PAPER *Special Issue on New Developments on QoS Technologies for Information Networks*

Statistical Analysis of Packet delays in the Internet and Its Application to Playout Control for Streaming Applications

Kouhei Fujimoto[†], *Student Member*, **Shingo Ata**[††], *and* **Masayuki Murata**[†††], *Members*

SUMMARY A packet transmission delay is an important quality characteristic for various applications including real-time and data applications. In particular, it is necessary to investigate not only a whole distribution of the packet transmission delay, but also the tail part of the distribution, in order to detect the packet loss. In this paper, we analyze the characteristics of the tail part of packet delay distributions by statistical analytic approach. Our analytic results show that the Pareto distribution is most appropriate in 95–99.9% region of the cumulative distribution of packet transmission delays. Based on our statistical analysis, we next propose an adaptive playout control algorithm, which is suitable to real-time applications. Numerical examples show that our algorithm provides the stable packet loss ratio independently on traffic fluctuations.

key words: packet transmission delay, one-way delay, distribution function, Pareto distribution, packet loss ratio

1. Introduction

The Internet is now widely deployed and the users can easily get the global accessibility from their home terminals. One of the main reasons for the prevalence of the Internet is in its routing mechanism. Routing of the Internet has two key features; flexibility and scalability. The Internet provides the dynamic routing based on the exchange of the routing information among routers. For example, when a network link becomes down because of some troubles, an alternative route will be prepared automatically. Second, the packet processing at the routers is simple (e.g., FIFO) to reduce the overhead of packet forwarding at the router.

From the users' point of view, on the other hand, the packet transmission delay is an important metric since it directly affects the end-to-end performance. One example can be found in the real-time application using RTP (Realtime Transport Protocol) [1]; a popular protocol for real-time applications in recent years. RTP uses RTCP (Real Time Control Protocol) to control the transmission rate. In RTCP, the sender maintains the transmission delay of packets based on RTT values to control the packet transfer rate. To keep the preferable performance in RTP-based applications, an accurate estimation of the packet transmission delay is essential. However, RTT estimation is insufficient in several situations. In real-time voice communications, for example, it is desirable to separately measure transmission delays of both downstream (sender to receiver) and upstream (receiver to sender) routes because the Internet routes are often asymmetric [2]. From these reasons, it is necessary to investigate not only the characteristic of RTT but also that of one-way transmission delays in order to develop an accurate delay estimation method.

Of course, the dynamic routing of the Internet makes it impossible for the end-users to select the appropriate route for satisfying the users' quality of service (QoS). Furthermore, due to a simple packet processing at routers, it is difficult to predict the transmission delay of the packet. In this paper, we show the accurate packet transmission delay estimation based on the statistical analytic approach.

The studies about the characteristics of the end-to-end packet transmission delay have been made in some literatures [3]–[5], but most of those studies have focused on the average characteristics and the entire distributions only. If we want to detect the packet loss, the tail distribution is more important than the entire distribution. For example, in UDP based real-time applications, control of the *playout* time should be accurate to provide the high-quality real-time service. Here, the playout time is a time when the application client actually begins to play the packet. In the playout control, the client application changes its buffering time, which directly affects the communication quality of the application. While the playout is essential to absorption of the delay variation, too short playout time leads to the fact that the client treats packets to be lost even if those packets eventually arrive. On the contrary, large playout time may introduce an unacceptable delay that the client user cannot be tolerant. A more difficulty exists in determining the playout time. The packet transmission delay between the server and client is changed according to time in the Internet environment. The adequate playout time is heavily dependent on variations of packet transmission delays; i.e., the time-dependent behavior, the delay distribution is also important in determining the playout time.

The issue of playout control has been made by some previous works [6], [7], but most of these algorithms are based on calculation method of the time-out threshold in TCP [8] which tries to manage the packet loss ratio to be closely zero. However, recent streaming applications are robust from some packet losses with keeping the sufficient reproduction quality. If we consider the playout control al-

[†] The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka 560–8531 Japan

[††] The author is with the Faculty of Engineering, Osaka City University, Sumiyoshi, Osaka 558–8585 Japan

[†††] The author is with the Cybermedia Center, Osaka University, Ibaraki, Osaka 567–0047 Japn

gorithm that allows some marginal packet losses, it is possible to shorten the playout delay without degradation of reproduction quality. In such an algorithm, it is necessary to know the characteristics of the packet transmission delay to provide the stable quality and packet loss ratio. From this reason, we first analyze the statistics of packet transmission delays to apply the playout control algorithm.

Keeping those facts in mind, we analyze the characteristics of the packet transmission delays. We first measure the distribution of the one-way transmission delay as well as the round-trip delay, and determine the suitable distribution function through a statistical analytic approach. We next apply the distribution function to estimate the playout time for real-time applications. In an actual situation, some user prefer the real-time reproduction of the media even if the packet loss becomes high, and another user may want high quality at the expense of the large playout time. By taking account of it, we propose a new playout control method which ensures the QoS of real-time application according to user's willingness while minimizing the overhead of playout time.

The paper is organized as follows. We first show a brief summary of the characteristics of the packet transmission delay and our measurement framework in Section 2. In Section 3, we explain our analytic approach to estimate parameters of distribution functions and select the most appropriate distribution. We next show the result of analysis in Section 4. In Section 5, we propose a new playout control method based on the results in Section 4, and show the effectiveness of our proposals. Finally, we summarize our work and describe our future research topic in Section 6.

