Performance Evaluation of the Architecture for End-to-End Quality-of-Service Provisioning

Katsuyoshi Iida and Kenji Kawahara, Kyushu Institute of Technology
Tetsuya Takine, Kyoto University
Yuji Oie, Kyushu Institute of Technology

ABSTRACT
Real-time communications services over the Internet need a new architecture to meet their required quality. From a viewpoint of quality of service provisioning architecture, the Internet can mainly be divided into three types of subnetworks: domain networks, access networks, and stub networks. In this article we focus on issues arising in the former two networks for end-to-end QoS provisioning. First, the access networks are of rather low-speed links, so delay is still of major concern. We examine the statistical delay bound through numerical results derived from our analysis. Schemes to reduce delay are proposed, and their performance is evaluated. Next, domain networks are likely to be of very high-speed links, which can accommodate a huge number of voice flows of low bit rates. Thus, effective flow management will be of major concern because per-flow management is a very costly proposition. Therefore, we pay attention to a flow aggregation scheme, and evaluate its performance by analyzing its blocking probability.

INTRODUCTION

The Internet has significantly changed in qualitative as well as quantitative features, such as transmission capacity. In particular, multimedia real-time communications services over the Internet require research on their qualitative aspects and new architectures for quality of service (QoS) provisioning. Internet telephony is one promising service because voice communications is a widespread traditional real-time communications service, and it can further provide a method for voice communications among various devices connected to the Internet.

Voice traffic can be relayed over several networks of different link capacity; there are mainly three types of networks: domain networks, access networks, and stub networks, as illustrated in Fig. 1. The Internet service providers’ (ISPs’) domain networks, which are the backbone of the Internet, must have high transmission capacity — for example, using asynchronous transfer mode (ATM) and/or wavelength-division multiplexing (WDM) technologies — whereas access networks provide access to the domain networks. Some of the access networks will still have relatively low transmission capacity simply because low-capacity links are good enough for voice traffic. On the other hand, the stub networks represent networks of relatively small size and large transmission capacity, such as internal networks of a building or campus.

The fundamental mechanism to provide QoS there is standardized by Internet Engineering Task Force (IETF) integrated services (IntServ), Resource ReSerVation Protocol (RSVP), and other working groups. However, there are still two issues that remain unsolved in this context. One is how to handle a vast number of voice flows carried over the high-transmission-capacity domain network. For example, a link of 10 Gbit/s can accommodate as many as 1 million voice flows of 10 kbit/s. The delay experienced by the voice packets there will be negligible due to its high-speed packet forwarding capability; thus, only the bandwidth assignment will be of major concern. Another issue is how to guarantee the delay requirement over the low-transmission-capacity access network, in which the delay experienced could be greater than the requirement without adequate control due to the networks' low transmission capacity. Since the number of voice flows handled there is very small, complicated management for guaranteeing delay is acceptable. Consequently, as QoS parameters, bandwidth and delay are of major concern in domain and access networks, respectively.

From the above discussion, we will attack two issues in what follows. First, the performance of delay experienced by voice packets over a link of relatively low transmission capacity will be examined by means of our exact analysis. Our analysis provides an explicit expression for the delay distribution, and thus allows call admission control (CAC) based on the statistical rather than the deterministic delay bound. In our analysis, voice packets have priority over other packets such as
Transmission Control Protocol (TCP) packets. The impact of the TCP packet length on the delay performance of voice packets will also be investigated. Next, we will study an efficient way to handle many voice flows: a flow aggregation technique in which the routers manage the flows as trunks consisting of many flows. This successfully reduces the number of trunks to be treated, but can cause high blocking probability due to inefficient bandwidth assignment compared to per-flow management. For keeping low and acceptable blocking probability, we will examine a bandwidth management scheme called provisioned capacity, which refers to the capacity actually provided to the real-time traffic class, which must be greater than that required on an average basis. We will give numerical results on the blocking probability by our exact analysis, and show how the flow aggregation can be done effectively while avoiding inefficient bandwidth assignment.

