Delay analysis for CBR traffic under static-priority scheduling

著者	Iida K, Takine T, Sunahara H, Oie Yuji
journal or publication title	Networking, IEEE/ACM Transactions on
volume	9
number	2
page range	177-185
year	2001-04
URL	http://hdl.handle.net/10228/997

doi: 10.1109/90.917074

Delay Analysis for CBR Traffic Under Static-Priority Scheduling

Katsuyoshi Iida, *Member, IEEE*, Tetsuya Takine, *Member, IEEE*, Hideki Sunahara, *Member, IEEE*, and Yuji Oie, *Member, IEEE*

***Abstract*—We examine the delay performance of packets from constant-bit-rate (CBR) traffic whose delay is affected by nonreal-time traffic. The delay performance is analyzed by solving the $nD/D/1$ queue with vacations. We obtain an exact and closed form solution, hence obviating the need of any approximations or numerical Laplace inversions. We then provide various numerical results for low-bit-rate transmission links, in which packets can experience large delay. From our quantitative evaluation, we conclude that there exists an optimum packet size for a given delay bound. In extremely slow links, such as modem links, transmission control protocol (TCP) packets should be segmented to reduce the CBR delay. We therefore investigate the delay impact of TCP packet sizes as well.**

***Index Terms*—Access networks, constant bit rate, delay analysis, G.723.1, static priority scheduling.**

I. INTRODUCTION

THE FUTURE Internet should support both traditional nonreal-time and real-time services. Examples of the real-time service are interactive video, voice, and other time-stringent applications. Such applications require the network to meet a deadline for delivering packets to destinations. If many packets arrive too late, the quality of the real-time service will deteriorate significantly. Thus, quality-of-service (QoS) control based upon packet delivery time should be supported by the network.

In this paper, we examine the delay performance of packets from constant-bit-rate (CBR) traffic as real-time traffic whose delay can be affected by nonreal-time traffic. More specifically, our model has two types of traffic: superposition of N independent CBR streams and nonreal-time streams. At a multiplexer, multiple CBR streams share a single buffer, whereas nonreal-time streams share another distinct buffer. The CBR buffer has nonpreemptive priority over the nonreal-time buffer. This strategy can be easily implemented and can minimize the effects of nonreal-time traffic on the delay time of packets from CBR streams. We analyze the delay for CBR packets by solving the $nD/D/1$ queue with vacations. The vacation time represents service time of a packet belonging to nonreal-time streams. We obtain an exact and closed form solution of the delay, hence obviating the need of any approximations or numerical Laplace inversions.

Future backbone networks are operated at a very high speed, e.g., gigabits per second. On such networks, the delay time experienced by packets is very low. On the other hand, low-speed links are likely to remain in access networks, for example, 33.6 kb/s links or T1 links. The delay experienced by CBR packets on such links is very large, so that reducing the delay on the links to an acceptable value is of practical importance to provide the real-time communication service. Our analysis enables us to easily calculate a statistical delay bound, which is lighter than a deterministic bound. The difference of those two bounds is oftentimes an order of magnitude. Thus, a connection admission control (CAC) based on a statistical bound can achieve efficient use of the bandwidth. Furthermore, our numerical results show that the size of packets from both types of streams has a great impact on CBR packet delay; longer packets cause longer delay. Conversely, shorter packets degrade the throughput performance because it raises a ratio of the header size to the payload size. Therefore, there might exist an optimal size of packets due to the tradeoff between these factors. We examine this issue in the body of the paper.

Several past works exist about analyzing delay time of a superposition of CBR streams. This problem can be represented by finding the waiting time distribution of $nD/D/1$ queue. Roberts and Virtamo have developed a closed-form formula [1]. Dron and others have proposed an asymptotic formula whose time complexity is independent of the system size N [2]. The formula works well over large N. Ramamurthy and Sengupta have suggested an approximate solution method for $N \leq 100$ [3]. More recently, studies for slotted networks have appeared in the context of asynchronous transfer mode (ATM) networks. Humblet and others have presented an analytical method based on the Ballot theorems [4]. The method provides steady-state delay distribution as well as transient behavior of the system. Privalov

Manuscript received November 11, 1999; revised July 13, 2000; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. de Veciana. The work of T. Takine was supported in part by Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant 12650380. The work of Y. Oie was supported in part by the Research for the Future Program of the Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems" (JSPS-RFTF97R16301), by the Telecommunication Advancement Organization of Japan under the Project "Global Experimental Networks for Information Society Project," and by a Grant-in Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant 12450156. A preliminary version of this paper was presented at the SPIE International Symposium of Voice, Video, and Data Communications, Dallas, TX, November, 1997.

K. Iida is with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma 630-0101, Japan (e-mail: katsu@is.aist-nara.ac.jp).

T. Takine is with the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: takine@amp.i.kyoto-u.ac.jp).

H. Sunahara is with the Information Technology Center, Nara Institute of Science and Technology, Ikoma 630-0101, Japan (e-mail: suna@wide.ad.jp).

Y. Oie is with the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka 820-8502, Japan (e-mail: oie@cse.kyutech.ac.jp).

Publisher Item Identifier S 1063-6692(01)03230-7.

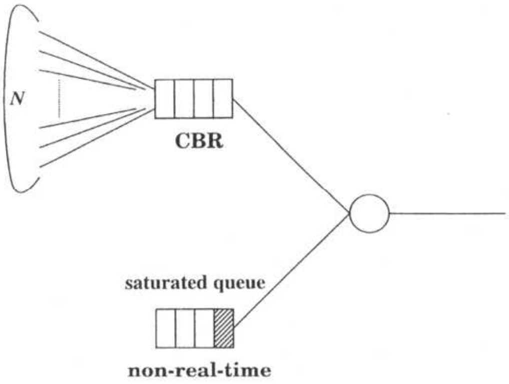


Fig. 1. System overview.

and Sohraby have provided an exact analysis of the jitter process [5]. Their results indicate that jitter variance is bounded and never exceeds the constant 2/3.

Our study has two differences from the cited past research: 1) we consider nonreal-time traffic that can affect CBR delay time and obtain an exact and closed form solution in that context; and 2) we discuss an adequate size for packets based on the tradeoff stated above. These evaluations are for multiple classes of service on low-speed links such as in access networks.

The rest of the paper is organized as follows. In Section II, we describe the system treated here and its mathematical model. In Section III, we analyze the model. In Section IV, we give some numerical results to show a quantitative evaluation of the statistical delay bound. Especially, we focus on the impact of the packet sizes on the delay performance. In Section V, we conclude the paper.

