Master's Thesis

Title

A rate control mechanism with attractor perturbation model for stabilizing end-to-end delays

Supervisor

Professor Masayuki Murata

Author

Midori Waki

February 12th, 2013

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

Master's Thesis

A rate control mechanism with attractor perturbation model for stabilizing end-to-end delays

Midori Waki

Abstract

The information network as a multiple access medium has unavoidable issues that end-to-end delay and delay jitter always fluctuate. The main reason is that the internal condition of the information network dynamically changes due to competitive sessions which share network resources and change their throughput for their own convenience. Whereas the delay fluctuation is unavoidable in information networks, in these days, even delay sensitive applications, which traditionally preferred a circuit-switched network, such as voice and video communication, come to rely on packet-switching information networks. For smooth and interactive communication by using such applications over the information networks, the QoS (Quality of Service) guarantee of delay is required. For such delay sensitive applications, buffering at a host and packet scheduling at routers would solve the delay fluctuation problem to some extent. However, these mechanisms to remove delay fluctuations requires all intermediate routers from a server to a receiver to be equipped with the designated algorithm, which is impractical in a large-scaling information network. Therefore, to realize the stable end-to-end delay, we need a novel mechanism for end nodes to leverage delay fluctuations. In biology, it is well known that biological systems are exposed to internal and external fluctuations. It is expected that biological systems inherently have respectable mechanisms against fluctuations, but we could not understand these mechanisms very well. In the recent interdisciplinary activities in biology, a relationship between fluctuation inherent in biological systems and their response against an external force is modeled as the attractor perturbation model. The information network is also a large complex system exposed to fluctuations, and it is difficult to comprehend the whole system and its dynamics. As the information network has a lot of similarities to the biological system, we focus on utilizing the attractor perturbation model for network control. Our preliminary experiments show that the end-to-end delay over a TCP session similarly fluctuates with the biological system in the attractor perturbation model. This implies that we can estimate the appropriate amount of increase or decrease of the sending rate to achieve the desired end-to-end delay based on the attractor perturbation model. In this paper, we propose a novel rate control mechanism that can achieve and maintain the desired end-to-end delay in the changing environment. Our proposal does not filter or conceal fluctuation, but it exploits inherent traffic fluctuation to accomplish the goal according to the attractor perturbation model. Based on the attractor perturbation model, even if the traffic characteristics of a network dynamically changes, our proposal can estimate the appropriate amount of increase or decrease of the sending rate to achieve the desired end-to-end delay only by using instantaneous observation of the end-to-end delays and its variance. Through simulation experiments, we confirmed that our proposal could achieve and maintain the target delay even when background traffic changed. Furthermore, it was found that our proposal could achieve the target delay at higher sending rate than the delay-based AIMD rate control mechanism by nearly 20 % under the same conditions.

Keywords

Attractor perturbation model
Rate control mechanism
End-to-end delay
Delay fluctuation

Contents

List of Figures

List of Tables

1 Introduction

Information network is one of indispensable social infrastructures and it enriches and ensures our daily life. In these days, a wide variety and number of applications, including even voice and video communication which prefer a circuit-switched network, rely on packet-switching information networks. However, as an information network is a shared medium, there occur non-negligible issues. Especially when there are multiple sessions sharing the same physical network resources, delay, delay jitter, and packet loss observed by a session always fluctuate, regardless of adopted protocol or characteristics of generated traffic. Since the origin of fluctuation includes changes in the number of sessions and the amount of traffic, the shadowing and fading of a wireless channel, rerouting of paths and others, that cannot be predicted or controlled by an individual session, researchers has made efforts to suppress the delay fluctuations for delay sensitive applications, researchers had made an effort to suppress fluctuations especially for delay-sensitive applications such as IPTV (Internet Protocol TeleVision) and video conference.

Delay fluctuation is generally managed by a playout buffer at a receiver [1, 2]. A playout buffer defers video playout to deposit the sufficient number of packets at the beginning and then provides a video player with buffered packets. As such, as far as packets arrive at a receiver before a buffer becomes empty, a video can be presented to a user without interruptions. However, delay and delay jitter are not predictable. Therefore, it is very likely that a buffer runs out of packets and a user experiences freezes. Increasing the number of packets to buffer merely degrades the interactivity and timeliness of an application.

For delay-sensitive applications, researchers proposed methods to control and reduce delay jitter by developing an intelligent packet scheduling algorithm at routers [3–7] and by multipath routing [8]. In [3], comparative analysis shows that packet scheduling at routers can reduce delay jitter even when buffering at a receiver cannot prevent freezes. However, it requires all intermediate routers from a server to a receiver to be equipped with the designated algorithm. On the contrary, a multipath routing method relies on prior knowledge of the average delays in the steady state, but its precise prediction is difficult due to the fluctuation in networks. As a mechanism adopted at end systems, many rate control algorithms have been studied [9, 10]. They infer the network state by observing, for example, delay, delay jitter, and packet loss and regulate the sending rate to avoid network congestion. Although they can reduce the packet loss probability, they do not take into account the delay sensitivity of interactive applications.

As long as the network condition, such as the degree of congestion, can easily be predicted or estimated, it is trivial to control delay, delay jitter, and packet loss. However, the ever-increasing size, complexity, and dynamics of an information network prevent a control mechanism revealing the network condition even with active and aggressive probing. Going back to the simplest paradigm, given a complex system, what an end system can do is only to apply a force and see how it reacts. Only if there exists the clear relationship between them, one can obtain the desired result by putting the appropriate force to a system. An answer can be found in biology, which has the long history of investigating and understanding complex systems, i.e. living organisms. The relationship is modeled by a mathematical expression, called an *attractor perturbation model* [11,12]. It is derived from the relationship between fluctuations inherent in a biological system and its response against an external force. Biological systems are always exposed to internal and external fluctuation or noise caused by, for example, thermal fluctuation and phenotypic fluctuation. As a result, size, metabolic concentrations, and gene expression differ among cells cultured in the same medium and individuals are all different. Furthermore, gene expression of a cell dynamically changes to adapt to the surrounding conditions such as temperature, pH, and concentrations of chemical substances. Therefore, a cell is not always the same. It is considered that such fluctuation or diversity is a source of flexibility and adaptability of biological systems to environmental changes.

In the work [11], the authors call the distribution observed quite often in biological experiments "Gaussian-like" distribution. Under the assumption that a variable in biological systems follows the Gaussian-like distribution, the authors derive a mathematical expression which is the base of the attractor perturbation model. The attractor perturbation model explains how biological systems respond to environmental changes, which put force or pressure on them and trigger their responses. Based on the model, given a change in the external force, the average of a measurable variable, such as the concentration of metabolic substances and the number of cells, shifts by the amount in proportional to the degree that a biological system fluctuates, i.e. the variance of the measurable variable. That is, more a biological system fluctuates, more it responds to the environmental change and alters its behavior.

Fluctuation is intrinsic to an information network as well. However the distribution would differ depending on the condition of measurement, including network topology and loading status.