Through our examinations, we provide an architecture for end-to-end QoS provisioning. The following sections are as follows. The next section treats low-speed access networks. We then focus on high-speed backbone networks, and finally conclude this article.

**Delay Performance in Access Networks**

Relatively low-speed (e.g., T1) links are likely to still be used even in future access networks, as mentioned earlier. Packets can experience very large delay over those links; hence, it is practical to reduce delay there. Thus, the related issues are being discussed in the IETF IntServ-specific link layers (ISSLL) working group.

As solutions to reducing delay, two technologies have been proposed in the IETF: header compression and segmentation [1]. Header compression is applied to voice packets, which reduces the overhead of voice packets, leads to efficient use of network resources such as intermediate routers and links, and can reduce delay as a result. On the other hand, segmentation is applied to TCP packets sharing network resources with voice packets, which makes TCP packet transmission time smaller and in turn the waiting time of voice packets smaller, as mentioned below.

First, header compression is particularly effective for low-bit-rate voice codecs. For example, the G.723.1 codec generates a frame of 24 bytes at intervals of 30 ms [2; 3, p. 110]. Each frame is conveyed over the Internet on the payload of an Internet Protocol (IP) packet, which is usually called an IP datagram. It is very likely that each IP packet includes at least a header of 40 bytes in IPv4, which consists of IP, User Datagram Protocol (UDP), and Real-Time Protocol (RTP) headers, as shown in Fig. 2. This results in inefficient use of network resources: the effective utilization is only 0.375. The header size can be significantly reduced to only 2 bytes without header cyclic redundancy check (CRC) or four bytes with CRC by use of the compression scheme developed so far.

Next, even if voice packets have priority over non-real-time packets, they will be forced to wait until the transmission of non-real-time packets currently being served ends. Therefore, large non-real-time packets or TCP packets should be segmented in this context [1]. This enables voice packets to be interleaved with segmented TCP packets of small size, resulting in small waiting times for voice packets.

In what follows, considering the impact of the above features, we examine the performance of queuing delay of voice packets in a router or multiplexer by means of numerical results derived from our exact analysis of the following model. Voice packets, called constant bit rate (CBR) packets below, have priority over non-real-time packets, called TCP packets from now on only for convenience. More precisely, N independent CBR flows and TCP flows share the multiplexer. If there are any CBR packets waiting, the multiplexer continues to transmit them; otherwise, TCP packets will be transmitted. Non-preemptive priority service discipline is employed so that CBR packets will be forced to wait until the transmission of currently served TCP packets ends. Furthermore, we deal with a case where at

---

**Figure 1.** An architectural model of the Internet.

**Figure 2.** Encapsulation of an IP datagram.
The dotted line is added in Fig. 3b due to a numerical inversion, which can result in inaccurate numerical outputs and further needs computation of a large amount of time, particularly for large N. On the other hand, when the service time is constant, we can obtain the explicit expression for the waiting time distribution function [4]. The equations described in [4] are quite complicated, so we only show the numerical results.

First, Fig. 3a shows numerical results regarding the impact of CBR link shares on the queuing delay of CBR packets, where the CBR link shares refer to a ratio of the bandwidth used by all CBR flows to the total link capacity, which is set to 1536 kb/s here. A G.723.1 voice codec of 6.4 kb/s is used, and each CBR packet consists of a compressed header of 4 bytes and a payload of 24 bytes. As a result, a CBR flow arrives at the multiplexer at a rate of 7.47 kb/s. The 1536 kb/s link accommodates a maximum of 205 CBR flows at a link share of 1.0 (1536/7.47 = 205.7). The figure illustrates three kinds of performance measure; one is the maximum delay time or the strict upper bound for the delay, called the deterministic delay bound. The second is a statistical delay bound, which can be obtained by the delay time distribution function shown in [4]. A 99.9-percentile delay bound is given in Fig. 3 as a statistical delay bound. It has been reported (e.g., in [5]) that a voice codec can tolerate packet losses of at most 10 percent. This allows the 90-percentile delay bound to be used for guaranteeing acceptable QoS. For this reason, the statistical delay bound is of practical importance.