II. System and Model Description

In this section, we provide the system description and the mathematical model used for the system analysis.

A. System Description

At a multiplexer, multiple CBR streams and nonreal-time streams share a bottleneck link as shown in Fig. 1. The packets from CBR streams are required to be transmitted as fast as possible. Therefore, we use priority scheduling to meet this requirement. The unused capacity left by CBR streams should be utilized effectively by nonreal-time streams. Multiple CBR streams share a single buffer, whereas nonreal-time streams share a distinct buffer. The CBR buffer has nonpreemptive priority over the nonreal-time buffer.

Note that a well-known problem of *starvation* arises from the ordinary static-priority scheduling. Starvation prevents lower priority traffic from utilizing the network. This problem can be solved by class-based queueing (CBQ) [6], which is a variation of the ordinary priority scheduling. While CBQ is of practical interest, we do not go into the details of CBQ to get the nature of the simple nonpreemptive priority scheduler.

One of our main objectives is to describe a delay bound for CBR that is tight and suitable for admission control. Therefore, we assume the CBR buffer to be of infinite size in order to get the worst-case bound.

Most of the nonreal-time traffic in the Internet is carried by transmission control protocol (TCP). The congestion control mechanism of TCP is placed at the end nodes, so that TCP sources try to get as much link capacity as possible. Generally speaking, a number of packets from TCP sources are in transit at intermediate nodes. For this reason, we assume the nonreal-time buffer to be saturated in our system, so that at least one packet always exists in the buffer. This is the worst-case scenario for the CBR delay, considering the nonpreemptive priority.

On the CBR side, there are N streams. We assume they are homogeneous in terms of rates and packet sizes.

B. Mathematical Model

We consider a single server queue with a superposition of N independent input streams, where arrivals from each input stream are characterized by an equilibrium renewal process with an interarrival time of D (deterministic). This means that the first arrival from each input stream occurs uniformly in the interval $(0, D)$ and is independent of the other streams. Service times are independent and identically distributed (i.i.d.) with a distribution function $B(x)$. We assume that each service time is less than D/N and the mean service time is denoted by b. Arrivals receive service on a first-come first-serve basis. We call this queue the ordinary queue, where the server is idle when the queue is empty.

Next, we consider a queue with vacations, where arrivals and services have the same characteristics as in the ordinary queue. In the queue with vacations, the server takes a vacation when the queue becomes empty. Vacation times are i.i.d. according to a distribution function $U(x)$ with finite mean u. If the server finds packets waiting in the queue upon returning from a vacation, it serves the packets continuously until the queue becomes empty, and then it takes the next vacation. This service discipline is called exhaustive. If the server finds no packets upon returning from a vacation, it takes another vacation. This vacation discipline is called multiple vacations.

Remark: Each input stream corresponds to a CBR stream. Vacation times correspond to the services of nonreal-time streams. To investigate the worst-case delay scenario for the CBR streams that has (nonpreemptive) priority over the nonreal-time streams, we assume that the service of a nonreal-time packet starts every time the server becomes idle.

III. Analysis

A. General Service Time

We can obtain the waiting time distribution for the queue with vacations by combining known results, stated in this section. In [7], Sengupta analyzed the ordinary queue, where the mean service time b is normalized to one. Suppose the queue is empty at time zero. Let $V_k^*(s)$ and $W_k^*(s)$ denote the Laplace–Stieltjes Transforms (LST) of the virtual and actual waiting time distributions, respectively in the queue without vacations, when the number of arrival streams is equal to k. Following the same argument as in [7], we have for $k = 1, \ldots, N$ and $Nb < D$

$$V_k^*(s) = 1 - kb/D + (kb/D)\frac{1 - B^*(s)}{sb} W_k^*(s)$$

and

$$W_k^*(s) = V_{k-1}^*(s).$$

It then follows from these equations and $V_0^*(s) = 1$ that

$$W_N^*(s) = \sum_{k=0}^{N-1} \frac{(N-1)!}{(N-1-k)!} \left[\frac{1-B^*(s)}{sD}\right]^k \cdot \left(1 - \frac{(N-1-k)b}{D}\right), \qquad Nb < D. \quad (1)$$

Let $W^*(s)$ denote the LST of the actual waiting time distribution in the queue with vacations. Owing to the stochastic decomposition property of the $G/G/1$ queue with exhaustive and multiple vacations [8], we have

$$W^*(s) = \frac{1-U^*(s)}{su} W_N^*(s) \quad (2)$$

where $U^*(s)$ denotes the LST of $U(x)$.

B. Constant Service Time

In order to obtain the explicit expression of the distribution function $W(x)$, we consider the case of constant service time, i.e., $B^*(s) = e^{-sb}$ and constant vacation time, i.e., $U^*(s) = e^{-su}$. Note that the forward recurrence time of the constant service time b has a uniform distribution over $[0, b]$, whose distribution function $\tilde{B}(x)$ is given by

$$\tilde{B}(x) = \begin{cases} \dfrac{b - \alpha_b(x)}{b}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

where

$$\alpha_y(x) = (y-x)^+.$$

Let $\tilde{B}^{(k)}(x)$ denote the k-fold convolution of $\tilde{B}(x)$ with itself.

Lemma 1:

$$\tilde{B}^{(k)}(x) = 1 - \frac{1}{k!b^k} \sum_{n=1}^{k} \binom{k}{n} (-1)^{k-n} \alpha_{nb}^k(x) \qquad k = 1, 2, \ldots, \qquad x \geq 0. \quad (3)$$

The proof of Lemma 1 is given in Appendix A.

Let $F^{(k)}(x)$ denote the convolution of $\tilde{B}^{(k)}(x)$ and the forward recurrence time distribution $\tilde{U}(x) = (u - \alpha_u(x))/u$ of constant vacation time u. We then have

$$F^{(k)}(x) = \int_{x-u+\alpha_u(x)}^{x} \tilde{B}^{(k)}(y) \frac{dy}{u}. \quad (4)$$

Similar to the proof of Lemma 1, we can show that

$$F^{(k)}(x) = 1 - \frac{1}{(k+1)!b^k u} \sum_{n=0}^{k} \binom{k}{n} (-1)^{k+1-n} \cdot \{\alpha_{nb}^{k+1}(x) - \alpha_{nb+u}^{k+1}(x)\}, \qquad k = 1, 2, \ldots$$

which comes from (3) and (4).

