiate Stage 2 for playout delay adjustment algorithms. Towards the end of 2000 and the beginning of 2001, we will carry out a detail evaluation & redesigning of the adaptive algorithms (Stage 3).

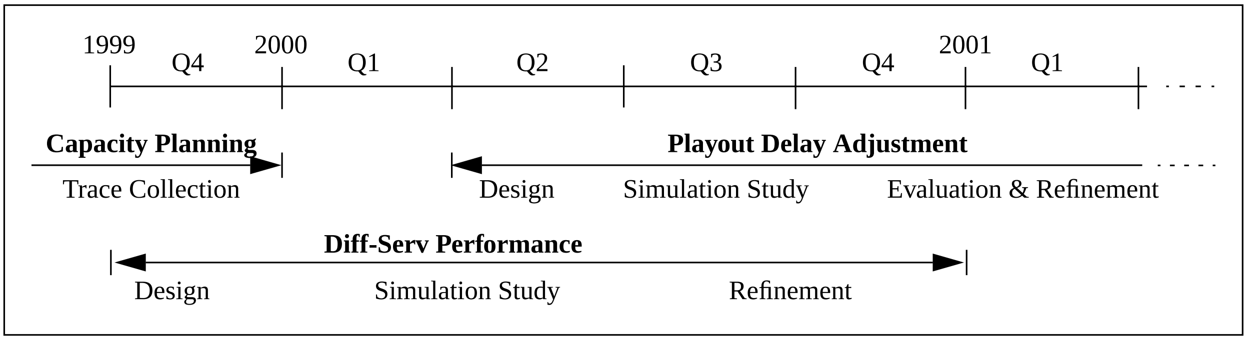

Figure 10. Proposed timeline for this research.

Acknowledgment

We would like to thank Dr. B. Lyles from Sprint Labs, L. Salgarelli and Dr. R. Nagarajan from Lucent Technology for sharing their expertise in this field. Special thanks are dedicated to Professor A. Joseph and other ICEBERG group members at Berkeley for their constructive feedback and enlightening discussions.

References

[1] W. A. Montgomery, “Techniques for packet voice synchronization,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no.6, pp. 1022-1028, December 1983.

[2] D. Clark, S. Shenker, and L. Zhang, “Supporting real-time applications in an integrated services packet network: Architecture and mechanism,” *Proc. ACM Sigcomm’92*, pp. 14-26, 1992.

[3] F. Alvarez-Cuevas, M. Bertran, F. Oller and J. M. Selga, “Voice synchronization in packet switching networks,” *IEEE Networks Magazine*, 7(5):20-25, September 1993.

[4] J. DeTreville and D. Sincoskie, “A distributed experimental communications system,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, pp. 1070-1075, 1983.

[5] H. Schulzrinne, “Voice communication across the Internet: A Network Voice Terminal,” *Technical report*, Dept. of CS, University of Massachusetts, Amherst, MA, July 1992.

[6] V. Hardman, M.A. Sasse, M. J. Handley, A. Watson, “Reliable Audio for Use over the Internet,” *Proc INET’95*.

[7] R. Cox and P. Kroon, “Low bit-rate speech coders for multimedia communications,” *IEEE Communications Magazine*, pp. 34-40, Vol. 34, No.2. December 1996.

[8] S. Fosse-Parisis Bolot, D. Towsley, “Adaptive FEC-Based Error Control for Interactive Audio in the Internet,” *IEEE Infocom’99*, New York, USA, March 1999.

[9] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” *IETF RFC 1889*, January 1996.

[10] Integrated Services Working Group: http://www.ietf.org/html.charters/intserv-charter.html

[11] S. Shenker and J. Wroclawski, “General Characterization Parameters for Integrated Service Network Elements,” *IETF RFC 2215*, September 1997.

[12] Differentiated Services Working Group: http://www.ietf.org/html.charters/diffserv-charter.html

[13] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W.Weiss, “An Architecture for Differential Services”, *IETF RFC 2475*, December 1998.

[14] S. Shenker, C. Partridge, R. Guerin, “Specification of Guaranteed Quality of Service”, *IETF RFC 2212*, September 1997.