The third is the average delay time, which is just for reference. From Fig. 3a, we can see that the maximum delay is heavily affected by the CBR link shares, while the 99.9-percentile delay is almost insensitive to it. To explain this, we give a queuing delay density function in Fig. 3b. The density function, which has been obtained in a link share equal to 1.0, is seriously biased toward a low value of queuing delay. This is because it is very unlikely that packets from most CBR flows arrive simultaneously because the packet arrival process from CBR flows is independent and identically distributed. The figure shows that there is a significant difference between the deterministic and statistical delay bounds, especially for a large CBR link share or a large number of CBR flows N. Thus, the statistical delay bound is strongly recommended for efficient CAC in the access networks over the deterministic one.

Next, we show the impact of link capacity on queuing delay bound in Fig. 4. The CBR link share is fixed at 1.0, which is the worst for CBR delay performance. The delay time decreases as the link capacity increases. The 99.9-percentile delay bound decreases to about 3 ms for a link of more than 5 M b/s, whereas the deterministic delay bound is as much as 30 ms even over a link of 10 M b/s. Therefore, the queuing delay of voice packets is negligible from a practical point of view when a link is more than 5 M b/s.

Finally, the impact of the header length on the 99.9-percentile delay bound is examined through Fig. 5. The x-axis indicates the size of actually forwarded TCP packets into which arriving TCP packets are divided; we suppose that segmentation is performed if necessary. Seven voice flows share a link of 128 kb/s, which can be provided by narrowband integrated services digital network (N-ISDN). Currently, TCP packets of more than 500 bytes are very common, and the uncompressed header is now dominant. This results in the delay of more than 40 ms in the figure. According to [6], business users of voice over IP (VoIP) prefer delay in access networks to be less than 10 ms. In this figure, the segmentation and header compression techniques can meet this requirement when a TCP packet is less

---

1 By our analysis, we can obtain only the solid line in Fig. 3b due to a numerical stability condition. The dotted line is added for reference.
than 100 bytes. It is noted that these techniques need related processing capabilities in the routers; hence, the effective throughput of TCP degrades with the decrease of their packet size due to segmentation. The Internet telephony can be achieved at these costs even in the access networks of low transmission capacity.

So far, we have been restricted to homogeneous CBR flows. Please refer to [7] for the performance in heterogeneous environments, in which there are CBR flows of different rates and different packet sizes.

**FLOW MANAGEMENT IN HIGH-SPEED BACKBONE NETWORKS**

Recently, the traffic conveyed over the Internet has been growing explosively, requiring backbone networks of huge transmission capacity. ATM and WDM enable such networks to be built and provide various alternatives for real-time communication over the Internet.

The Internet community has been intensively discussing ways to achieve the real-time communication over the Internet. First, the IntServ architecture and RSVP signaling scheme have been standardized. However, RSVP is based on the per-flow management, which causes a scaling problem [8, p.156]. As a solution, the IETF has discussed a new architecture, called differentiated services (DiffServ) [9]. The DiffServ architecture has introduced the edge concept, in which traffic marking and shaping is done at the edges of the network and intermediate routers constituting high-speed backbone networks process each packet based upon its mark; the related processing is called per-hop behavior (PHB). This architecture allows flow aggregation, that is, a number of traffic flows handled as one unit. Ingress edge routers can enforce the flow aggregation, and egress edge routers perform the flow segregation.

A high-speed link (e.g., 10 Gb/s) can accommodate 1 million voice flows of 10 kb/s as mentioned earlier. Per-flow management is costly in this context, while flow aggregation can reduce the number of units managed independently to acceptable levels. Nevertheless, since it assigns the amount of bandwidth required by one managed unit even if only one flow of the unit is accepted, it leads to inefficient use of the link. As a result, it is expected to suffer a high call blocking probability in CAC compared to the per-flow management as long as available bandwidth is fixed.

In what follows, we study the performance of flow aggregation by analyzing the call blocking probability of a voice flow. We show features related to flow aggregation through numerical results, and propose a way to keep the blocking probability less than some acceptable one.