For example in an early research [13], they authors found that the distribution of one-way delay changed from gamma, Pareto, to log-normal distribution as the sending rate increased. We also conducted experimental measurement in real information networks as will be reported in this paper. Analyzing obtained results, we found that round trip time (RTT) experienced by a TCP session followed the Gaussian distribution. It means that the attractor perturbation model could hold in an information network and there is high probability that we can successfully develop a novel control mechanism based on the model.

When we regard a network as a biological system and injected traffic as an external force imposed on a system, we can estimate how a network responds to a change in the injected traffic. More specifically, by adopting the end-to-end delay as a measurable variable of the attractor perturbation model, we can derive the appropriate amount of increase or decrease of the sending rate to achieve the desired end-to-end delay from the observed variance of delay. For example, assume that the measured end-to-end delay is larger than the desired delay. When the variance is large, it is enough to slightly decrease the sending rate to push down the delay to the desired level. On the contrary, aggressive rate control is required in a network with small fluctuation, which implies that a network is stable. With such a control mechanism based on the attractor perturbation model, efficient and effective rate adaptation can be accomplished without detailed information about a network system or tailored facilities.

The remainder of this paper is organized as follows. In Section 2, we measure the delays in the real information network and confirm the delay distributions. In Section 3, we briefly introduce the attractor perturbation model and its applications. In Section 4, we verify the attractor perturbation model in an information network. In Section 5, we propose a novel rate control mechanism to achieve the stable end-to-end delay based on the attractor perturbation model. Then we describe the scenarios of simulation experiments to evaluate the proposal in Section 6 and show the results of the simulation experiments in Section 7. Finally, Section 8 concludes the paper.

2 Fluctuations in information networks

In order to understand how the end-to-end delay fluctuates in real information networks, we conduct experiments to measure the end-to-end delays and investigate its distribution. In Sec. 2.1, we describe the settings of the and show obtained results in Sec. 2.2.

2.1 Experimental settings of delay measurements

As the end-to-end delay we measure the round trip time (RTT) experienced by TCP (Transmission Control Protocol) session. We consider the server and client model, which is common in the Internet. Figure 1 shows the experimental environment. We use an Android tablet as a client host which is equipped with a network interface capable of accessing three wireless networks, i.e., LTE (Long Term Evolution), W-CDMA (Wideband Code Division Multiple Access), and Wi-Fi (IEEE 802.11g). Each access network uses different network technologies and has different characteristics in terms of bandwidth, communication range, delay, and their fluctuation.

We use iperf [14] to establish a TCP session between a server and a client. An iperf server runs on a server and communicates with an iperf client running on a client over a TCP session. Both of the server and the client assign 256 Kbytes to both of a TCP send buffer and a TCP receive buffer. A file of 100 Mbytes is transmitted from the server to the client by a series of segments, whose maximum size is set to 1460 bytes. On the TCP session, all of timestamp option, selective acknowledgement (SACK), and Nagle algorithm are disabled, while the delayed acknowledgement (delayed ACK) is enabled as summarized in Table 1. TCP segments exchanged on a TCP session are captured by tcpdump on the server. The distance between the server and the client during experiments is measured by traceroute and it is 13 hops when using LTE and W-CDMA and 17 hops over a Wi-Fi access network, respectively.

Now we explain how to derive RTT of the communication between a client and a server. Here we represent the segment whose sequence number is j by the j-th segment, and the corresponding ACK whose acknowledgement number is j by the j-th ACK, respectively. Basically, RTT can be derived by subtracting the time of emission of the j-th segment from the time of reception of the j-th ACK. However we should note that the delayed ACK mechanism disturbs accurate delay measurement using such algorithm. With the delayed ACK enabled, a client stops sending an ACK per received segment. Instead, when a client receives the $(j-1)$-th segment, it defers ACK

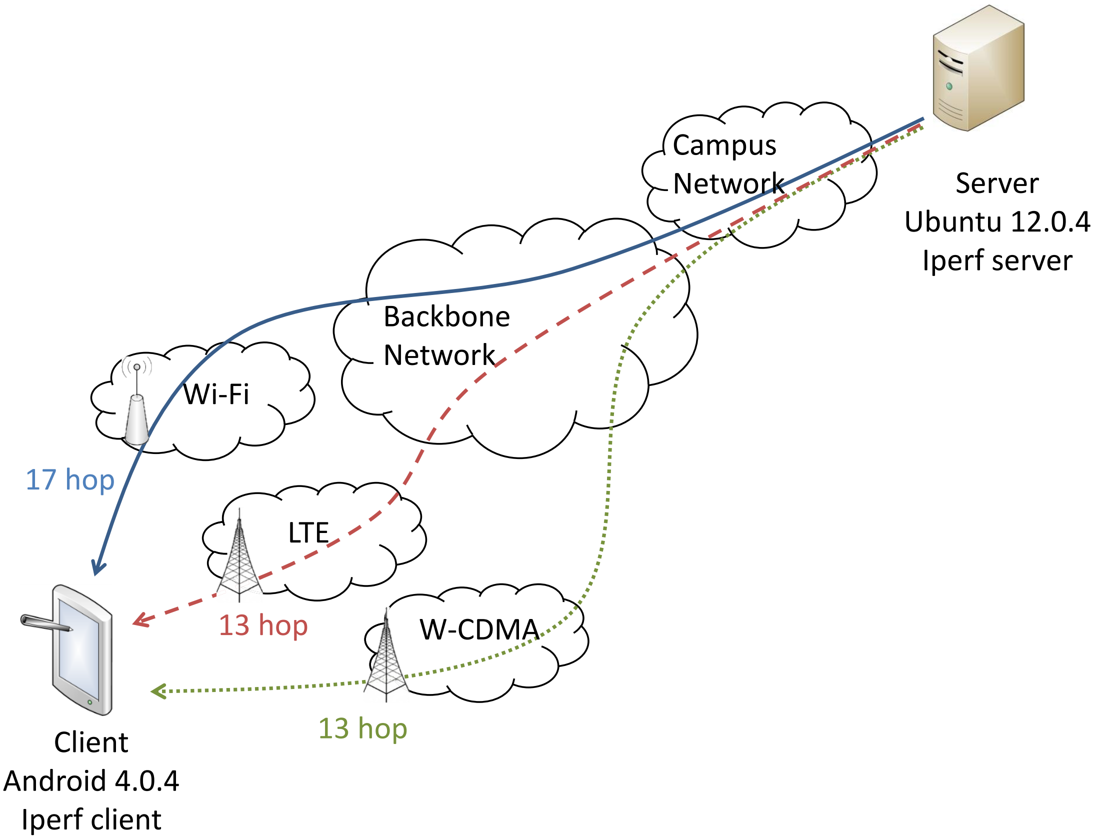


Figure 1: Experimental environment

Table 1: Experimental settings

Entry	value
Size of transmitted file	100 [Mbyte]
Maximum segment size	1460 [byte]
Send buffer size	256 [Kbyte]
Receive buffer size	256 [Kbyte]
Time stamp option	disabled
Selective acknowledgement	disabled
Nagle algorithm	disabled
Delayed acknowledgement	enabled