[15] J. Wroclawski, “Specification of the Controlled-Load Network Element Service,” *IETF RFC 2211*, September 1997.

[16] D. D. Clark, S. Shenker, and L. Zhang, “Supporting Real-Time Applications in an Integrated Services Packet Architecture and Mechanism,” *Proc. ACM Sigcom’92*, pp.14-26, August 1992.

[17] B. Braden, L. Zhang, S. Berson, S. Jamin, “Resource ReSerVation Protocol (RSVP),” *IETF RFC 2205*, September 1997.

[18] K. Nichols, S. Blake, F. Baker and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, *IETF RFC 2474*, December 1998.

[19] V. Jacobson, K. Nichols and K. Poduri, “An Expedited Forwarding PHB,” *IETF RFC 2598*, June 1999.

[20] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, “Assured Forwarding PHB Group,” *IETF RFC 2597*, June 1999.

[21] D. Clark, “Adding Service Discrimination to the Internet,” *LCS MIT Technical Report TPRC2-0*, September, 1995.

[22] K. Nichols, V. Jacobson and L. Zhang, “A Two-Bit Differentiated Services Architecture for the Internet,” *Internet draft: draft-nichols-diff-svc-arch-00.txt*, November 1997.

[23] B. Gleeson et al., “A Framework for IP Based Virtual Private Networks,” *IETF Internet Draft <draft-gleeson-vpn-framework-00.txt>*, September 1998.

[24] E. Clark, “SLAs: In Search of the Performance Payoff,” *Network Magazine*, May 1999.

[25] M. May, J. Bolot, A. Jean-Marie and C. Diot, “Simple Performance Models of Differentiated Services Schemes for the Internet,” *Proc. IEEE Infocom’99*, March 99.

[26] H. Naser, and O. Aboul-Magd, “Voice over Differentiated Services,” *IETF Internet Draft <draft-naser-voice-diffserv-eval-00.txt>*, 1999.

[27] S. Sahu, D. Towsley, J. Kurose, “Quantitative Study of Differentiated Services for the Internet”, *UMASS CMPSCI Technical Report 99-09*, 1999.

[28] I. Yeom, A. L. Narasimha Reddy, “Realizing Throughput Guarantees in a Differentiated Services Network,” *Proc. IEEE Conference on Multimedia Computing and Systems*, vol. 2, pp. 372-6, June 1999.

[29] A. Stephenson, “Diffserv and MPLS: a quality choice,” *Data Communications International*, vol. 27, no. 17, pp. 73-7, Nov 1998.

[30] H. Adiseshu, G. Parulkar and T. Yavatkar, “A State Management Protocol for IntServ, DiffServ and Label Switching,” *Proc. IEEE ICNP’98*, pp. 272-81, October 1998.

[31] N. Seddigh, B. Nandy, P. Pieda, J. Hadi Salim and A. Chapman, “Experimental Study of Assured Services in a Diffserv IP QoS Network,” *Proc. SPIE*, pp. 217-30, 1998.

[32] J. Ibanez, and K. Nichols, “Preliminary Simulation Evaluation of an Assured Service,” *IETF Internet Draft <draft-ibanez-diffserv-assured-eval-00.txt>*, August 1998.

[33] Y. Moret, S. Fdida, “A Proportional Queue Control Mechanism to Provide Differentiated Services,” *ISCIS’98*, pp. 17-24, October 1998.

[34] J. Bolot, “End-to-end packet delay and loss behavior in the Internet,” *Proc. ACM Sigcomm’93*, pp. 289-298, September 1993.

[35] D. Sanghi, O. Gudmundsson, A. Agrawala, and B. N. Jain, “Experimental assessment of end-to-end behavior on Internet,” *Proc. IEEE Infocom’93*, pp. 867-874, April 1993.

[36] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks,” *Proc. IEEE Infocom’94*, vol. 2, pp. 680-8, April 1994.

[37] S. B. Moon, J. Kurose, and D. Towsley, “Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms,” *ACM/Springer Multimedia Systems*, vol. 6, pp. 17-28, January, 1998.

[38] ITU-T Recommendation G. 114, “General Characteristics of International Telephone Connections and International Telephone Circuits: One-Way Transmission Time,” February 1996.