Figure 6 illustrates our model of two stages for analyzing flow aggregation. A trunk is defined as a unit consisting of a number of flows, and each intermediate router recognizes only trunks instead of flows. First, when a voice flow setup request arrives at an ingress router, it will establish a trunk for an egress router if no trunk for that egress router is available, and the required bandwidth is available on the output link of the ingress router. If the available bandwidth on the link is not enough, the flow setup request will be rejected. The link can accept a maximum of K trunks in Fig. 6. Furthermore, suppose that a trunk is already established. A trunk can accommodate a maximum of, say, m flows, which is the trunk capacity (Fig. 6). If the trunk already includes m accepted flows, the new flow setup request will be rejected; otherwise, it will be accepted. Thus, call blocking can occur in establishing a flow within the already established trunk as well as a new trunk. This leads to a two-stage model, as shown in the figure. In the first stage, the arrivals of flows in a trunk are characterized by a Poisson process, and service times of flows are independent and identically distributed.

The model described above can be exactly analyzed [10]. In this article we only show an overview of the analysis due to limited article length. The first stage can be modeled by an M/G/m/m queuing system, and the second stage can be analyzed by the generalized Engset model.

To give quantitative evaluations, we then show the numerical results using the analysis. In Fig. 7, we illustrate the impact of traffic intensity on the blocking probability in CAC compared to the per-flow management as long as available bandwidth is fixed.
on blocking probability as well as on the number of allocated, arriving, and accepted flows. The x-axis indicates traffic intensity $\rho$ for each trunk. In this figure, each trunk can accommodate a maximum of 100 flows, and there are 100 egress routers. Among them, only up to 80 trunks can be accepted at a time due to link capacity, which means that the link capacity is 8000 flows. From the figure, about 7886 flows are already allocated for trunks at $\rho = 2$, whereas only 182 flows are actually accepted on average over the link. Furthermore, the blocking probability reaches approximately 0.1 at $\rho = 2$, which is not acceptable. The reason this sort of event happens can be explained as follows. Arriving flows go to edge routers evenly. Thus, as the traffic intensity increases, most pairs of ingress and egress routers will need their own trunks, but at most only 80 trunks among 100 egress routers can be accepted. Consequently, flow aggregation causes both high blocking probability and low link utilization.

In order to improve the blocking probability, we have to provide more capacity for the real-time traffic class than that required on average. More specifically, each pair of ingress and egress routers should have a trunk because the blocking probability at the second stage is dominant. At the first stage, each trunk should have a capacity more than the mean arrival traffic rate to decrease the blocking probability to an acceptable level.

In Table 1 we give the provisioned capacity required to keep the blocking probability less than $10^{-4}$. We focus on Internet telephony so that every flow is considered CBR at a rate of 10 kb/s. The number of egress routers is fixed at 100. We then define two terms, the flow aggregation aware system and unaware system, as follows. In the flow aggregation aware system, the edge routers set up the trunks, and each flow is accommodated by one of the trunks. In the unaware system, the edge routers do not set up the trunks, so the intermediate routers have to manage every flow in the unaware system. For example, in this table, if the mean amount of traffic equals 1000 Mb/s, there are 100,000 flows, which should be managed separately, while the aware system manages at most only 100 trunks at the intermediate routers. As shown in the table, the bandwidth of 11.0 Mb/s, which is called the provisioned capacity here, is needed to carry traffic of 10 Mb/s to keep the blocking probability less than $10^{-4}$ in the unaware system, although the unaware system assigns each flow as much bandwidth as required. On the other hand, bandwidth of 24 Mb/s is needed in the aware system. The ratio of the provisioned capacity needed in the flow aggregation aware system to that in the unaware system is given in the table. The ratio is indeed small for larger amounts of traffic. In other word, trunks of stronger intensity or larger capacity make the ratio smaller. From the discussion, we can conclude that flow aggregation can indeed be implemented while keeping the blocking probability within acceptable levels, without wasting significant amounts of bandwidth in future high-speed networks.

**CONCLUSIONS**

We investigate two types of issues in realizing end-to-end QoS provisioning. One is related to delay performance in the access networks of rather low-speed links; the other is an effective flow management scheme in the domain networks of very-high-speed links.