emission until reception of the next segment, the j-th segment. That is, only the j-th ACK is sent in order to report reception of two successive segments, the $(j-1)$-th and the j-th segments. The problems is that, if the j-th segments does not arrive in the heavily loaded condition, a deadlock could happen due to waiting for arrival of the j-th segment. To avoid this, a client adopts a soft state mechanism, that is, a client sends the $(j-1)$-th ACK when the timer expires without receiving the j-th segment. In that case, the duration from emission of the $(j-1)$-th segment to reception of the $(j-1)$-th ACK includes a timeout period. Therefore, we need to distinguish a normal delayed ACK from an ACK experiencing timeout. Now we consider that a server receives the j-th ACK. If a server received the $(j-1)$-th ACK as well, we assume that the j-th ACK is emitted due to the timer expiration. Otherwise, we assume the j-th ACK is emitted on receiving the two successive segments, i.e. the $(j-1)$-th and the j-th segments. In our experiment, we take the RTTs only in the latter case into account for accurate measurement.

2.2 Delay distributions

We show histograms of measured RTT for each of access networks in Figs. 2, 3 and 4, respectively. The class interval are set at 20 ms in Figs. 2 and 3 and 2 ms in Fig. 4, respectively. Experiments were conducted at different time as shown in captions. We approximate distributions by a Gaussian function $f(x)$ by the least square approximation, where $f(x)$ is given as follows.

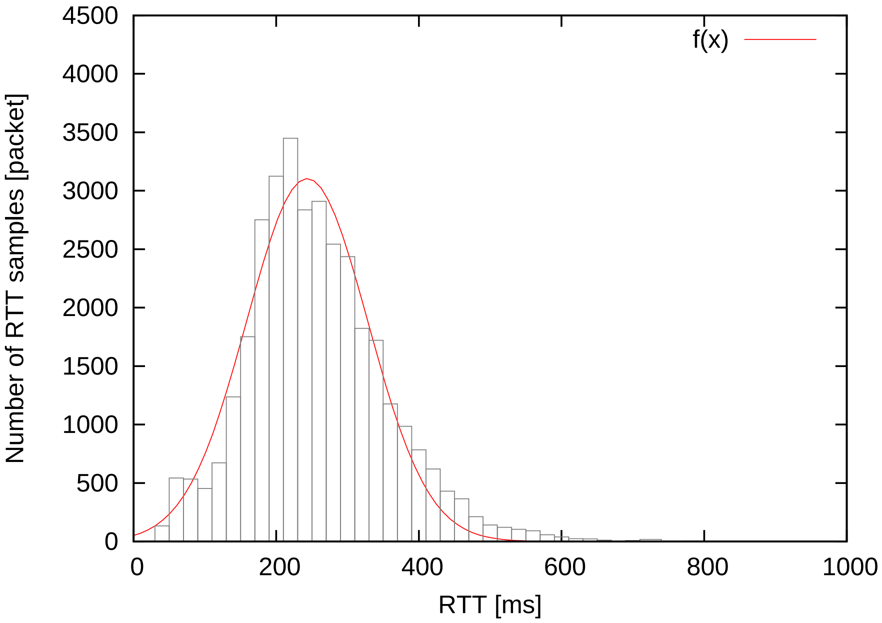
$$f(x) = a \exp(-\frac{(x-b)^2}{2c^2}), \tag{1}$$

where a, b, and c are fitting parameters. The approximated $f(x)$ are shown in a red curve in each figure. Table 2 summarizes parameters.

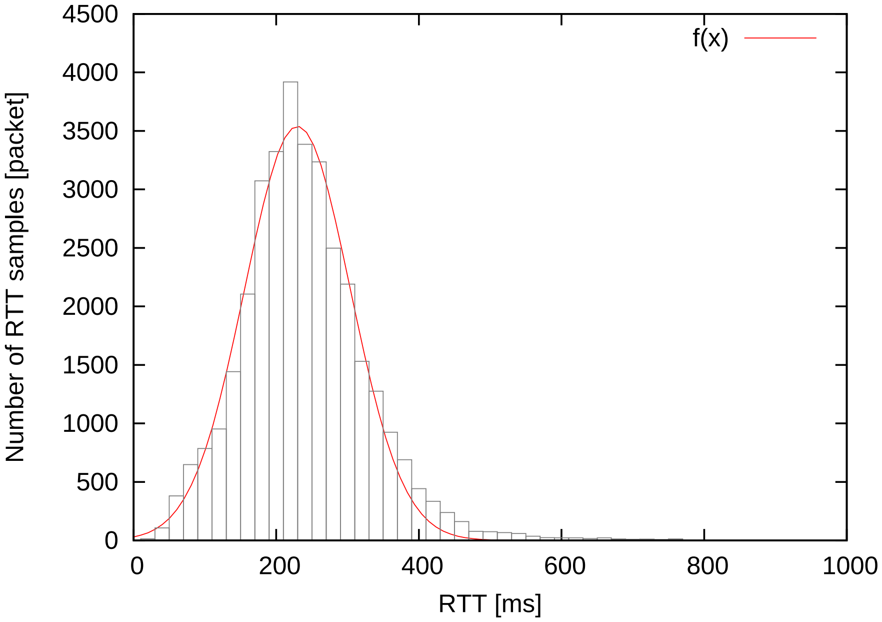
As shown in Fig. 2, RTT distribution of an LTE network has a peak near 220 ms. While it slightly is leaning to the left, the distribution has a bell shape. Regarding a W-CDMA network, we see hundreds of packets has small RTT forming the left tail of the distribution in Fig. 3. On the contrary, there is no packets experiencing small delay in the case of a Wi-Fi network. A reason that the left tail is truncated to 25 ms is that the round-trip propagation delay is about 25 ms in the experiments. Although there are some skewness and truncation, red curves superimposed on the histograms show a good match between the approximation and the actual measurement. In conclusion, we consider that fluctuation of RTT of communication over a wireless network has the Gaussian distribution and there is the possibility of successful application of the attractor perturbation model.

		a	b	c
LTE	Fig. 2(a)	3103.91	243.343	84.9791
	Fig. 2(b)	3539.11	229.747	74.1703
W-CDMA	Fig. 3(a)	4152.94	399.22	65.1429
	Fig. 3(b)	3596.07	398.618	75.3365
Wi-Fi	Fig. 4(a)	5200.37	37.0336	8.47742
	Fig. 4(b)	5153.41	37.7743	8.47124