2. Methods of Packet Transmission Delay Measurements

In this section, we show a brief summary of our measurement method. We measured two types of the packet transmission delay; the round-trip transmission delay and the one-way transmission delay. We first show the outline of the measurement approach, and next describe our measurement environments.

2.1 Measurements of the Round Trip Time

There are several tools to measure RTT values. See [9] and references therein. We adopted `pchar` [10] for RTT measurements. `Pchar` (an updated version of `pathchar` [11]) was developed to measure the bandwidth of intermediate links between two end hosts. `Pchar` uses the ICMP (Internet Control Message Protocol) Time Exceeded message to measure the RTT. More specifically, `pchar` utilizes the TTL (Time To Live) field in the IP packet. By protocol specification, the router decreases the value of TTL by one before the packet forwarding. If the value of TTL becomes zero, the router sends the ICMP 'Time Exceeded' packet back to the sender. Thus, `pchar` intentionally sets the value of TTL to a smaller value to indicate the number of hops the packet can traverse. After the sender receives the ICMP Time Exceeded packet, the sender can obtain the RTT which is the duration between when the host sends the packet and when it receives the ICMP packet. The advantage of using ICMP messages is that it is not necessary to deploy any other hosts to measure the RTT. In addition, `pchar` provides events of routing changes and the packet loss ratio. Those are the reasons why we adopted `pchar`.

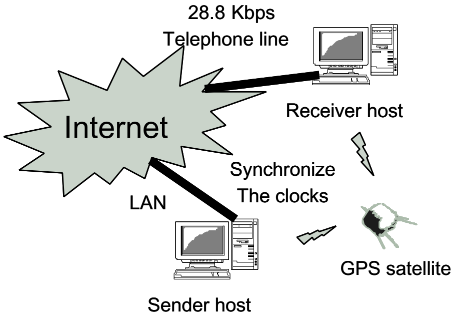


Fig. 1 The Measurement Infrastructure for One-way Delay

2.2 Measurements of the One-way Delay

Figure 1 shows the measurement infrastructure. To measure the one-way delay, we developed the server-client based tool, in which the sender host records the current time into the packet before sending. When the packet arrives at the receiver host, the delay is calculated using the receiver's clock. For this, time clocks of the sender and the receiver should be synchronized. However, synchronization among distributed hosts in the Internet is difficult and a still open issue [12], [13]. To avoid this problem, we use GPS (Global Positioning System) for time synchronization. We measured the one-way delay by considering the real-time applications like a continuous media, in which data packets are periodically sent by the sender host. The Internet radio and the live event concert are categorized into this class.

2.3 Measurement Methodology

In our experimental setting, the measurement host is connected to ISP (Internet Service Provider) via 28.8 Kbps telephone line, since we suppose the case that customers use the streaming based real-time application at their home terminals. We measured RTTs to some famous WWW servers in Japan in January 2000. We next measured one-way delays between two hosts which are connected by 28.8 Kbps modems to different ISPs in July 2000. We also investigate the influences of the following two factors on the determination of suitable distribution functions.

- **Effects of the Time of Day:** It is known that the Internet traffic pattern repeats every day [14]. Thus, it is important to investigate the patterns of the suitable distribution function caused by the effects of "time of day".

- **Effect of the Timescale:** If the timescale for parameter estimation is too short, it may mislead to the wrong estimation. Thus, it is essential to investigate the effect of timescale for the determination of the suitable distribution function.

3. Modeling the Packet Transmission Delay

In this section, we apply the statistical analysis methods to the measurement data following the method described in [15] where the authors analyzed characteristics of `telnet` and `ftp` traffic. In modeling, we emphasis the coincidence at the tail part of delay distribution, because it is useful to detect packet loss in streaming applications. In what follows, we summarize our statistical method.

3.1 Distribution Functions

We selected four distribution functions as candidates to adequately represent delay distributions. The normal and exponential distributions are given by

$$F(x) = \int_0^x \frac{1}{\sqrt{2\pi}\sigma} exp\left[\frac{-(y-\zeta)^2}{2\sigma^2}\right] dy, \tag{1}$$

and

$$F(x) = 1 - \exp(-\frac{x}{\beta}), \quad \beta > 0, \tag{2}$$

respectively. The lognormal distribution is the function, of which variable is the logarithmic variable of the normal distribution, i.e.,

$$F(x) = \int_0^x \frac{1}{\sqrt{2\pi}\sigma y} exp\left[\frac{-(\log y-\zeta)^2}{2\sigma^2}\right] dy. \tag{3}$$