[39] S. Kent and R. Atkinson, “Security Architecture for the internet Protocol,” *RFC 2401*, November 1998.

[40] B. Gleeson, A. Lin, J. Heinanen, and G. Armitage, “A Framework for IP BAsed Virtual PRivate Networks,” *Internet draft: draft-gleeson-vpn-framework-00.txt*, September 1998.

[41] Chen-Nee Chuah and Randy Katz, “Network Provisioning and Resource Management for IP Telephony,” submitted to ICNP’99. Full paper can also be found in *Report No. UCB//CSD-99-1061*, CS Division, University of California, Berkeley, August, 1999.

[42] H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, Princeton, N.J., 1946.

[43] S. Keshav, “REAL: A Network Simulator”, *Computer Science Department Technical Report 88/472*, UC Berkeley, 1988.

[44] ITU-T Study Group 16, “ITU-T Recommendation H.323 - Packet Based Multimedia Communications Systems,” *ITU-T H-Series Recommendations H.323,* Audiovisual and Multimedia Systems, February 1998.

[45] A. Demers, S. Keshav, and S. Shenkar, “Analysis and Simulation of a Fair Queueing Algorithm,” *Journal of InterNetworking Research and Experiment*, Vol. 1, pp. 3-26, October, 1990.

[46] A. K. Parekh, and R. Gallager, “A Generalized Processor Sharing Approach to Flow COntrol in Integrated Services Networks - The Single NOde case,” *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 344-357, June 1993.

[47] H. Zhang, “Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks,” *Proc. IEEE*, vol. 83, no. 10, October 1995.

[48] L. Braslau, S. Shenker, “Is Service Priority Useful in Networks?” *Proc. of ACM Sigmetrics*, pp. 66-77, June 1998.

[49] L. Braslau, S. Shenker, “Best-Effort versus Reservations: A Simple Comparative Analysis,” *Proc. ACM Sigcomm’98*, pp. 3-16, October 1998.

[50] F. Alvarez-Cuevas, M. Bertran, F. Oller and J. M. Selga, “Voice Synchronization in Packet Switching Networks,” *IEEE Networks Magazine*, 7(5), pp.20-25, September 1993.

[51] J. Rosenberg, “Slides on Internet Loss Measurements,” presented to the Voice on the Net Conference, Boston MA, Sep. 25, 1997 (http://www.cs.columbia.edu/~jdrosen/).

[52] I. Wakeman, A. Ghosh, J. Crowcroft, V. Jacobson, S. Floyd, “Implementing real time packet forwarding policies using Streams,” *Proc. USENIX Technical Conference’95*, pp.71-82, 1995.

[53] A. Floyd, and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol.1, no. 4, pp. 397-413, August 1993.

[54] D. Clark and J. Wroclawski, “An Approach to Service Allocation in the Internet,” *Internet draft, draft-clark-different-svc-alloc-00.txt*, July 1997.

[55] Z. Wang and J. Crowcroft, “Analysis of burstiness and jitter in real-time communications,” ACM Sigcomm’93, pp. 13-19, September, 1993.

[56] A. Banerjea, D. Ferrari, B.A. Mah, M. Moran, D.C. Verma, H. Zhang, “The Tenet Real-time Protocol Suite: Design, Implementation, and Experiences,” *IEEE/ACM Transactions on Networking*, vol. 4, noo.1, pp. 1-10, February, 1996.

Appendix A: Overview of Diff-Serv Network Architecture

The following describes the assumptions we make about the Diff-Serv network architecture. These assumptions are based upon projections of technical advances and the publicly available documentation of proposed commercial products.

Diff-Serv Architecture

Diff-Serv uses edge-based packet-marking and per-class queue management to support multiple service level agreements (SLAs) over an IP-based network. An SLA is a contract between customer and network provider for a service, and may include packet classification and re-marking rules, traffic profiles, actions to traffic streams which are in or out of profile, reliability, security, routing constraints, pricing and billing, etc. Here is a quick overview of several major Diff-Serv components.