From (1) and (2), $W^*(s)$ is given by

$$W^*(s) = \sum_{k=0}^{N-1} \frac{(N-1)!}{(N-1-k)!} \left(\frac{b}{D}\right)^k \left(1 - \frac{(N-1-k)b}{D}\right) \cdot \left[\frac{1-B^*(s)}{sb}\right]^k \frac{1-U^*(s)}{su}. \quad (5)$$

We note here that the factor

$$\left[\frac{1-B^*(s)}{sb}\right]^k \frac{1-U^*(s)}{su} \qquad (k = 1, 2, \ldots)$$

is the LST of $F^{(k)}(x)$. For simplicity, we define $F^{(0)}(x)$ as

$$F^{(0)}(x) = \frac{u - \alpha_u(x)}{u}.$$

It then follows from (5) that

$$W(x) = \sum_{k=0}^{N-1} \frac{(N-1)!}{(N-1-k)!} \left(\frac{b}{D}\right)^k \cdot \left(1 - \frac{(N-1-k)b}{D}\right) F^{(k)}(x), \qquad x \geq 0. \quad (6)$$

Further we note that

$$\sum_{k=0}^{N-1} \frac{(N-1)!}{(N-1-k)!} \left(\frac{b}{D}\right)^k \left(1 - \frac{(N-1-k)b}{D}\right) = 1.$$

Thus (6) is rewritten to be

$$W(x) = 1 - \frac{1}{Nu} \sum_{k=0}^{N-1} \binom{N}{k+1} \left(1 - \frac{(N-1-k)b}{D}\right) \cdot \left(\frac{1}{D}\right)^k \sum_{n=0}^{k} \binom{k}{n} (-1)^{k+1-n} \cdot \{\alpha_{nb}^{k+1}(x) - \alpha_{nb+u}^{k+1}(x)\}, \qquad x \geq 0. \quad (7)$$

Note that (7) consists of terms that alternate in sign and therefore is not numerically stable.

To obtain a numerically stable expression, we rewrite (7) as

$$1 - W(x) = \frac{D}{Nu} \sum_{k=0}^{N-1} \binom{N}{k+1} \sum_{n=0}^{k} \binom{k}{n} (-1)^n \cdot \left\{\left(\frac{-\alpha_{nb}(x)}{D}\right)^{k+1} - \left(\frac{-\alpha_{nb+u}(x)}{D}\right)^{k+1}\right\} - \frac{b}{u} \sum_{k=0}^{N-2} \binom{N-1}{k+1} \sum_{n=0}^{k} \binom{k}{n} (-1)^n \cdot \left\{\left(\frac{-\alpha_{nb}(x)}{D}\right)^{k+1} - \left(\frac{-\alpha_{nb+u}(x)}{D}\right)^{k+1}\right\}. \quad (8)$$

Lemma 2:

$$\sum_{k=0}^{N-1}\binom{N}{k+1}\sum_{n=0}^{k}\binom{k}{n}x_n^{k+1}y_n^n = \sum_{n=0}^{N-1}x_n(x_ny_n)^n\sum_{k=0}^{N-n-k}\binom{k+n}{n}(1+x_n)^k. \quad (9)$$

The proof of Lemma 2 is given in Appendix B. Applying Lemma 2 to (8), we obtain

$$\begin{aligned}1-W(x) &= \frac{D}{Nu}\left[\sum_{n=0}^{N-1}\left(\frac{\alpha_{nb+u}(x)}{D}\right)^{n+1}\sum_{k=0}^{N-n-1}\binom{k+n}{n}\right.\\ &\quad\cdot\left(1-\frac{\alpha_{nb+u}(x)}{D}\right)^k-\sum_{n=0}^{N-1}\left(\frac{\alpha_{nb}(x)}{D}\right)^{n+1}\\ &\quad\left.\cdot\sum_{k=0}^{N-n-1}\binom{k+n}{n}\left(1-\frac{\alpha_{nb}(x)}{D}\right)^k\right]\\ &\quad-\frac{b}{u}\left[\sum_{n=0}^{N-2}\left(\frac{\alpha_{nb+u}(x)}{D}\right)^{n+1}\sum_{k=0}^{N-n-2}\binom{k+n}{n}\right.\\ &\quad\cdot\left(1-\frac{\alpha_{nb+u}(x)}{D}\right)^k-\sum_{n=0}^{N-2}\left(\frac{\alpha_{nb}(x)}{D}\right)^{n+1}\\ &\quad\left.\cdot\sum_{k=0}^{N-n-2}\binom{k+n}{n}\left(1-\frac{\alpha_{nb}(x)}{D}\right)^k\right].\end{aligned}$$

The above expression is numerically stable when $\max(0, (N-1)b+u-D) \le x \le (N-1)b+u$.

IV. Numerical Results

In this section, we show the numerical results obtained through our exact analysis of the delay time distribution. First, we focus on a low-bit-rate voice-coding algorithm, called G.723.1 [9]–[11]. Although its bit rate is 5.3 or 6.4 kb/s, its quality is rated at as high as 3.98 on the mean opinion score (MOS) scale, while that of the traditional 64 kb/s voice coder is rated at 4.0 in the MOS tests [11, pp. 110–111]. Therefore, as far as MOS test is concerned, G.723.1 can be considered an algorithm suitable for low-bit-rate networks. However, G.723.1 suffers from large algorithmic delay: lookahead delay of 7.5 ms, frame delay of 30 ms, and processing delay similar to frame delay. According to ITU-T G.114, the communication quality is good if the total end-to-end delay is less than 150 ms. Therefore, the transmission delay should be minimized to satisfy this end-to-end delay requirement, in particular if G.723.1 is used.

Second, we consider the effect of the packet sizes. In case of CBR traffic, each frame in G.723.1 is usually generated at an interval of 30 ms, so that payload size is 24 bytes. On the other hand, the header size of Internet Protocol/User Datagram Protocol/Real-Time Protocol (IP/UDP/RTP) in IP version 4 is 40 bytes or more [12, p. 34]. Therefore, the size of G.723.1 encoded CBR packets is at least 64 bytes over the Internet. On low-speed networks, the header overhead can severely affect the delay performance. Recently an IP/UDP/RTP header compression scheme has been developed to reduce this large overhead [13]. The compression can reduce the header size to two [without header cyclic redundancy check (CRC)] or four bytes (with header CRC). On the other hand, TCP packet size is typically around 40, 500, or 1500 bytes [14]. Packets of around 40 bytes are acknowledgments. Packets of around 1500 bytes are usually used as data packets when path maximum transfer unit (MTU) discovery function is implemented [12, p. 237]. Otherwise, data packets are usually of size around 500 bytes. In this paper, we assume TCP packets to be of 500 bytes unless stated otherwise. We also discuss the effect of varying the TCP packet size at the end of this section.