The Pareto distribution is widely known to be able to represent a self-similarity [16], [17], which is given by

$$F(x) = 1 - \left(\frac{k}{x}\right)^{\alpha}, \quad x \geqq k. \tag{4}$$

3.2 Parameter Estimation

In order to detect the packet loss from the distribution of packet transmission delays, the coincidence at the tail part of distributions is more important, even if the measured data are far from the model distribution function in the other part of the entire distribution. To fit the distribution function accurately, we estimate parameters by utilizing only the tail part (e.g., 90–99.9%) of collected delays. For parameter estimation of each distribution function, we use the maximum-likelihood-estimator (MLE) method [18]. Parameters of the exponential and normal distributions can be estimated by calculating the mean and variance of measured delays. In the lognormal distribution, two parameters (ζ, σ) are calculated from

$$\widehat{\zeta} = \frac{1}{n}\sum_{i=1}^{n} \log x_i, \tag{5}$$

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (\log x_i - \bar{x})^2, \tag{6}$$

where n is the number of measurements and $\bar{x}$ is the mean of $log(x_i)$ for all i. Parameters $(\widehat{k}, \widehat{\alpha})$ of the Pareto distribution are obtained from [18];

$$\widehat{k} = \min(x_1, x_2, \cdots, x_n), \tag{7}$$

$$\widehat{\alpha} = n\left[\sum_{i=1}^{n} \log\left(\frac{x_i}{\widehat{k}}\right)\right]^{-1}. \tag{8}$$

3.3 Determination Method of Adequate Distribution

We determine the most appropriate probability distribution function by χ^2-test. Noting that a typical application of our analysis is the playout control for streaming applications, the estimation of the value around the target point (e.g., 99%, 99.9%) of the cumulative delay distribution should be accurate, since it directly affects the packet loss ratio in streaming applications and the reproduction quality of real-time applications. From this reason, we evaluate the coincidence between the candidate functions and measured delays on 95–99.9% region of the cumulative distribution by the χ^2-test.

In the χ^2-test, we investigate the coincidence between delay distribution and candidate functions by $\widehat{\lambda}^2$, which is calculated as follows. We first separate the range of n measured delays into N subranges. For each subrange i, we obtain the probability p_i that an arbitrary value x belongs to the subrange i and the number of measurements Y_i falling into the subrange i. Then, λ^2 is given by

$$\widehat{\lambda}^2 = \frac{X^2 - K - N + 1}{n-1}, \tag{9}$$

where

$$X^2 = \sum_{i=1}^{N} \frac{(Y_i - np_i)^2}{np_i}, \tag{10}$$

$$K = \sum_{i=1}^{N} \frac{Y_i - np_i}{np_i}. \tag{11}$$

The distribution having the smallest value of $\widehat{\lambda}^2$ is most appropriate to represent the measured data. Consequently, we determine the appropriate model distribution.

4. Analytic Results

In this section, we show results of our statistical analysis described in the previous section. Note here that in this paper, we define a “target value” as a probability that the packet can be “successfully” playouted at the destination. We also note that the packet loss within the network is not considered here, and only the packets received within the playout time. Packets are treated as successfully received. For example, if the user wants to play the streaming audio with 1 % of packet loss, the target value should be set to 99 %.

4.1 Essential Results and Effects of Time of Day

We summarize results of χ^2-test in Table 1. The first and second columns of Table 1 show the type of delay (RTT or One-way) and measured time, respectively. Values of $\widehat{\lambda}^2$ for four distributions are shown in columns 3 through 6. The smallest value of $\widehat{\lambda}^2$ among four distributions is shown in bold. As an example, Figure 2 compares the distribution of the measured RTTs with candidate probability functions in busy hours (corresponding to the second row in Table 1). We set the target value to 99% of the cumulative distribution. The distribution labeled by "Sample" is the tail part (90–99.9%) of the cumulative density distribution of measured RTTs. The cumulative distribution of RTT values during non-busy hours is shown in Figure 3, which corresponds to the eighth row of the table. It also shows the tail part of the measured RTTs' distribution and candidate probability functions.

We can observe from Table 1 that $\widehat{\lambda}^2$ of the Pareto distribution is always smallest in all experiments, i.e., the Pareto distribution is most suitable to estimate the 99% value of cumulative distribution in busy hours (e.g., 11 PM[†]) and standard hours (e.g., 2 PM). It is applicable to both RTTs and one-way delays.

To illustrate the importance of examining the tail part of the distribution, we next present the case where the χ^2-test is applied to the entire cumulative distribution. Table 2 shows the result. Comparing with Table 1, the model determination method picked up different distributions (normal or lognormal distribution), which were not observed when examining only the tail part of distributions. Note that, when the network is busy, the Pareto distribution which has heavy-tail becomes most suitable. It coincides the past researches, which showed that the distribution of packet delays is heavy-tailed as the network becomes congested [15]. In Figure 4, we can observe the significant increase of delay at 11 PM. From 11 PM to around 1 AM, the delay becomes heavy-tailed, because the ratio of long delay is larger then the other period.

It is also worth noting that as we will describe in the next section, we will apply the statistical results presented in this section to on-line estimation of the delay, which is necessary in adaptive playout control. Thus, we want a light-weight estimation method for the delay distribution. Since we found that the Pareto distribution is most appropriate regardless of the "time of day", it is not necessary to examine the χ^2-test for each measurement, and we only have to determine the parameters of the Pareto distribution. If the appropriate model is varied according to the "time of day", we need to examine the χ^2-test for each playout controls. However, the computational overhead of χ^2-test is not small, and it is inadequate for real-time applications.