Table 2: Parameters of fitted Gaussian function

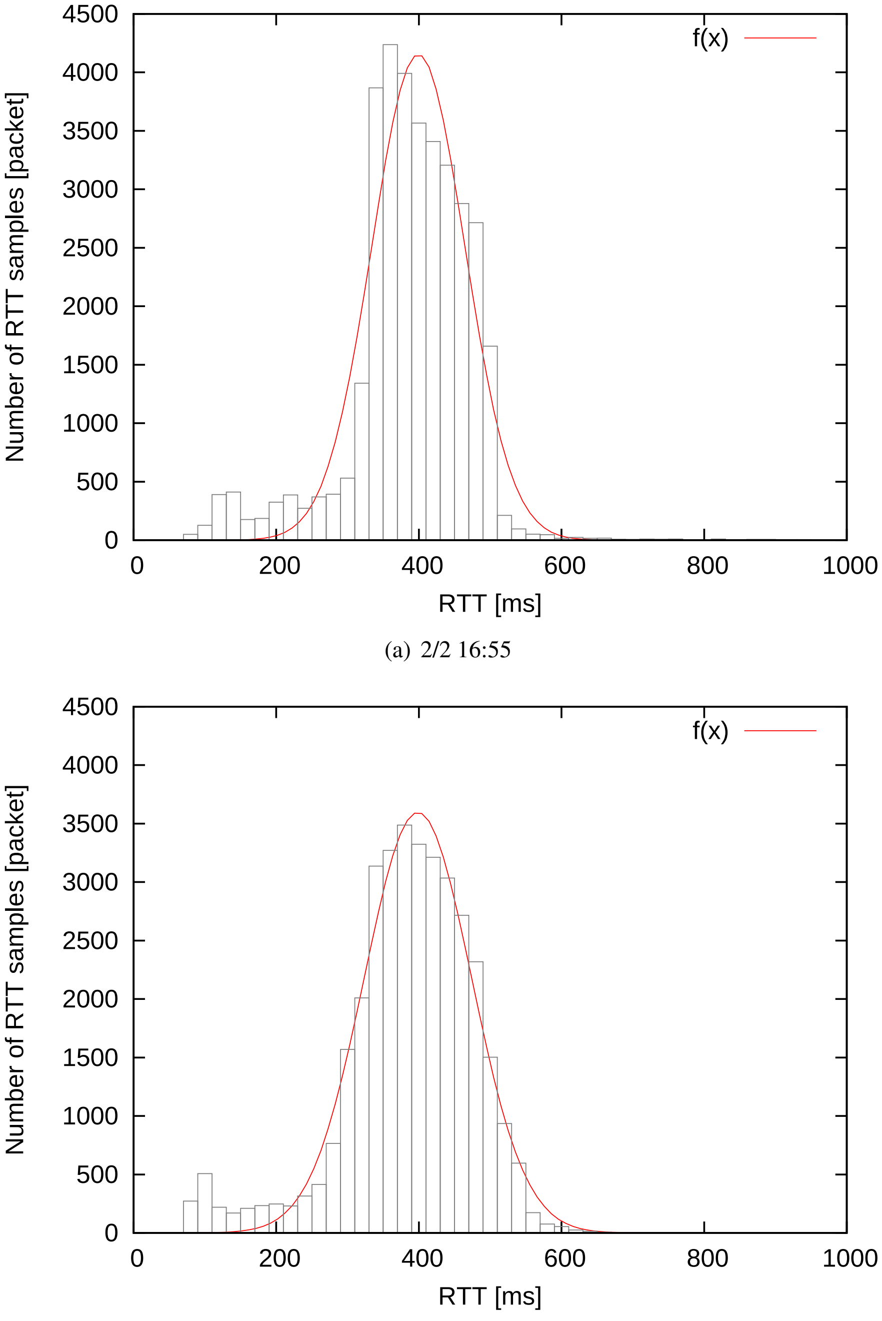


(a) 2/2 16:35



(b) 2/2 16:50

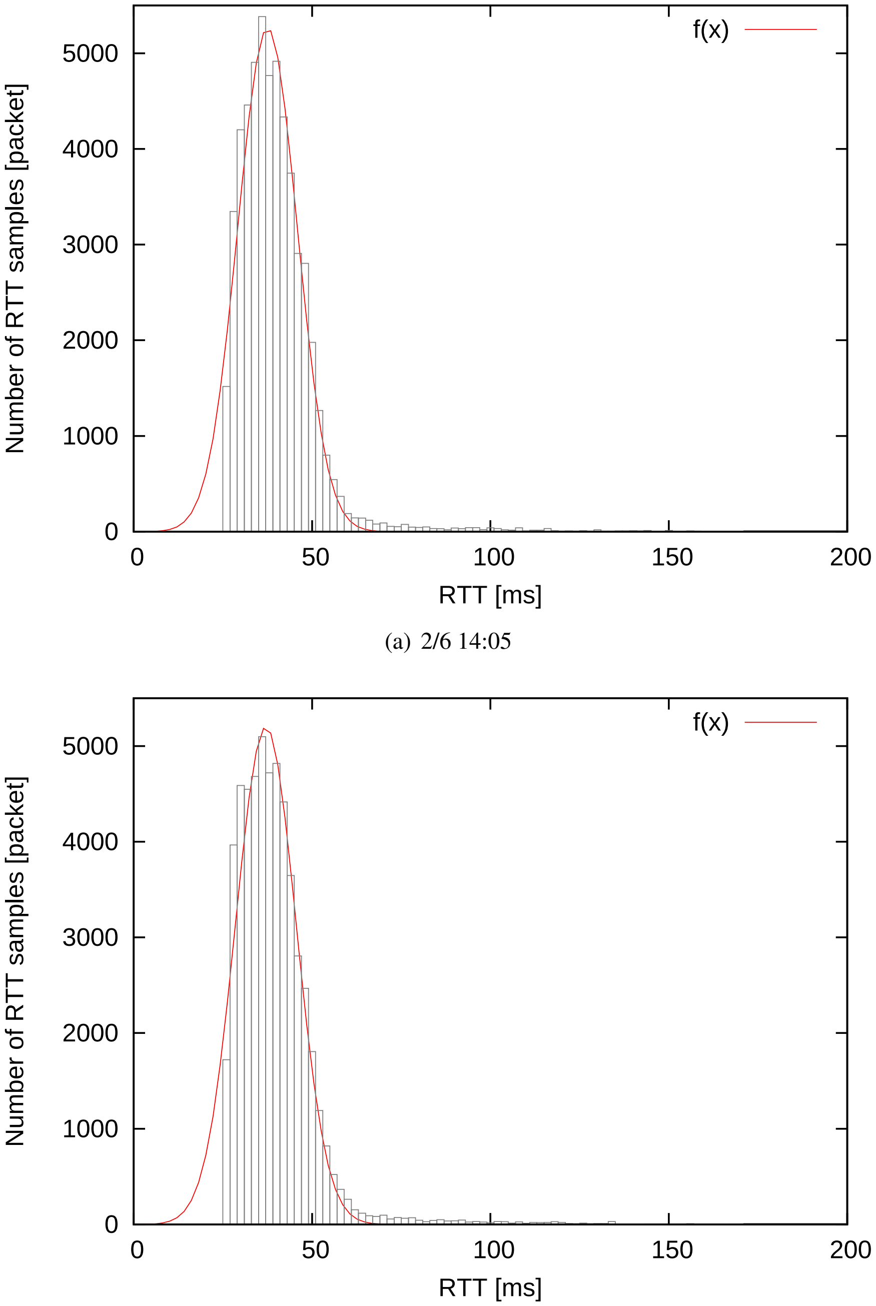
Figure 2: Delay distribution of LTE access network