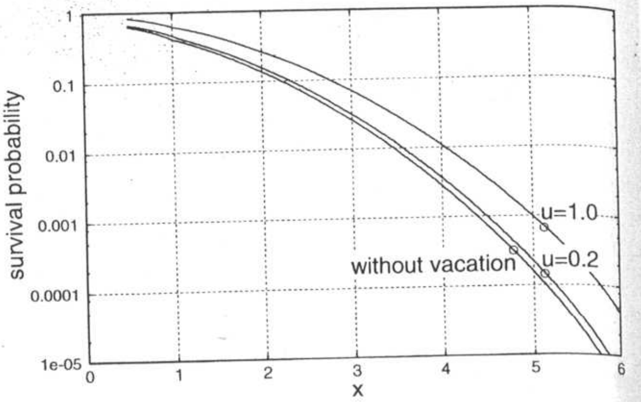


Fig. 2. Comparison between the queue with vacations and the queue without vacations. $N = 9, D = 10, b = 1$.

For these reasons, CBR packet size, TCP packet size, and others are parameterized as follows.

- CBR packet size: 28 bytes (4 byte header + 24 byte payload), 64 bytes (40 byte header + 24 byte payload).
- TCP packet size: 500 bytes.
- Link speed: $L = 33.6, 128, 1536$ kb/s and 10 Mb/s.
- Payload rate of CBR stream: $C = 6.4$ kb/s.
- Rate of CBR stream with header overhead: 7.47 kb/s (4-byte header case) or 17.07 kb/s (40-byte header case).

For simplicity, we introduce the notation CBR $= (4+24)$ bytes, which represents a CBR packet consisting of a 4-byte header and a 24-byte payload.

Before showing numerical results regarding the parameters above, we briefly provide a comparison of delay-time distribution functions of both the queue with vacations and the queue without vacations in Fig. 2. Let the number N of streams be nine, the arrival interval D of each stream be ten, and the constant service time b be one. We choose 0.2 and 1.0 as the constant vacation time u. As mentioned in the previous section, the queue without vacations means the ordinary $nD/D/1$ queue. In this paper, we compute the delay-time distribution of the ordinary $nD/D/1$ queue using the approach for continuous-time arrivals described in [4]. In Fig. 2, the delay time of the queue with vacations is always greater than that of the queue without vacations, and the delay time for small u approaches that of the queue without vacations, as expected.

In Fig. 3, we give an example of the survival function associated with the queueing delay distribution. We assume a link connecting with a modem of link speed L of 33.6 kb/s, and that link is shared by one voice stream of CBR $= (40+24)$ bytes and TCP packets of 500 bytes. A TCP packet transmission takes about 119 ms on the

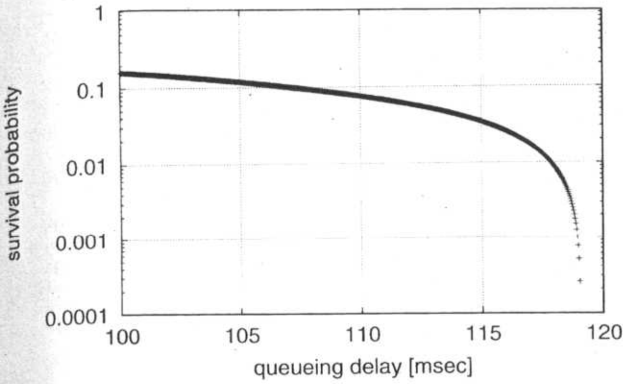


Fig. 3. Example of the survival function on the queueing delay. $L = 33.6$ kb/s, $N = 1, C = 6.4$ kb/s, CBR $= (40 + 24)$ bytes, TCP $= 500$ bytes.

link, so that the delay performance of packets from voice streams can be heavily affected by TCP packets, as shown in the figure. This large delay can be reduced by the header compression and the segmentation of TCP packets. We demonstrate the effect of these schemes at the end of this section.

The link share is defined as a ratio of the total bandwidth of all CBR streams to the link capacity. For example, if a link share is 1.0, $N = 205$ streams of CBR $= (4 + 24)$ bytes are multiplexed because 1536 kb/s divided by 7.47 kb/s is roughly 205.7. We investigate the impact of the CBR link share on the queueing delay in Fig. 4. The 100-percentile delay bound can be calculated in a straightforward way, and that is $(N-1)b+u$; this bound is also called the deterministic delay bound. The 99.9-percentile delay can be calculated by our analysis. This kind of delay performance is called a statistical delay bound here. From the figure, unlike the 99.9-percentile delay, the 100-percentile delay is heavily affected by the CBR link share. This can be explained as follows. It is very unlikely that packets from a large number of CBR streams arrive at the same time because the arrival process from CBR streams is i.i.d. For this reason, by increasing the number of CBR streams, one should not expect a significant impact on the statistical bound.

Next, we investigate the impact of the link capacity on queueing delay in Fig. 5. Since CBR delay performance is the worst when the CBR link share is equal to 1.0, we have fixed the link share to that value. The number N of CBR streams at each amount of link capacity is given in Table I. In this figure, TCP packet size is 500 bytes, each CBR stream is of CBR $= (4 + 24)$ bytes, and CBR interarrival time D is 30 ms. Since the link share is approximately 1.0, Nb is approximately 30 ms and the deterministic delay $(N - 1)b + u$ is always greater than 30 ms because TCP packet transmission time u is larger than CBR packet transmission time b. On the other hand, the 99.9-percentile delay is very small over a wide range of the link capacity. In particular, when the link capacity L is greater than 5 Mb/s, the 99.9-percentile delay is nearly equal to 3 ms, i.e, 1/10 of the interarrival time D. Thus we can say that the queueing delay is small enough to be negligible when the link capacity L is greater than 5 Mb/s. In other words, when more than 669 CBR sources are multiplexed (see Table I), the queueing delay is immaterial in terms of the QoS of CBR streams. Note that it has been reported, e.g., in [10], that the G.723.1 coder can tolerate packet losses of at most 10%. This allows the 90-percentile delay bound to be used for guaranteeing acceptable QoS. Thus, the statistical delay bound is of practical importance. In this paper, we employ the 99.9-percentile delay bound as a very strict one.