Table 1 Results on Model Determination (Tail-Part of Delay Distributions) Nor. : Normal, Exp. : Exponential, Lognor. : Lognormal

Measurement		Result of χ^2-test			
Delay Type	Hour	Nor.	Exp.	Lognor.	Pareto
RTT	10 PM	332.17	2371.91	266.60	**79.75**
RTT	11 PM	122.22	471.56	103.56	**74.32**
RTT	0 AM	156.09	670.34	128.86	**58.45**
RTT	1 AM	157.21	2189.33	139.47	**49.81**
RTT	2 AM	362.24	1691.48	242.74	**115.28**
RTT	7 AM	292.30	3598.50	240.55	**124.03**
RTT	10 AM	169.64	970.60	360.29	**80.57**
RTT	2 PM	147.02	599.37	250.51	**56.25**
RTT	7 PM	194.33	584.95	257.05	**55.63**
One-way	9 PM	83.82	602.56	71.96	**19.56**
One-way	11 PM	53.86	470.90	49.67	**30.10**
One-way	1 AM	55.06	426.46	49.99	**24.01**
One-way	5 AM	94.45	500.91	85.77	**25.16**
One-way	9 AM	107.76	754.09	98.74	**45.33**
One-way	12 PM	108.66	1218.95	101.09	**30.61**
One-way	3 PM	109.07	336.49	85.41	**21.21**

Table 2 Results on Model Determination (Entire Delay Distributions) Nor. : Normal, Exp. : Exponential, Lognor. : Lognormal

Measurement		Result of χ^2-test			
Delay Type	Hour	Nor.	Exp.	Lognor.	Pareto
RTT	11 PM	173.59	830.91	126.45	**100.22**
RTT	1 AM	164.39	1136.62	**130.49**	130.64
RTT	7 AM	154.59	1780.39	**97.49**	189.54
RTT	10 AM	**21.09**	49.27	32.16	36.873
RTT	2 PM	**22.07**	46.27	26.75	34.51
One-way	2 AM	4.71	25.43	**2.33**	3.51
One-way	4 AM	13.11	84.54	**12.85**	20.75
One-way	10 PM	20.93	268.79	**20.19**	281.81
One-way	5 PM	14.53	149.62	**13.11**	26.17
One-way	8 PM	4.18	5.55	**5.22**	16.96
One-way	11 PM	13.66	33.03	5.31	**3.92**

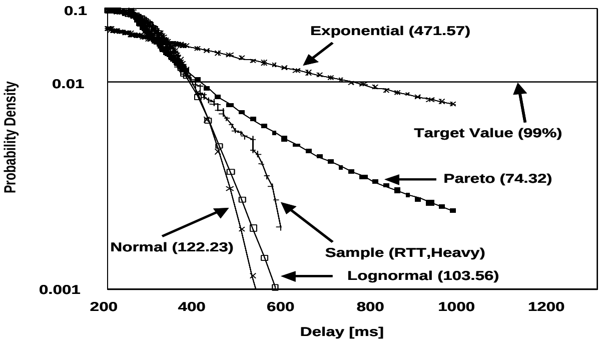


Fig. 2 Comparisons among Sample and Candidate Functions (RTT, 11 PM)

4.2 Effects of Timescale

We next examine the effects of the timescale by changing the number of samples for the parameter estimation. Figures 5(a) and 5(b) show the degree of differences against the number of measured data for RTT and one-way delays, respectively. We calculate the difference between 99% values of the Pareto distribution and those of the cumulative

[†] It is because NTT (one of largest carriers in Japan) offers the service with unlimited accesses at a fixed charge from 11 PM to 8 AM.

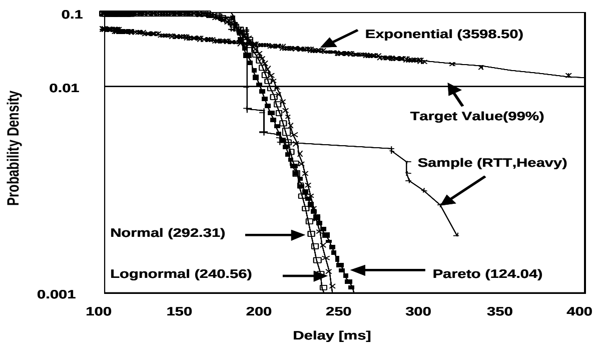


Fig. 3 Comparisons among Sample and Candidate Functions (RTT, 2 PM)