(a) 2/2 16:55

(b) 2/2 17:10

Figure 3: Delay distribution of W-CDMA access network



(a) 2/6 14:05

(b) 2/6 14:20

Figure 4: Delay distribution of Wi-Fi access network

3 Fluctuations in biological systems

Since biological systems are always exposed to internal and external fluctuation or noise, most cellular variables, such as the quantities of molecules, will vary not only from strain to strain, but from cell to cell. In the work [11], they named the distribution of observed variables in biological experiments as the Gaussian-like distribution. Under the assumption that biological variables follow the Gaussian-like distribution, a mathematical expression, which is the base of the attractor perturbation model, is derived. The attractor perturbation model represents a relationship between fluctuation inherent in biological systems and their response against an external force or stress put on them. Similarly to biological systems, fluctuation is quite common in information networks. There is no static or stable network. Furthermore, the end-to-end delay of a session fluctuates following a Gaussian distribution as verified in Sec. 2. Therefore, we expect that the attractor perturbation model enables an efficient and effective network control mechanism, which does not rely on any additional assumption or knowledge about an information network. In this section, we first gives details of the attractor perturbation model in Sec. 3.1 and describe some examples of existing application of the attractor perturbation model in Sec. 3.2.

3.1 Attractor perturbation model

The attractor perturbation model represents a relationship between fluctuation inherent in biological systems and their response [11]. The following is a mathematical expression of the attractor perturbation model.

$$\langle w \rangle_{a+\Delta a} - \langle w \rangle_a = b \Delta a \sigma_a^2 \tag{2}$$

where $\langle w \rangle_a$ and σ_a^2 are the average and variance of measurable quantity w, e.g. protein concentration, under the influence of the force a, e.g. genetic mutation, respectively. Δa is a small change in the force and b is a constant coefficient. The equation indicates that a shift in the average of a measurable variable against a change in the force is proportional to the variance of the measurable variable.

From Eq. (2), one can derive the following equation.

$$\langle w \rangle_{a+\Delta a} = \langle w \rangle_a + b \Delta a \sigma_a^2 \tag{3}$$

Equation (3) gives an estimate of an influence of an increase in the force a to $a + \Delta a$ when the current average is $\langle w \rangle_a$ and the variance is σ_a^2. From a viewpoint of control of the force, we can further extract the following equation.

$$\Delta a = \frac{\langle w \rangle_{a+\Delta a} - \langle w \rangle_a}{b\sigma_a^2} \tag{4}$$

The equation gives the amount of change in the force, i.e. Δa, or the amount of force, i.e. $a + \Delta a$, to obtain the shifted average $\langle w \rangle_{a+\Delta a}$ from the current conditions $\langle w \rangle_a$, a, and σ_a^2. This brings a basic idea of our proposal.

3.2 Applications of attractor perturbation model

The attractor perturbation model enables a control mechanism to achieve the desired outcome while leaving an information network as a black box. Our research group has adopted the attractor perturbation model to network control. For example in [15], the authors utilize inherent traffic fluctuations to smoothly control the access rate at a gateway node injecting traffic into the core network. For the benefits of the whole system, controlling the rate of the local traffic at a gateway node should be performed not in a strict way, which would lead to timeouts and packet drops, but rather in a smooth way. Based on the attractor perturbation model and only by using instantaneous observation of local buffer occupancy and its variance, the local traffic arrival rate can be controlled moderately, and as a result, the total flow of both local traffic and global traffic at a gateway can be balanced.

In [16], they consider distribution of traffic over multiple paths established on a mobile ad hoc network. Each path i has the injected traffic a_i and the end-to-end delay w_i. The attractor perturbation model is used for a source node to derive the optimal distribution strategy to achieve the minimum average end-to-end delay of all packets traversing different paths to a destination node. From the simulation results, the authors confirm that there is a relationship between the amount of average delay difference and the variance when changing the traffic rate of a session in a mobile ad hoc network.

4 Attractor perturbation in information networks

In this section, in addition to experimental investigation of fluctuation of information networks in Sec. 2, we verify that the attractor perturbation model holds for a network system by mathematical analysis and simulation experiments, where we can manage the experimental condition. We regard the end-to-end delay as the variable w and the rate of injected traffic as the external force a, and try to confirm the linear relationship between fluctuation and response, i.e. the variance of delay and the shift in the average delay.

4.1 Analytical verification of attractor perturbation by M/D/1 queueing model

First in this section, we prove the attractor perturbation model in an M/D/1 queuing system assuming Poisson arrival of fixed-length packets. In the following, λ is the arrival rate, μ is the service rate, and $\rho = \lambda/\mu < 1$ is the traffic intensity or the load.

In [17], the author analyzes the mean time spent in an M/G/1 system, where the service time has a general distribution with mean $E(X)$. The first and second moment of time spent in an M/G/1 system are denoted by $E(T)$ and $E(T^2)$.

$$E(T) = \frac{\lambda}{2(1-\rho)} E(X^2) + E(X) \tag{5}$$

$$E(T^2) = \frac{\lambda}{3(1-\rho)} + \frac{\lambda^2}{2(1-\rho)^2}\{E(X^2)\}^2 + \frac{E(X^2)}{1-\rho} \tag{6}$$

Since the service time in an M/D/1 is constant, by substituting $E(X) = 1/\mu$ and $E(X^2) = 1/\mu^2$ into the above equations, we can obtain the mean $d(\lambda)$ and variance $\sigma^2(\lambda)$ of time spent in an M/D/1 system as functions of the arrival rate λ.

$$d(\lambda) = \frac{2\mu - \lambda}{2\mu(\mu - \lambda)} \tag{7}$$

$$\begin{aligned} \sigma^2(\lambda) &= E(T^2) - \{E(T)\}^2 \\ &= \frac{\lambda(4\mu - \lambda)}{12\mu^2(\mu - \lambda)^2} \end{aligned} \tag{8}$$

By differentiating $d(\lambda)$ with respect to λ we obtain

$$d'(\lambda) = \frac{1}{2(\mu - \lambda)^2} \tag{9}$$

Assuming that $\Delta\lambda$ is small, we further obtain the following relationship.

$$\frac{d(\lambda + \Delta\lambda) - d(\lambda)}{\Delta\lambda} = d'(\lambda) \tag{10}$$

$$\begin{aligned}
d(\lambda + \Delta\lambda) - d(\lambda) &= d'(\lambda)\Delta\lambda \\
&= \frac{1}{2(\mu - \lambda)^2}\Delta\lambda \\
&= \frac{6}{\rho(4 - \rho)}\sigma^2(\lambda)\Delta\lambda \qquad (11) \\
&= b(\rho)\sigma^2(\lambda)\Delta\lambda \qquad (12)
\end{aligned}$$

Therefore, the shift in the mean time spent in an M/D/1 queueing system is given as a product of the variance $\sigma^2(\lambda)$, the change $\Delta\lambda$ of arrival rate, and the coefficient $b(\rho)$. The coefficient $b(\rho)$ is depicted in Fig. 5. Whereas $b(\rho)$ exponentially decreases in the region of $\rho < 0.5$, it can be represented by a constant in the region of $\rho \geq 0.5$. We consider that rate adaptation is necessary especially in a moderately or highly loaded network. Therefore, we can conclude that the attractor perturbation model is applicable to rate control in such an information network.