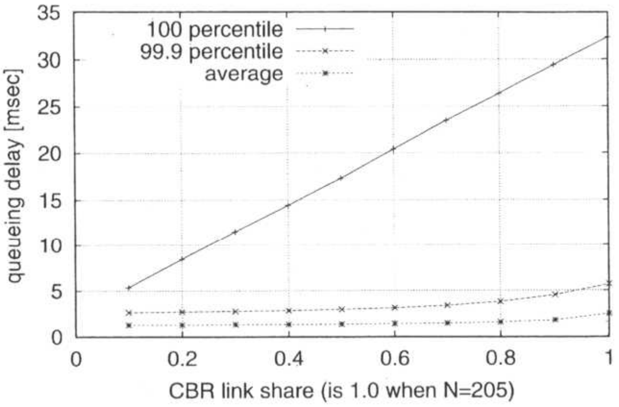


Fig. 4. Impact of the CBR link share on the queueing delay. $L = 1536$ kb/s, CBR $= (4 + 24)$ bytes, TCP $= 500$ bytes.

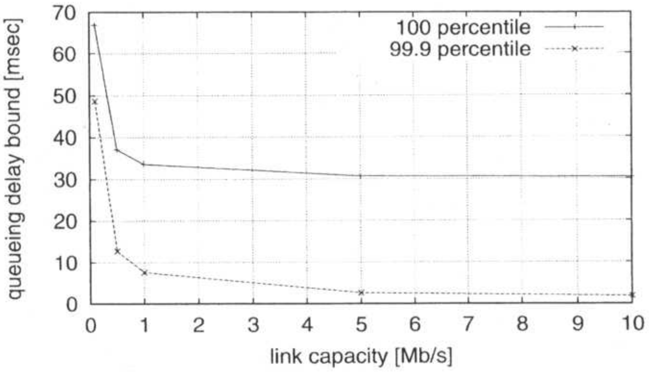


Fig. 5. Impact of link capacity on the queueing delay. CBR link share $= 1.0, C = 6.4$ kb/s, CBR $= (4 + 24)$ byte, TCP $= 500$ bytes.

TABLE I
NUMBER OF CBR STREAMS IN FIG. 5

Link capacity	# N of CBR streams
100Kb/s	13
500Kb/s	66
1Mb/s	133
5Mb/s	669
10Mb/s	1339

So far, we have dealt with CBR packets each including a payload of 24 bytes and a header of 4 bytes. TCP packet size was fixed at 500 bytes. In what follows, the impact of these factors is discussed. First, we show the impact of CBR frame size on the delay performance in Fig. 6. Each CBR frame is created by a voice encoder at a constant time interval. The frame is then packetized into an IP packet with a 4-byte compressed header. The interarrival time of packets from each CBR stream is strongly related with the CBR payload size. For example, a 24-byte frame leads to 30 ms interarrival time D, or *packetization delay*, due to 6.4-kb/s coding rates. A larger payload size results in a larger packetization delay.

Nevertheless, as shown in Fig. 6, the 99.9-percentile queueing delay is almost insensitive to the CBR payload size, whereas the 100-percentile queueing delay linearly increases with the CBR payload size.

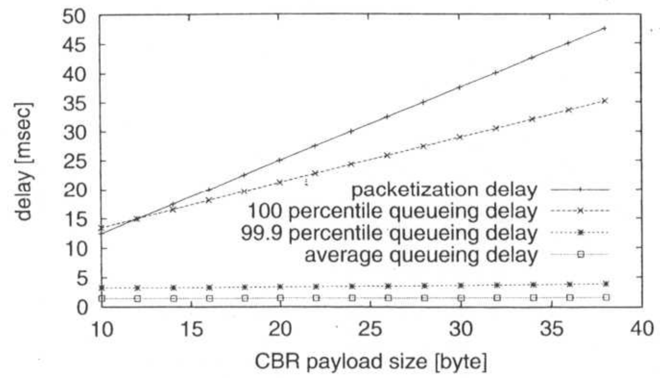


Fig. 6. Impact of CBR frame size on the delay performance. $L = 1536$ kb/s, $N = 150, C = 6.4$kb/s, CBR $= (4 + x)$ bytes, TCP $= 500$ bytes.

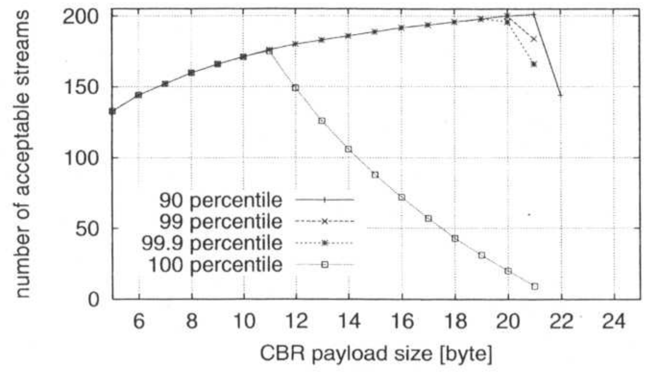


Fig. 7. Impact of the payload size on the number of acceptable streams. 30-ms delay bound, $L = 1536$ kb/s, $C = 6.4$ kb/s, CBR $= (4 + x)$ bytes, TCP $=$ 500 bytes.

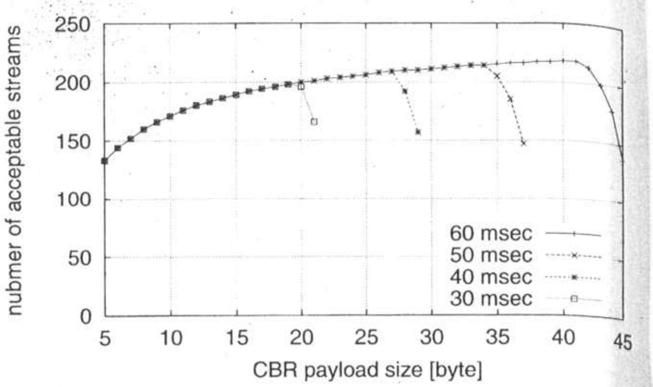


Fig. 8. Impact of the delay bound on the maximum number of acceptable streams. 99.9-percentile delay bound, $L = 1536$ kb/s, $C = 6.4$ kb/s, CBR $=$ $(4 + x)$ bytes, TCP $=$ 500 bytes.