First, we obtain some features of the statistical delay bound by means of our explicit expression for the delay time distribution function, and show that the statistical delay bound is effective in accommodating voice flows efficiently on low-
speed links. We treat some ways to reduce delay, such as header compression and segmentation schemes, and investigate their effect. Next, we deal with flow aggregation as a cost-effective way of flow management on high-speed links. We have evaluated its blocking probability and the amount of extra bandwidth required to keep the blocking less than some acceptable value. End-to-end QoS provisioning needs different approaches in different types of subnetworks. In this article we have investigated effective ways of QoS provisioning for both low-speed access and high-speed domain networks.

### REFERENCES


### BIOGRAPHIES

**KATSUYOSHI IIDA** (iida@cse.kyutech.ac.jp) received a B.E. degree in computer science and electronics from Kyushu Institute of Technology, Iizuka, Japan, in 1996. In 1998 he received an M.E. degree in information science from Nara Institute of Science and Technology, Ikoma, Japan. Since 1998 he has been a Ph.D. candidate at the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include real-time communications in computer networks and the analysis of traffic management. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan and the WIDE project in Japan.

**KENJI KAWAHARA** (kawahara@cse.kyutech.ac.jp) received B.E. and M.E. degrees in computer science and electronics from Kyoto Institute of Technology, Iizuka, Japan, in 1991 and 1993, respectively, and a Ph.D. degree in information and computer science from Osaka University in 1996. From 1995 to 1997 he was an assistant professor in the Information Technology Center, Nara Institute of Science and Technology. From 1997 to 1999 he was an assistant professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. Since April 1999 he has been an associate professor in the same department. His research interests include computer networks, and error and congestion control in high speed networks. He is a member of the IEICE.

**TETSUYA TAKINE** (takine@amp.i.kyoto-u.ac.jp) received B.Eng., M.Eng., and Dr.Eng. degrees in applied mathemat-ics and physics from Kyoto University, Japan, in 1984, 1986, and 1989, respectively. In April 1989 he joined the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, as an assistant professor. Beginning in November 1991, he spent one year at the Department of Information and Computer Science, University of California, Irvine, on leave of absence from Kyoto University. In April 1994, he joined the Department of Information Systems Engineering, Faculty of Engineering, Osaka University as a lecturer, and from December 1994 to March 1998 he was an associate professor in the same department. Since April 1998 he has been an associate professor in the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University. His research interests include queuing theory and performance analysis of computer/communication systems. He is a member of the Operations Research Society of Japan, the IEICE of Japan, the Information Processing Society of Japan, and the Institute of Systems, Control and Information Engineers. He received the 1996 Best Paper Award from the Operations Research Society of Japan.

Yuu Oie (M) (oie@cse.kyutech.ac.jp) received B.E., M.E., and D.E. degrees from Kyoto University, Japan in 1978, 1980, and 1987, respectively. From 1980 to 1983 he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990 he was with the Department of Electrical Engineering, Sasebo College of Technology, Japan. From 1990 to 1995 he was an associate professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka, Japan. From 1995 to 1997 he was a professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997 he has been a professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high-speed networks, and queuing systems. He is a member of the IPSJ and IEICE.

### Table 1

<table>
<thead>
<tr>
<th>Mean amount of traffic for all trunks</th>
<th>Mean # of flows that can be accepted</th>
<th>Ratio of the provisioned capacity of an aggregation-aware system to that of an unaware system</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Mb/s</td>
<td>1000</td>
<td>24.0Mbs/11.0Mbs = 2.18</td>
</tr>
<tr>
<td>100 Mb/s</td>
<td>10,000</td>
<td>137Mbs/103Mbs = 1.33</td>
</tr>
<tr>
<td>1000 Mb/s</td>
<td>100,000</td>
<td>1.10Gbs/1.01Gbs = 1.09</td>
</tr>
</tbody>
</table>

A flow aggregation aware system vs. an unaware system: the effectiveness of provisioned capacity.