1. Edge Mechanisms

The following functionality is required from the Diff-Serv boundary routers which reside at the ingress and egress points to and from the Diff-Serv IP core network:

- **Classification** Edge components classify and set the DS field in packet header based on SLAs.
- **Monitoring** A monitoring interface collect statistics regarding traffic carried at various Diff-Serv service levels, which are important for accounting purposes and for traffic policing.
- **Traffic policing and marking** Edge routers can compare the statistics collected against the SLAs that specify an upper bound on the amount of traffic that senders negotiate to send in the specified service classes. The packets that are within the profile are *in-profile* and the excess packets are *out-profile*. The edge routers can choose to either discard the *out-profile* packets, or mark the packets accordingly before forwarding them into the network [20].
- **Traffic Engineering** If necessary, edge routers interact with the Diff-Serv admission control component in order to coordinate resource requests. Admission policy is subjected to the constraints of the local resources and PHBs that are requested.
- **Traffic Shaping** Ingress routers may delay certain packets to make the particular flow compliant with a traffic profile. Egress routers may shape behavior of aggregate traffic before it is submitted to a subsequent provider’s network.

2. Packet Classifier

Classification is a necessary function for Diff-Serv routers that treat certain traffic differently from other traffic. It can be implemented in various degrees of granularity (for each source-destination pair, or for a specific service class). Each Diff-Serv router must have a packet classifier that selects packets based on the DS-byte and decide which queues to insert these packets into.

3. Scheduler and Queue Management

Scheduling policy is part of queue management which decides which packet to transmit next. It can choose a packet to transmit from a single queue on first come first serve basis, or from multiple queues e.g. Class Based Queueing (CBQ) [52], in a modified round robin fashion. For example, in CBQs with priority scheduling, the lower priority queues receive service only when the all the higher priority queues are empty. Another commonly used scheduler is Weighted Fair Queueing (WFQ) ([45]-[47]).

The buffer management scheme is responsible for putting packets in a queue as they arrive and decide which packets to drop when buffer memory is exceeded. Examples include Drop-tail, Drop-front, Drop from longest queue and Random Early Detection (RED) [53]. RIO [54] extends RED to handle two classes of packets and includes two sets of drop parameters, one for *In* packets and one for the *Out* packets, and service discrimination between the two classes can be achieved either by using two different thresholds, or using different dropping policies. For instance, *Assured Service* can be implemented by sending all packets to an *Assured Queue* managed by RED with In and Out - RIO.

Appendix B: VoIP over Diff-Serv

The origin of voice traffic can be PSTN phones, cellular phones in digital wireless access networks (GSM, CDMA etc.) or even multimedia applications such as vat or NetMeeting from Local Area Networks (LANs). Each of them can use different voice codecs and transmission formats. Therefore, we assume that IPTGs perform some/all of the following functions when necessary to convert input voice traffic to RTP packets:

- coding (PCM, ADPCM, DVI, GSM format etc.)
- silence suppression
- packetization (collect enough voice frames to form a packet), and
- convert into RTP/RTCP packets with the TOS bytes properly marked in IP header.

We assume that the voice samples are packetized and carried over RTP/UDP/IP protocol stack. RTP[9] is a general-purpose real-time data streaming protocol that provides intra- and inter-media payload synchronization, payload identification and sequencing. RTCP is the transport control protocol that enables application to send/receive traffic statistics, jitter and loss estimates. If the packet loss and delay jitter estimates are obtained from RTCP sender/receiver reports, the applications can choose to tolerate the degradation in QoS or terminate the communication. This is equivalent to “dropping” a call in PSTN terminology.

Appendix C: Derivation of C_v in Section 4.3

Each VoIP source is modeled as an on-off Markov process as described in Section 3.2. For our analysis, we assume the random processes $X_i(t)$ are i. i. d. (independent & identically distributed). The total rate of the aggregate traffic is $Y(t)$:

$$Y(t) = \sum_{i=1}^{N_A} X_i(t) \tag{1}$$

where N_A is the number of actual voice calls in progress. The maximum possible value of N_A is N, which is the maximum number of possible voice sources in the network. For example, N in a VPN can be the total number of telephone handsets or other end devices that are capable of generating voice traffic.