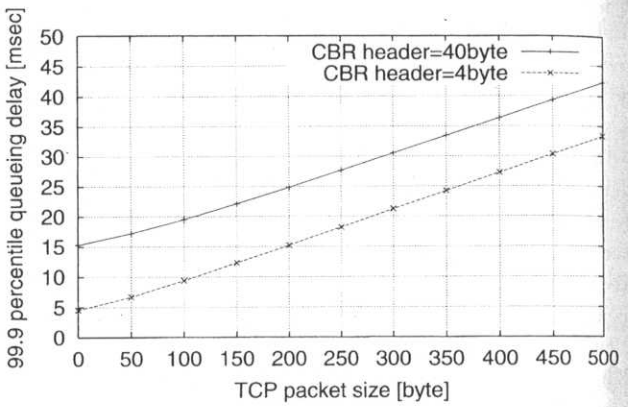


Fig. 9. Impact of the header overhead on the 99.9-percentile delay bound. $L =$ 128 kb/s, $N = 7, C = 6.4$ kb/s, CBR payload size $=$ 24 bytes.

Next, we calculate the CBR payload size so as to minimize the end-to-end delay. Generally, the end-to-end delay consists of several parts: packetization delay, queueing delay, transmission delay, propagation delay, and others. The CBR payload size affects the packetization delay, the queueing delay, and processing delay. The processing delay is due to the computation time to compress the voice signal. For example, a coder chip using digital signal processor (DSP) can process the voice signal within the frame delay. Likewise, four complete coder chips can reduce the processing delay to a quarter of the frame delay [9]. In this way, processing delay depends directly on the coder's implementation. For simplicity, we do not consider the processing delay here.

The queueing delay occurs at each of the intermediate nodes. In the future Internet, most of the intermediate nodes have a high-speed link of, say, gigabits per second, so that the queueing delay is negligible. Nevertheless, access networks still employ low-speed links. For this reason, the queueing delay occurring in the ingress node (an entrance point of the network) and the egress node (an exit point of the network) is not negligible, but should be evaluated precisely. In what follows, we focus on the sum of the queueing delay occurring in the network, which can be regarded as an ingress node or an egress node, and the packetization delay. This sum includes only part of the end-to-end delay. Here, we suppose that the above delay should be less than 30 ms for the required quality, and examine the number of acceptable streams under this condition. In Fig. 7, we illustrate the number of acceptable streams as a function of the CBR payload. For example, a 20-byte payload and a 24-byte payload lead to a packetization delay of 25 and 30 ms, respectively. Thus, a 24-byte payload is not acceptable, and the queueing delay should be less than 5 ms even in the former case. We note that the G.723.1 coder cannot be employed under this condition because the coder uses a 24-byte frame. From the figure, an 11-byte frame is the optimal for the 100-percentile delay bound, which allows 175 CBR streams. On the other hand, a 19-byte frame is optimal for the 99.9-percentile delay bound, which allows 198 streams.

Fig. 8 illustrates the impact of CBR payload sizes on the number of acceptable streams under various delay bounds. A payload of 24 bytes cannot be employed in the G.723.1 standard with 30-ms delay bound, while 40 ms or more delay bounds allow a 24-byte payload, as shown in this figure. Note that the maximum number of acceptable streams is limited to 240 even with large payload size because 1536 kb/s : 6.4 kb/s = 240 : 1.

Next, the impact of the header length on the 99.9-percentile delay bound is illustrated in Fig. 9. The x-axis indicates the size of TCP packets. Seven voice streams share a link of 128 kb/s, which can be provided by Narrowband-Integrated-Service Digital Network (N-ISDN). Note that the delay time performance in the case of TCP packet of zero size is equivalent to that in the queue without vacations, which was computed by [4] and in a similar way in Fig. 2. As we mentioned earlier, TCP packets of about 500 bytes are very common, and the uncompressed header

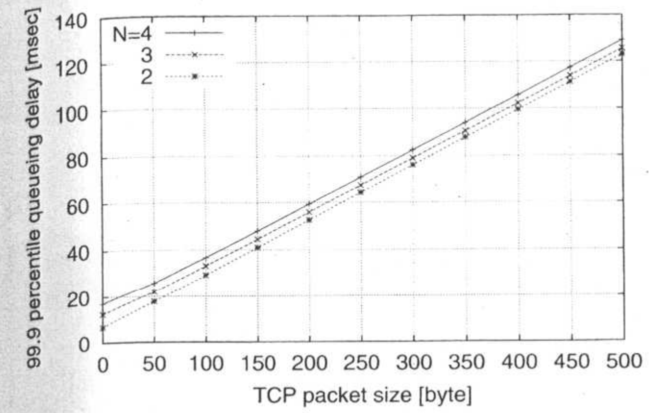


Fig. 10. Impact of TCP packet size on the queueing delay. $L =$ 33.6 kb/s, $C = 6.4$ kb/s, CBR $= (4 + 24)$ bytes.

is now dominant. As seen in the figure, this results in a delay time of more than 40 ms. Thus, in this case, TCP packet is better to be segmented into some shorter packets. We can see that segmentation and header compression techniques are effective to improve delay performance, successfully reducing the delay time within the acceptable level. Notice that there is a need for related processing capabilities in the routers, hence the effective throughput of TCP degrades with the decrease of their packet size due to segmentation. Nevertheless, Internet telephony can be operated at these performance levels even in an access network of low transmission capacity.

Finally, we show that header compression and segmentation of TCP packets can be an effective strategy to reduce the impact of TCP packet size on queueing delays on slow links. In Fig. 10, we provide the impact of TCP packet size on the queueing delay. We suppose $L = 33.6$ kb/s modem link with some voice streams. For TCP packet size of 500 bytes, the delay is too long. On the other hand, small TCP packets or segmented TCP packets, for example 100-byte packets, significantly improve the delay performance. On this link, we can see that the number of voice streams has little impact on the queueing delay. Furthermore, the network can accept four voice streams by using segmentation and header compression, whereas it cannot accept even one voice stream without them (see Fig. 3).

V. Conclusion

We have investigated the delay performance of CBR traffic whose delay is affected by TCP traffic. The delay performance has been analyzed by solving the $nD/D/1$ queue with vacations. An exact and closed form solution has been obtained.