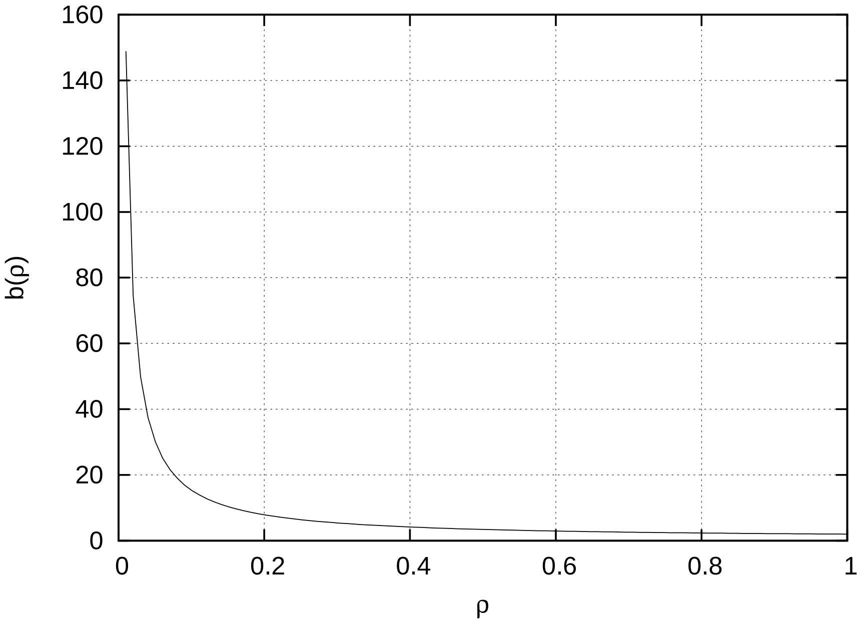


Figure 5: Variation of $b(\rho)$

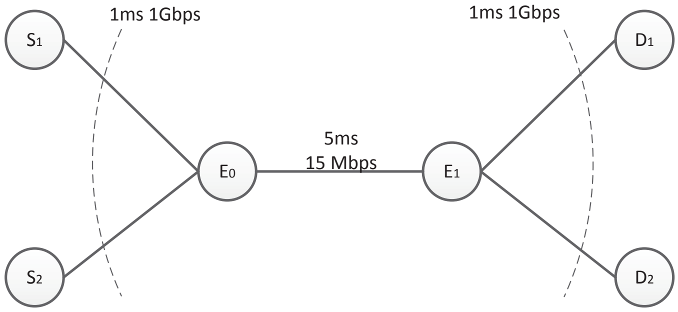


Figure 6: Network topology used in simulation experiments

4.2 Simulation-based verification of linearity between fluctuation and response

In this section, we verify the attractor perturbation model in a packet-based network by simulation experiments using ns-2 [18]. Figure 6 illustrates topology that we used for simulation. The dumbbell network models a bottleneck link of a network of arbitrary topology, which affects the end-to-end delay the most on a path. Two senders S_1 and S_2 are connected with two receivers D_1 and D_2, respectively, through routers E_0 and E_1. All links are full-duplex. The bandwidth and the propagation delay of a link between routers E_0 and E_1 are 15 Mbps and 5ms, respectively. Those of the other links are 1 Gbps and 1ms.

A drop-tail FIFO buffer with the capacity of 1000 packets is deployed on each router. A CBR session called "session 1" is established between nodes S_1 and D_1. We observe the one-way end-to-end delay on session 1 while changing the sending rate of UDP datagrams of a 1000-bytes payload. As background traffic, another UDP session, where the inter-arrival time of datagrams follows the exponential distribution and the payload size of a datagram is 1000 bytes, is set between nodes S_2 and D_2. It is called "session 2".

We observe the average $\langle w \rangle_a$ and variance σ_a^2 of one-way end-to-end delay of session 1 at the sending rate a Mbps. We prepared 10 traffic patterns of session 2 whose sending rate is 9 Mbps. For each of the pattern, we conducted 44 simulation experiments by increasing the sending rate a from 0.1 Mbps to 4.5 Mbps by 0.1 Mbps, i.e. $\Delta a = 0.1$. Then, from averages and variance obtained from 440 simulation experiments, we derive 430 pairs of σ_a^2 and $\langle w \rangle_{a+0.1} - \langle w \rangle_a$, i.e. $\sigma_{1.0}^2$ and $\langle w \rangle_{1.1} - \langle w \rangle_{1.0}$.

If the attractor perturbation model holds, there exists the linear relationship between $\Delta a \cdot \sigma_a^2$ and $\langle w \rangle_{a+\Delta a} - \langle w \rangle_a$ as Eq. (2) indicates. 430 pairs of $0.1\sigma_a^2$ and $\langle w \rangle_{a+0.1} - \langle w \rangle_a$ are plotted on

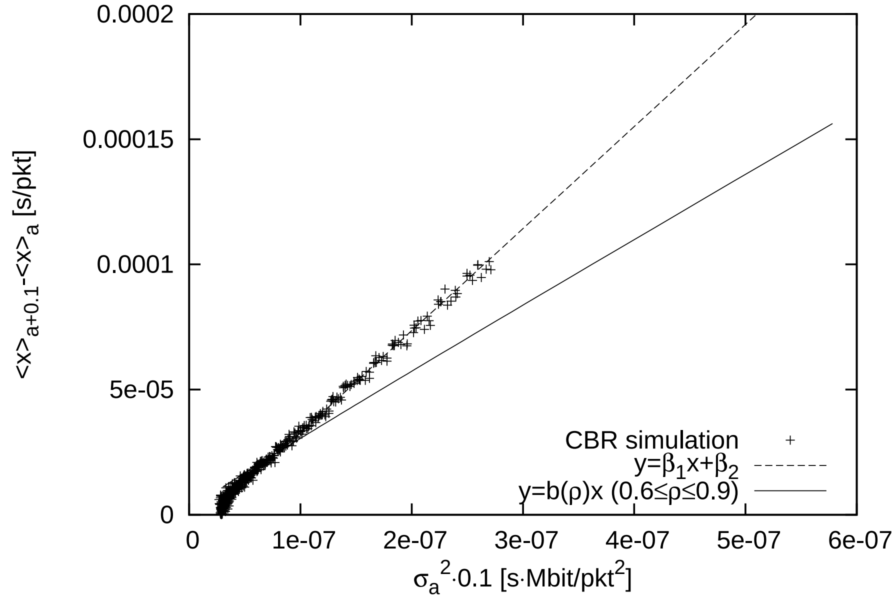


Figure 7: Attractor perturbation relationship of CBR traffic

Fig. 7 as crosses. The figure shows the positive correlation between $0.1\sigma_a^2$ and $\langle w\rangle_{a+0.1} - \langle w\rangle_a$ and we can confirm the attractor perturbation model in a packet-based network. When the sending rate of CBR session is low, there is little chance for packets to experience buffering at routers. As a result, the variance becomes small and the small increase of sending rate does not affect the delay much. Therefore, when the variance is small, the shift in the average delay becomes small as well. On the contrary, as the sending rate increases, the number of packets buffered at routers begins to fluctuate. It leads to both of the larger delay and the variance. Consequently, we observe the linear relationship between the variance and the shift in delay.