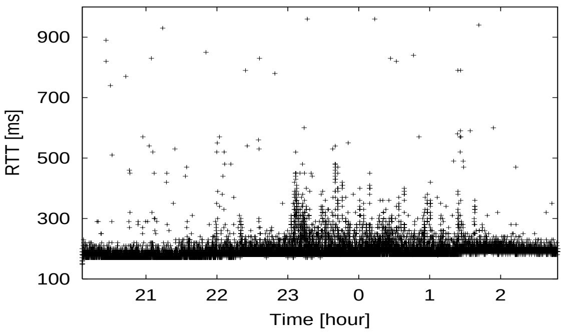


Fig. 4 Time Dependent Fluctuations of RTTs

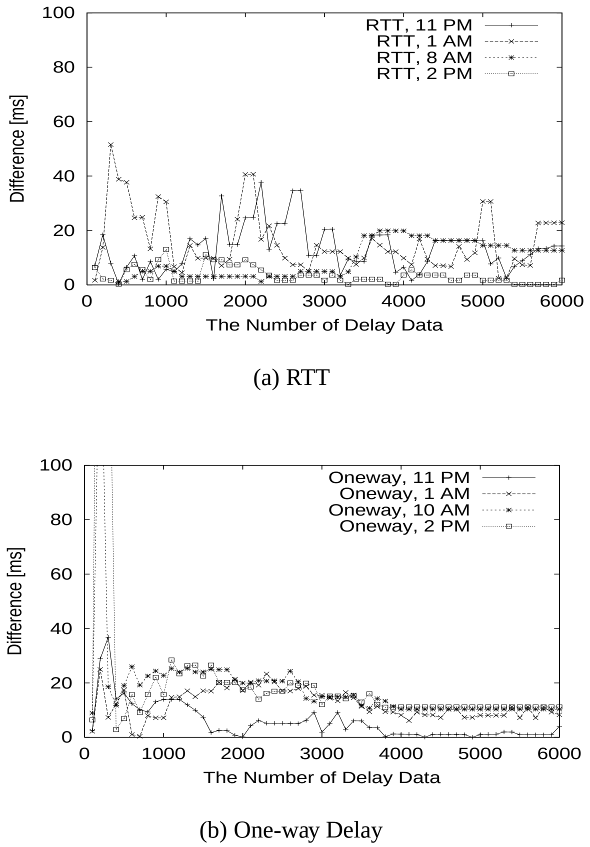
distribution of collected samples. As shown in the figure, the difference gets remarkable if the number of samples are less than 500. On the other hand, we cannot observe critical changes when the number of measurement data are equal to or more than 500. Since our objective is to perform on-line estimation of the delay distribution, it is preferable that the number of sample is as small as possible, then the parameter can be estimated faster with the less number of samples. From the results, we can conclude that the required number of measurements should be equal to or more than 500, in which $500 \times (99.9\% - 90\%) \cong 50$ samples are at least necessary for the accurate parameter estimation of the Pareto distribution. We will evaluate the required number of samples for quick and still accurate parameter estimation in the next section.

5. Playout Time Estimation Method Based on Statistical Analysis

In this section, we propose a new playout control algorithm in which playout time is determined based on our statistical analysis. Then, we evaluate our playout control algorithm by the trace-driven simulation, and we investigate an effectiveness of the proposed algorithm.

5.1 Proposed Algorithm

To provide a high-quality communication in streaming applications, the packet loss ratio should be kept small. Because packets arriving after the playout time is not meaningful, playout time should be chosen carefully. In addition, the playout control should provide some means to determine the quality level of “real-time” transmission of the media that the user is acceptable. The main goal of our algorithm is to minimize the playout time while keeping the reproduction quality specified by user’s requirements. We use the results obtained through the statistical analysis presented in the previous section to determine the proper playout time.



(a) RTT

(b) One-way Delay

Fig. 5 The Difference between Values of the Pareto Distribution and Measured Data at the Target Value (99%).

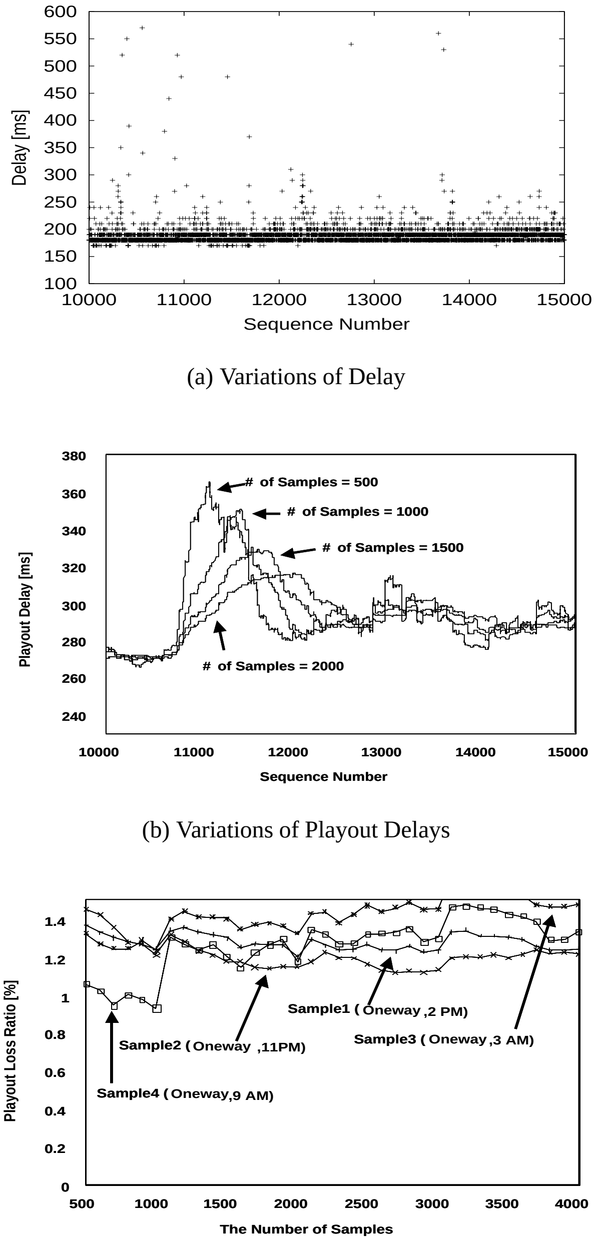
More specifically, our playout algorithm records the history of one-way delays of packets. On each packet arrival, parameters of the Pareto cumulative density function $F(x)$ is updated to estimate the playout delay d_i from the equation $F(d_i) = X$ where X is the target value. Here, the playout delay is a period from the time when the sender host sends a packet to the time when the receiver host starts to playout the received packet. Additionally, the playout time p_i is the expiration time that the receiver regards the packet i as the valid data for streaming application, which is after d_i seconds from the sender transfers the packet i. Furthermore, we consider 95, 99, and 99.9% as the target value X through our numerical results. For example, if we choose $X = 95\%$, our algorithm tries to minimize the playout time while keeping the packet loss to be 5%. Of course, if the packet loss within the network exceeds 5%, our method has no means to keep the packet loss to be 5%. In what follows, we will assume that the packet loss within the network does not exceed the target value.