At any particular time instant, say $t = T$, $X_i(T)$ are just discrete time random variables. We assume that the random process $X_i(t)$ is ergodic, i.e., time averages see the ensemble averages, and stationary, i.e., $X_i(T)$ have the same statistics at any time instant T. Since $Y(t)$ is the sum of i. i. d. stationary processes, $Y(t)$ is also stationary. At any time instant, $t = T$, $Y(T)$ is just the same of i. i. d. random variables. For simplicity, we omit the time dependence and use the notations X_i, Y instead. Assume the stationary distribution of X_i is:

$$P(X_i = x) = \begin{cases} \dfrac{\lambda}{\lambda+\mu} & \text{when } x = R \\ \dfrac{\mu}{\lambda+\mu} & \text{when } x = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We are interested in estimating the minimum capacity C_v that need to be allocated to VoIP so that the loss rate per flow is less than δ. Assume scheduler/server is work conserving and non-preemptive. If the queues are bufferless, then losses occur when the sum of arrival rates is greater than the rate at which the queue is served: $Y > C_{\mathrm{v}}$. We made the following approximations:

- Consider the worst case where all the potential voice users have calls in progress, i.e., $N_{\mathrm{A}} = N$.
- As soon as the aggregate rate Y exceeds the server rate, information (in bits) from some connections are lost. Let say this happens with probability $P(Y > C_{\mathrm{v}}) = \delta$. The losses can be shared by some connections, or in the worst case, the losses may happen to only one connection. Assume that in any cases when losses happen, the worst loss rate suffered by an individual voice flow is not more than the total loss rate, δ.
- To achieve satisfactory voice quality, loss rate per source has to be bounded: $\delta \le \delta_{\max}$.

Therefore, the following worst case constraint has to be satisfied:

$$P(Y > C_{\mathrm{v}}) \le \delta_{\max} \tag{3}$$

$$\Rightarrow P\left(\sum_{i=1}^{N} X_i > C_{\mathrm{v}}\right) \le \delta_{\max} \tag{4}$$

When N gets large, Y tends to have a normal (Gaussian) distribution under the *Central Limit Theorem*[42]. In short, the Central Limit Theorem states that the sum of a large number of independent observations from any distribution tends to have a normal distribution, and this is true for observations from all distributions. Given that X_i's are i. i. d. with mean m and variance σ^2 (this easily can be determined from Eq. (2)), and Y is the sum of X_i's, the mean and variance of Y is simply the sum of the mean and the sum of the variance of X_i's, respectively. Therefore Y is normal distributed with mean Nm, and variance $N\sigma^2$, $Y \sim N(Nm, N\sigma^2)$. To solve Eq. (3), one can use the well-known Q-function:

$$\begin{aligned} P(Y > C_{\mathrm{v}}) \le \delta_{\max} &\Rightarrow P\left(\frac{Y - Nm}{\sqrt{N} \cdot \sigma} > \frac{C_{\mathrm{v}} - Nm}{\sqrt{N} \cdot \sigma}\right) \le \delta_{\max} \\ &\Rightarrow P\left(Z \ge \frac{C_{\mathrm{v}} - Nm}{\sqrt{N} \cdot \sigma}\right) \le \delta_{\max} \qquad \text{(since Z is continuous)} \\ &\Rightarrow Q\left(\frac{C_{\mathrm{v}} - Nm}{\sqrt{N} \cdot \sigma}\right) \le \delta_{\max}, \end{aligned} \tag{5}$$

where Z is a normalized zero mean unit variance normal random variable, and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{\left(\frac{-t^2}{2}\right)} dt \tag{6}$$

Using the inverse Q-table, one can determine the value of C_{v} for any given $\delta_{\max}$:

$$C_v \ge Nm + Q^{-1}(\delta_{\max}) \cdot \sqrt{N} \cdot \sigma \tag{7}$$

Appendix D: ICNP paper

Please refer to a separately attached paper that we submitted to ICNP'99. The full paper can also been found in the following report:

- Chen-Nee Chuah and Randy Katz, "Network Provisioning and Resource Management for IP Telephony," *Technical Report No. UCB//CSD-99-1061*, CS Division, University of California, Berkeley, August 1999.