The proportional constant of the relationship between $0.1\sigma_a^2$ and $\langle w\rangle_{a+0.1} - \langle w\rangle_a$ corresponds to the coefficient b of Eq. (2). In Fig. 7, we show an approximate line $y = \beta_1 x + \beta_2$ obtained by the least squares approximation where x is $0.1\sigma_a^2$ and y is $\langle w\rangle_{a+0.1} - \langle w\rangle_a$. The slope, i.e. β_1, of the line can be regarded as the coefficient b, and its value is 407.63. The load ρ at the variance σ_a^2 is calculated by λ/μ, where μ is the service rate of the bottleneck link and λ is the arrival rate when the variance is σ_a^2. Therefore, ρ depends on σ_a^2 and the x-axis can be mapped to ρ. In Fig. 7, $y = b(\rho)x$ in the range of $0.6 < \rho < 0.9$ is depicted. To compare the analytical result of an M/D/1 system discussed in the previous section, we convert $b(\rho)$ to $\frac{750}{\rho(4-\rho)}$ by $\Delta\lambda = \frac{\Delta a \times 10^6}{8000}$

in Eq. (11). Although it is not a linear function due to the variation of ρ, the slope $b(\rho)$ is about 300 on average in the range of $0.6 < \rho < 0.9$. As shown in Fig. 7, there is a difference in slope between the analytical result and the simulation result. For the same variance, the shift $\langle w \rangle_{a+0.1} - \langle w \rangle_a$ is larger in the simulation than in the analysis. Given the variance σ_a^2, the load on a network in the case of the analysis, which can be derived from Eq. (8), is smaller than that of the simulation, which can be derived as $(a + 9)/15$ considering that the amount of background traffic is 9 Mbps and the capacity of the bottleneck link is 15 Mbps. In general, when the sending rate increases, the end-to-end delay becomes larger in a congested network than in an unloaded network. Consequently, the growth rate or the slope is larger in the simulation than in the analysis. In Sec. 6, we used three alternatives of coefficient b, that is, 407.63, 300, and $b(\rho)$, to evaluate its influence.

5 Rate control mechanism with attractor perturbation model

In this section, we propose a novel rate control mechanism based on the attractor perturbation model. We regard the end-to-end delay as the measurable variable w and the sending rate as the force a. Then a sender derives the appropriate sending rate to accomplish the target delay under the fluctuating environment.

5.1 Details of rate control mechanism with attractor perturbation model

We consider an application which lasts at least for several minutes to have room for measurement and rate control. An application specifies the target one-way delay T s, the maximum sending rate a_{max} Mbps, and the minimum sending rate a_{min} Mbps. RTP/UDP and RTCP/UDP are employed and the sending rate is adjusted by adapting a transmission interval of RTP packets. Figure 8 illustrates how packets are exchanged between a sender and a receiver and the sending rate is adjusted at a sender.

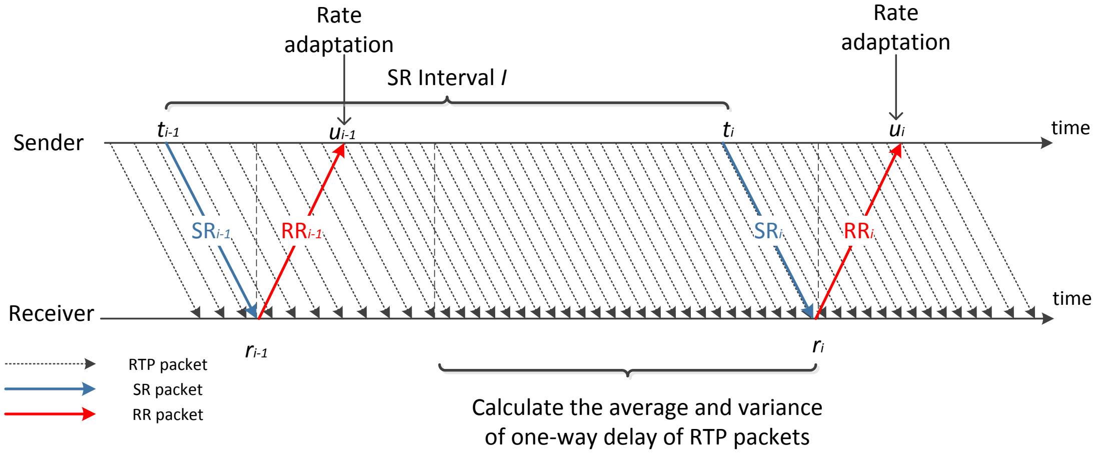


Figure 8: Outline of proposal

At the beginning of a session, a sender sends RTP packets at the minimum rate a_{min} Mbps. In our proposal, a sender periodically tries to adjust the sending rate according to the average and variance of end-to-end one-way delays of RTP packets. In order to calculate average and variance, a receiver memorizes the one-way delay and timestamp of each received RTP packet.

The average and variance of measured one-way delay of RTP packets is calculated by a receiver based on the recorded timestamp of RTP packets which are sent after the last rate adaptation, as will be explained below. At regular intervals of the SR interval I, a sender sends a sender report (SR) packet to request a receiver to report the average and variance. When a receiver receives an SR packet, it sends back a Receiver Report(RR) packet which contains the average and variance of one-way delay in its extended header. Then, on receiving a RR packet, a sender calculates the next sending rate based on the informed average and variance.