In the following subsections, we will provide the trace-driven simulation results. A set of one-way delays of packets collected from the operating Internet is used in our simulation. The packet size was set to be 160 Bytes, and an interval of packet emissions is fixed at 80 msec. Then, we estimate the playout time p_i of the packet i according to the algorithm. In our simulation, we check whether the next packet arrives within the estimated playout time or not, and if the packet does not arrive, it is treated as packet loss. We note here that in an actual implementation, some control is necessary to return the collected information from the receiver to the sender, but we do not consider it here. Such information may be lost in the actual Internet, but it is also not considered in the simulation.

5.2 Parameter Setting

In our algorithm, the number of measurements for the parameter estimation becomes a dominant factor. The accuracy of parameter estimation can be improved by increasing the number of measurement data. However, the larger number of samples inhibits to follow the dynamic changes of the network condition, and the playout control cannot follow a drastic variation of one-way delays. We then investigate how many number of samples is adequate for playout control. Note that the minimum number of samples should be 500 to estimate accurate parameters of the Pareto distribution as described in the Subsection 4.2.

We now demonstrate the influence of the number of samples on the playout control by changing the number of delays for the parameter estimation. The results of experiments are shown in Figure 6. Figure 6(b) shows the four variations of playout delay evaluated by the simulation with the traced data shown in Figure 6(a); we change the number of samples to be 500, 1,000, 1,500 or 2,000. From Figure 6(b), it is clear that the smaller the number of samples is, the more quickly the playout delay follows the delay variation. This result shows the smaller number of samples is better to follow the changes of delay. Next, in Figure 6(c), the packet loss ratio is plotted against the number of samples. In this figure, we plot four cases of the busiest (11 PM) and non-busy hours (2 PM, 3 AM, 9 AM), and we set the target packet loss ratio to be 1%. In Figure 6(c), we cannot observe a significant improvements at any case even if the number of samples is large. From this result, we consider there is a trade-off relationship between the improvement of accuracy with the increased number of samples in parameter estimation and the degradation of adaptability to delay variations. However, the less number of samples in parameter estimation means the less computing, which would be desirable for the real-time applications. Based on above results, the number of samples can be set to be 500 for parameter estimation of playout controls in our algorithm.



(a) Variations of Delay

(b) Variations of Playout Delays

(c) Playout Loss Ratio dependent on the Number of Measured Samples

Fig. 6 Effects of the Number of Samples in Playout Control

5.3 Performance Comparisons

For comparison purpose, we also examined three algorithms which have been proposed in [6], [8]. Note here that we refer to our proposed algorithm as **Algorithm 1** throughout this section.

In **Algorithm 2**, the playout time is determined from the approximated values of the mean $\widehat{d}_i$ and variance $\widehat{v}_i$ of

Table 3 Comparison of PLR and Mean Playout Delay

Hour	Algorithm	Target	PLR [%]	Mean of d_i [ms]
11 PM	1	95%	5.13	221.17
		99%	1.37	265.12
		99.9%	0.14	855.38
	2	-	2.54	237.36
	3	-	0.08	734.90
	4	-	3.95	230.72
3 AM	1	95%	5.42	247.66
		99%	1.08	270.11
		99.9%	0.10	386.23
	2	-	1.18	273.96
	3	-	0.05	555.51
	4	-	2.96	258.53
2 PM	1	95%	5.32	242.67
		99%	1.34	267.29
		99.9%	0.10	414.48
	2	-	1.49	259.19
	3	-	0.04	466.57
	4	-	3.08	252.50

one-way delays, which are given by

$$p_i = t_i + \widehat{d}_i + 4\widehat{v}_i, \tag{12}$$

$$\widehat{d}_i = \alpha\widehat{d}_i + (1-\alpha)n_i, \tag{13}$$

$$\widehat{v}_i = \alpha\widehat{v}_i + (1-\alpha)|\widehat{d}_i - n_i|. \tag{14}$$

That is, the playout time is decided without a knowledge on the delay distribution. **Algorithm 3** is a modified version of **Algorithm 2**, which uses the weighted mean of $\widehat{d'_i}$s as

$$\widehat{d'_i} = \begin{cases} \beta\widehat{d'}_{i-1} + (1-\beta)n_i & \text{if } n_i > \widehat{d'}_{i-1}, \\ \alpha\widehat{d'}_{i-1} + (1-\alpha)n_i & \text{otherwise,} \end{cases} \tag{15}$$

where α and β are constant values, satisfying $0 < \beta < \alpha < 1$. We set $\alpha = 0.998500$ and $\beta = 0.750000$ by following [6] . **Algorithm 4** focuses on *spike* which represents a sudden and large increase in delays on a sequence number of packets. Examples of spikes are shown at 16,130, 16,170 and 16,200 in Figure 7(a). **Algorithm 4** usually obtains the playout time from Eq. (13), which is same as **Algorithm 2**. During spike, on the other hand, **Algorithm 4** uses the following equation;

$$\widehat{d}_i = \widehat{d}_{i-1} + n_i - n_{i-1}, \tag{16}$$

to catch up the sudden increase of delays. In **Algorithm 4**, we use $\alpha = 0.875$.