We have presented a number of numerical results. 1) CBR link utilization affects only slightly the 99.9-percentile delay bound while it heavily affects the 100-percentile delay bound. 2) At large link capacity, unlike the 100-percentile delay, the 99.9-percentile delay is small enough to be negligible. 3) The 99.9-percentile delay bound can accept a larger number of streams than the 100-percentile delay bound when CBR payload size is large. 4) Header compression scheme can reduce delay to half in the N-ISDN networks. 5) Large TCP packets should be segmented into small sizes, such as 50 or 100 bytes, in extremely slow networks.

Appendix A
Proof of Lemma 1

We prove (3) by induction. Clearly, (3) holds for $k = 1$. Suppose the equation holds for some $k = k^*$. We then have

$$
\begin{aligned}
&\tilde{B}^{(k^*+1)}(x)\\
&= \int_0^{\min(x,\,b)} \tilde{B}^{(k^*)}(x-y)\frac{dy}{b}\\
&= \int_{x-b+\alpha_b(x)}^{x} \tilde{B}^{(k^*)}(y)\frac{dy}{b}\\
&= \frac{b-\alpha_b(x)}{b} + \frac{1}{(k^*+1)!b^{k^*+1}}\sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{k^*-n}\\
&\quad \cdot\left[\alpha_{nb}^{k^*+1}(x) - \{\alpha_{(n+1)b}(x)-\alpha_b(x)\}^{k^*+1}\right]\\
&= 1 - \frac{1}{(k^*+1)!b^{k^*+1}}\\
&\quad \cdot\left[(k^*+1)!b^{k^*}\alpha_b(x) + \sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x)\right.\\
&\qquad +\sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{k^*-n}\alpha_{(n+1)b}^{k^*+1}(x)\\
&\qquad +\sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{k^*-n}\\
&\qquad \left.\cdot\sum_{m=0}^{k^*}\binom{k^*+1}{m}(-1)^{k^*+1-m}\alpha_{(n+1)b}^{m}(x)\alpha_b^{k^*+1-m}(x)\right]\\
&= 1 - \frac{1}{(k^*+1)!b^{k^*+1}}\\
&\quad \cdot\left[\sum_{n=1}^{k^*+1}\binom{k^*+1}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x) + (k^*+1)!b^{k^*}\right.\\
&\qquad \cdot\alpha_b(x) - (-1)^{k^*}\alpha_b^{k^*+1}(x) + \sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{k^*-n}\\
&\qquad \left.\cdot\sum_{m=0}^{k^*}\binom{k^*+1}{m}(-1)^{k^*+1-m}\alpha_{(n+1)b}^{m}(x)\alpha_b^{k^*+1-m}(x)\right]\\
&= 1 - \frac{1}{(k^*+1)!b^{k^*+1}}\\
&\quad \cdot\left[\sum_{n=1}^{k^*+1}\binom{k^*+1}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x) + (k^*+1)!b^{k^*}\right.\\
&\qquad \cdot\alpha_b(x) - (-1)^{k^*}\alpha_b^{k^*+1}(x) + \sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^{n}\\
&\qquad \left.\cdot\sum_{m=0}^{k^*}\binom{k^*+1}{m}(-1)^{m+1}(\alpha_b(x)+nb)^{m}\alpha_b^{k^*+1-m}(x)\right]\\
&= 1 - \frac{1}{(k^*+1)!b^{k^*+1}}\\
&\quad \cdot\left[\sum_{n=1}^{k^*+1}\binom{k^*+1}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x)\right.\\
&\qquad +(k^*+1)!b^{k^*}\alpha_b(x) - (-1)^{k^*}\alpha_b^{k^*+1}(x)
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{l=1}^{k^*+1}\left\{\sum_{n=1}^{k^*}\binom{k^*}{n}(-1)^n n^{k^*+1-l}\right\}\\
&\cdot\left\{\sum_{m=k^*+1-l}^{k^*}\binom{k^*+1}{m}\binom{m}{k^*+1-l}(-1)^{m+1}\right\}\\
&\left.\cdot b^{k^*+1-l}\alpha_b^l(x)\right]\\
=&1-\frac{1}{(k^*+1)!b^{k^*+1}}\\
&\cdot\left[\sum_{n=1}^{k^*+1}\binom{k^*+1}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x)\right.\\
&+(k^*+1)!b^{k^*}\alpha_b(x)-(-1)^{k^*}\alpha_b^{k^*+1}(x)\\
&+(k^*+1)!(-1)^{2k^*+1}b^{k^*}\alpha_b(x)\\
&\left.-\sum_{m=0}^{k^*}\binom{k^*+1}{m}(-1)^{m+1}\alpha_b^{k^*+1}(x)\right]\\
=&1-\frac{1}{(k^*+1)!b^{k^*+1}}\\
&\cdot\sum_{n=1}^{k^*+1}\binom{k^*+1}{n}(-1)^{k^*+1-n}\alpha_{nb}^{k^*+1}(x),
\end{aligned}
$$

which shows (3) holds for $k = k^* + 1$, too. Note here that we use the equalities

$$\binom{m-1}{n-1}+\binom{m-1}{n}=\binom{m}{n} \tag{10}$$

$$
\begin{aligned}
\alpha_p(x-q+\alpha_q(x)) &= \alpha_{p+q}(x)-\alpha_q(x)\\
\int_p^q \alpha_y^m(x)\,dx &= \frac{-1}{m+1}(\alpha_y^{m+1}(q)-\alpha_y^{m+1}(p))\\
\alpha_{(n+1)b}^m(x)\alpha_b(x) &= (\alpha_b(x)+nb)^m\alpha_b(x)
\end{aligned}
$$

and for $m = 0, \ldots, k$

$$\sum_{n=1}^{k}\binom{k}{n}(-1)^n n^m=\begin{cases}-1, & m=0,\\ 0, & m=1,\ldots,k-1\\ k!(-1)^k, & m=k,\end{cases} \tag{11}$$

in the above computation. Note that (11) can be shown to hold by

$$\sum_{n=1}^{k}\binom{k}{n}(-1)^n n^m=\lim_{z\to 0}\frac{d^m}{dz^m}[(1-e^z)^k-1].$$