Table 3 compares packet loss ratios (PLRs) and mean values of the playout delay in three periods. Each period is about an hour. In **Algorithm 1**, we used 95, 99, and 99.9% as the target values. We can observe that there is a clear trade-off between PLR values and the playout delay. Note again that the purpose of our proposed **Algorithm 1** is that the value of PLR can be kept close to the packet loss ratio requested by users. These results show that PLR in **Algorithm 1** is almost satisfied with the intended packet loss ratio $(1 - X)$. Although the target PLRs of **Algorithms 2**, **3** and **4** are 0%, the results show that playout time is extremely enlarged in spite that longer playout time makes it difficult to playback in real-time. Of course, the PLRs of **Algorithms 2** through **4** might be controlled by changing the multiplier of $\widehat{v}_i$, which is currently set to be 4 (see Eq. (13)). However, the fundamental problem is that there is no means to map the multiplier to the value of PLR in those algorithms. On the other hand, we can observe that our proposed algorithm can control the playout time so that PLR is kept close to the target packet loss ratio with accuracy. Namely, it is possible for our proposed playout control to keep the target packet loss ratio in any hour.

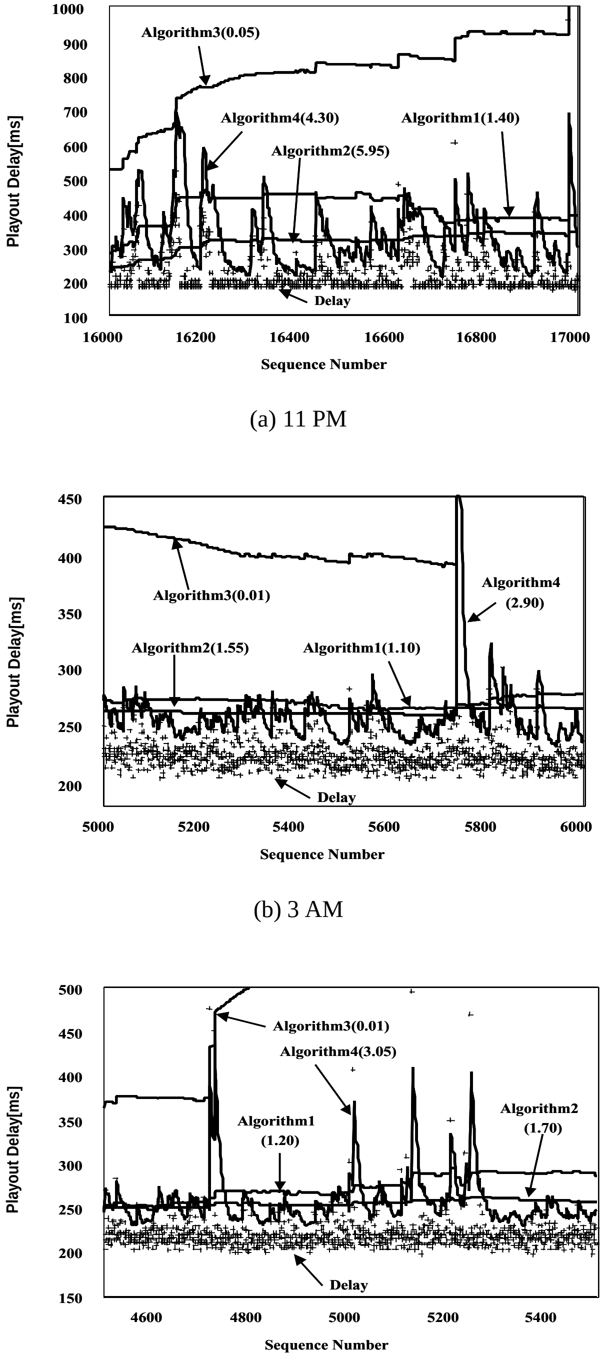
Figure 7 compares the playout delay variation among four algorithms in each period of Table 3. In **Algorithm 1**, we set the target value X to be 99%, and the PLR of each algorithm is also shown by the number in parentheses. Note here that PLRs in Figures 7 and Table 3 are different; e.g., 2.54% at (11PM, **Algorithm 2**) in Table 3, and 5.95% in Figure 7(a). It is because different numbers of samples are used to calculate the mean PLR. From these figures, we can find that **Algorithm 3** has a tendency to overestimate the playout delay, which is twice as large as that of **Algorithm 1**. Especially, in Figure 7(a), the playout delay increases up to 1 sec during busy hours. On the other hand, **Algorithm 2** always computes the smallest playout delay, which leads to many packet losses. Additionally, Figure 7(a) shows that **Algorithm 2** cannot follow the variation of packet delays adaptively, and therefore, **Algorithm 2** is not suitable during busy hours as expected. Figure 7(a) also shows that **Algorithm 4** can follow the drastic change of delays because of the spike detection. As shown in Figures 7(b) and 7(c), however, **Algorithm 4** is too sensitive from the spike in which the number of packets is quite small. It sometimes misleads to unnecessary increase of playout delay.

In summary, our simulation results show that **Algorithms 2**, **3**, and **4** cannot keep the small PLR by the heavy fluctuation of packet transmission delays. On the other hand, we can conclude that our algorithm is superior to others on which **Algorithm 1** provides a stable PLR specified by the user regardless of arbitrary delay changes.

6. Concluding Remarks

In this paper, we have measured packet transmission delays and analyzed their characteristics by taking into account the time of day. From statistically analytic results, we have found that the Pareto distribution is most appropriate as the model of one-way delay distribution, as well as RTT distributions.

Moreover, we have proposed a playout control algorithm based on our analysis. Numerical examples have shown that our proposed method can control the playout time with satisfying the target packet loss probability. For future research topics, it is necessary to consider the update process of the playout time in order to apply our algorithm to streaming applications.



(a) 11 PM

(b) 3 AM

(c) 2 PM

Fig. 7 Playout Delay Variations of Algorithms 1 through 4

Acknowledgements

This work was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems" (JSPS-RFTF97R16301).

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *RFC 1889*, January 1996.

[2] S. Savage, A. Collins, and E. Hoffman, "The end-to-end effects of Internet path selection," in *Proceedings of ACM SIGCOMM '99*, pp. 289–299, September 1999.

[3] D. Cohen, "Issues in transnet packetized voice communications," pp. 6–10–6–13, September 1977.

[4] J.-C. Bolot, "Characterizing end–to–end packet delay and loss in the Internet," in *Proceedings of ACM SIGCOMM '97*, pp. 289–298, September 1997.

[5] V. Paxson, "End–to–end Internet packet dynamics," in *Proceedings of ACM SIGCOMM '97*, pp. 139–152, September 1997.

[6] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide–area networks," in *Proceedings of IEEE INFOCOM '94*, pp. 680–688, April 1994.

[7] B. J. Dempsey and Y. Zhang, "Destination buffering for low-bandwidth audio transmissions using redundancy-based error control,," in *Proceedings of LCN, 21st Annual Conference on Local Computer Networks*, pp. 345–354, October 1996.

[8] J. Postel, "Transmission cotrol protocol specification," *RFC 793*, September 1981.

[9] "Caida measurement tool taxonomy," `http://www.caida.org/tools/taxonomy/index.xml`.

[10] B. A. Mah, "pchar: A tool for measuring Internet path characteristics," `http://www.ca.sandia.gov/~bmah/Software/pchar/`.

[11] A. B. Downey, "Using pathchar to estimate Internet link characteristics," in *Proceedings of ACM SIGCOMM '99*, pp. 241–250, August 1999.

[12] V. Paxson, "Measurements and analysis of end-to-end Internet dynamics," *Ph.D.Thesis, University of California Berkeley*, April 1997.

[13] S. Kalidindi and M. J. Zekauskas, "Surveyor: An infrastructure for Internet performance measurements," *in Proceedings of INET '99 available at* `http://www.advanced.org/csg-ippm/`, June 1999.

[14] K. Thompson, G. J. Miller, and R. Wilder, "Wide–area Internet traffic patterns and characteristics," *IEEE Network*, pp. 10–23, November 1997.

[15] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," in *Proceedings of ACM SIGCOMM '94*, pp. 257–268, August 1994.

[16] W. E. Leland, M. S. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," in *Proceedings of ACM SIGCOMM '93*, vol. 2, pp. 183–193, February 1993.

[17] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web; traffic evidence and possible causes," in *Proceedings of ACM SIGMETRICS '96*, pp. 160–169, May 1996.

[18] V. Brazauskas and R. Serfling, "Robust and efficient estimation of the tail index of a one-parameter pareto distribution," *North American Actuarial Journal available at* `http://www.utdallas.edu/~serfling/`, April 2000.

Kouhei Fujimoto received the B.E. degree Informatics and Mathematical Science from Osaka University, in 2000. He is a master course student of graduate school of Engineering Science, Osaka University. His research area is in the area of modeling and statistical evaluation of network characteristics.

Shingo Ata received the Ph.D. degree Informatics Science from Osaka University, in 2000. He is currently a Research Associate in the Faculty of Engineering, Osaka City University. His research work is in the area of performance modeling of high-speed communication networks. He is a member of the IEEE and ACM

Masayuki Murata received the D.E. degree in Information and Computer Sciences from Osaka University, in 1988. In 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From 1987 to 1989, he was an Assistant Professor with Computation Center, Osaka University. On 1989, he moved to the Department of Information and Computer Science, Faculty of Engineering Science, Osaka University From 1992 to 1999, he was an Associate Professor in the Graduate School of Engineering Science, Osaka University. He became a professor in 1999. He moved to the Cybermedia Center, Osaka University in 2000. His research interests include computer communication networks, performance modeling and queueing systems. He is a member of the IEEE and ACM.