Appendix B
Proof of Lemma 2

Note first that the right-hand side of (9) is rewritten to be

$$
\begin{aligned}
\sum_{n=0}^{N-1} x_n(x_ny_n)^n \sum_{k=0}^{N-n-k}&\binom{k+n}{n}(1+x_n)^k\\
&=\sum_{k=0}^{N-1}\sum_{n=0}^{k}\binom{k}{n}x_n(x_ny_n)^n(1+x_n)^{k-n}.
\end{aligned}
$$

Thus we shall prove the following equality:

$$
\begin{aligned}
\sum_{k=0}^{N-1}\binom{N}{k+1}\sum_{n=0}^{k}&\binom{k}{n}x_n^{k+1}y_n^n\\
&=\sum_{k=0}^{N-1}\sum_{n=0}^{k}\binom{k}{n}x_n(x_ny_n)^n(1+x_n)^{k-n}.
\end{aligned}
\tag{12}
$$

Clearly (12) holds for $N = 1$. Suppose it holds for $N = N^*$. Then, by (10),

$$
\begin{aligned}
\sum_{k=0}^{N^*}&\binom{N^*+1}{k+1}\sum_{n=0}^{k}\binom{k}{n}x_n^{k+1}y_n^n\\
&=\sum_{k=0}^{N^*-1}\binom{N^*}{k+1}\sum_{n=0}^{k}\binom{k}{n}x_n^{k+1}y_n^n\\
&\quad+\sum_{k=0}^{N^*}\binom{N^*}{k}\sum_{n=0}^{k}\binom{k}{n}x_n^{k+1}y_n^n\\
&=\sum_{k=0}^{N^*-1}\sum_{n=0}^{k}\binom{k}{n}x_n(x_ny_n)^n(1+x_n)^{k-n}\\
&\quad+\sum_{n=0}^{N^*}\binom{N^*}{n}x_n(x_ny_n)^n(1+x_n)^{N^*-n}\\
&=\sum_{k=0}^{N^*}\sum_{n=0}^{k}\binom{k}{n}x_n(x_ny_n)^n(1+x_n)^{k-n}.
\end{aligned}
$$

The above equation implies that (12) holds for $N = N^* + 1$, too, which completes the proof.

References

[1] J. Roberts and J. Virtamo, "The superposition of periodic cell arrival streams in an ATM multiplexer," *IEEE Trans. Commun.*, vol. 39, pp. 298–303, Feb. 1991.

[2] L. Dron, G. Ramamurthy, and B. Sengupta, "Delay analysis of continuous bit-rate traffic over an ATM network," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 402–407, Apr. 1991.

[3] G. Ramamurthy and B. Sengupta, "Delay analysis of a packet voice multiplexer by the $\sum D_i/D/1$ queue," *IEEE Trans. Commun.*, vol. 39, pp. 1107–1114, July 1991.

[4] P. Humblet, A. Bhargava, and M. Hluchyj, "Ballot theorems applied to the transient analysis of $nD/D/1$ queues," *IEEE/ACM Trans. Networking*, vol. 1, pp. 81–95, Feb. 1993.

[5] A. Privalov and K. Sohraby, "Per-stream jitter analysis in CBR ATM multiplexors," *IEEE/ACM Trans. Networking*, vol. 6, pp. 141–149, Apr. 1998.

[6] S. Floyd and Van Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 365–386, Aug. 1995.

[7] B. Sengupta, "A queue with superposition of arrival streams with an application to packet voice technology," in *Performance'90*, P. J. B. King *et al.*, Eds. Amsterdam, The Netherlands: Elsevier Science, North-Holland, 1990, pp. 53–59.

[8] B. Doshi, "Generalizations of the stochastic decomposition results for single server queues with vacations," *Commun. Statist.-Stochastic Models*, vol. 6, no. 2, pp. 307–333, 1990.

[9] R. Cox, "Three new speech coders from the ITU cover a range of applications," *IEEE Commun. Mag.*, pp. 40–47, Sept. 1997.

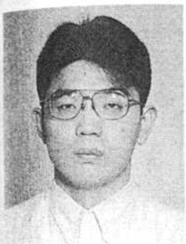
[10] T. Kostas *et al.*, "Real-time voice-over packet-switched networks," *IEEE Network Mag.*, pp. 18–27, Jan./Feb. 1998.

[11] G. Held, *Voice Over Data Networks*. New York: McGraw-Hill, Apr. 1998.

[12] W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison Wesley, 1994, vol. 1.

[13] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP headers for low-speed serial links,", IETF RFC2508, Feb. 1999.

[14] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an internet backbone," in *Proc. ISOC INET'98*, Geneva, Switzerland, July 1998.



Katsuyoshi Iida (M'00) received the B.E. degree in computer science and electronics from Kyushu Institute of Technology, Iizuka, Japan, in 1996. In 1998, he received the M.E. degree in information science from Nara Institute of Science and Technology, Ikoma, Japan. From April 1998 to September 2000, he was a Ph.D. student at the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology.

Since October 2000, he has been an Assistant Professor in the Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include Internet telephony and analysis of traffic management.

Mr. Iida is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan, and the WIDE project of Japan.

Tetsuya Takine (M'94) was born in Kyoto, Japan, in 1961. He received the B.Eng., M.Eng., and Dr.Eng. degrees in applied mathematics and physics from Kyoto University, Kyoto, in 1984, 1986, and 1989, respectively.

In 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 queueing theory and performance analysis of computer/communication systems.

Dr. Takine is a member of the Operations Research Society of Japan (ORSJ), IEICE, the Information Processing Society of Japan (IPSJ), and the Institute of Systems, Control and Information Engineers. He received the 1996 Best Paper Award from ORSJ.

Hideki Sunahara (M'88) received the B.S. and M.S. degrees in electrical engineering from Keio University in 1983 and 1985, respectively. He received the Ph.D. in computer science from Keio University in 1989.

He is currently an Associate Professor in the Information Technology Center, Nara Institute of Science and Technology, Ikoma, Japan. His research focuses on multimedia communication systems, digital libraries, computer architecture, parallel processing, distributed systems, operating systems, and computer networks.

Dr. Sunahara is a member of the Internet Society, the Japan Society for Software Science and Technology (JSSST), IPSJ, and IEICE. He is also a board member of the WIDE project of Japan.

Yuji Oie (M'83) received the B.E., M.E., and D.E. degrees from Kyoto University, Kyoto, Japan, in 1978, 1980, and 1987, respectively.

From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya, Japan. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo, 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, Ikoma, Japan. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Fàculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high-speed networks, and queueing systems.

Dr. Oie is a member of the IPSJ and IEICE.