Now assume that a sender sends the i-th SR packet at t_i s and it is received by a receiver at r_i s. We further assume that the sender received the RR packet corresponding to the i-th SR packet from the receiver at u_i s. At that time, the sender derives and adopts the new sending rate, which is denoted as $a(i)$ Mbps. For the sender to derive the appropriate sending rate $a(i)$, on receiving the i-th SR packet, the receiver first calculates the average d_{i-1} and variance v_{i-1}^2 from the memorized one-way delay of RTP packets which is sent at $a(i-1)$ Mbps. So that a receiver can distinguish RTP packets sent at $a(i-1)$ Mbps, the i-th SR packet carries the information about the time of the last rate adaptation, e.g., u_{i-1} s. Then, a receiver consider that RTP packets having timestamp after u_{i-1} were sent at the rate $a(i-1)$ Mbps. However, there is possibility that an SR or a RR packet is lost in a network as shown in Fig. 9. In this case, time of the last rate adaptation is u_{i-2}. Although it is possible to derive the average and variance of delays from RTP packets having timestamp from u_{i-2} to t_i, it means that the observation interval would change from time to time. To have consistent statistics among control intervals, we consider another mechanism. A sender places not u_{i-1} s but t_{i-1} s in a header of the i-th SR packet. From t_{i-1}, a receiver estimates a hypothetical u_{i-1}, an instant when a sender successfully receiving a RR packet changes the sending rate. Since it is not possible for a receiver to know the one-way delay from itself to a sender, we assume that one-way delay is identical independently of the direction.

$$r_i - t_i \approx u_i - r_i$$

The instant u_{i-1} s of the last rate adaptation can be approximated as follows.

$$\begin{aligned} u_{i-1} &= t_{i-1} + (r_{i-1} - t_{i-1}) + (u_{i-1} - r_{i-1}) \\ &\approx t_{i-1} + 2(r_{i-1} - t_{i-1}) \qquad (13) \end{aligned}$$

In the case of loss of the $(i-1)$-th RR packet, Eq. (13) is eligible. However, when the $(i-1)$-th SR packet is lost, a receiver does not know r_{i-1}. Therefore, we make another assumption that the one-way delay of the $(i-1)$-th SR packet is identical to that of the i-th SR packet.

$$r_{i-1} - t_{i-1} \approx r_i - t_i$$

Based on the assumption a receiver can estimate the instant u_{i-1} s of the last rate adaptation as follows.

$$\begin{aligned} u_{i-1} &= t_{i-1} + (r_{i-1} - t_{i-1}) + (u_{i-1} - r_{i-1}) \\ &\approx t_{i-1} + 2(r_{i-1} - t_{i-1}) \\ &\approx t_{i-1} + 2(r_i - t_i) \end{aligned} \tag{14}$$

Finally, a receiver calculates the average d_{i-1} and variance v_{i-1}^2 of the one-way delays of RTP packets which are received before r_i s and have timestamp after $t_{i-1} + 2(r_i - t_i)$ s. Since packets whose timestamp is before $t_{i-1} + 2(r_i - t_i)$ s are considered to be sent before the last rate adaptation, they are excluded from calculation. Then, the receiver sends a RR packet carrying the calculated average and variance in an extended header.

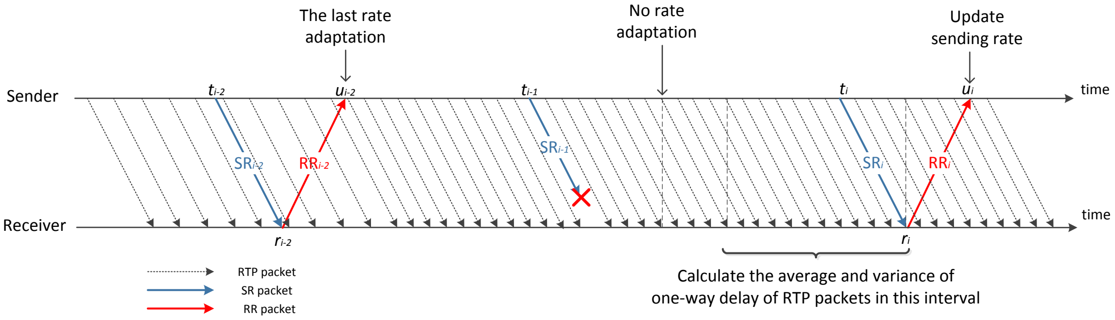


Figure 9: Loss of SR packet

On receiving a RR packet, a sender first calculates the amount Δa Mbps of rate change by substituting the received statistics, the target delay T, and the coefficient b to the following equation.

$$\Delta a = \frac{T - d_{i-1}}{bv_{i-1}^2} \tag{15}$$

Next, the sender updates the sending rate to $a(i)$ Mbps, which is derived from the following equation.

$$a(i) = \min\{a_{max}, \max(a_{min}, a_{cur} + \Delta a)\}, \tag{16}$$

where a_{cur} is the current sending rate.

If a sender does not receive any of the $(i-n)-$th RR packets ($n \in 1, 2, 3$) by t_i+1+I s, i.e. an instant to send the i-th SR packet, it considers that a network is considerably congested. Then, the sender reduces the sending rate by half and quits sending the i-th SR packet at $t_i + 1 + I$ s. After additional I s, the sender sends the i-th SR packet carrying $t_i - 1 + I$ s in an extended header to a receiver. On receiving the SR packet, the receiver calculates the average and variance from one-way delay of RTP packets were received before r_i s and have timestamp after $t_{i-1} + I + 2(r_i - t_i)$ s. After that the receiver sends a RR packet to the sender.

5.2 Discussion of our proposal

For the rate control with the attractor perturbation model, it is also possible to use a round-trip end-to-end delay instead of a one-way delay as the measured variable. However, the rate adaptation based on the round trip time could lead to a problem as discussed in the following.

So that a sender can obtain the information about RTT, there must be two-way communication. Since in our proposal, a sender and a receiver exchange SR and RR packets, it is possible for a sender to estimate RTT by observing time from emission of an SR packet and to reception of a corresponding RR packet. However, the frequency is not high enough to get the sufficient number of samples for reasonable estimation. For example, in RFC 1889 which specifies RTP and RTCP, it is suggested that RTCP should occupy 5% of the session bandwidth and an interval between RTCP packets must be larger than 5 s. It is too long to catch up with delay variation caused by dynamic traffic changes. Specifically, coarse granularity of sampling spoils the accuracy of variance. It might be possible to shorten the interval to a much smaller value to increase the sampling frequency, but it puts a strain on a network and even disturbs primary data communication. On the contrary, our proposal adopts the one-way delay which can be obtained from RTP packets at a receiver. The sampling frequency is 13 packets/s at the sending rate of 0.1 Mbps and 2048 packets/s at 15 Mbps, respectively. Furthermore, such frequent sampling does not involve additional overhead except for exchanges of RTCP packets to retrieve statistics every several seconds.

6 Evaluation scenarios

We verify that our proposal can achieve and maintain the target delay even when background traffic changes through simulation experiments. In Sec. 6.1, we describe the simulation settings including a method used for comparison. Then, we explain evaluation criteria used in evaluation in Sec. 6.2.

6.1 Simulation settings

We first explain a pseudo rate control mechanism with a delay-based AIMD algorithm used as a benchmark. Next we describe a simulation model and measures.

6.1.1 Delay-based AIMD rate control mechanism

The AIMD (Additive Increase Multiplicative Decrease) is a well-known and common algorithm employed by many rate control protocols [19, 20]. For example, the most conventional transmission protocol, TCP Reno, decreases the sending rate by having the congestion window size when congestion is detected, while linearly increases the sending rate in the congestion avoidance phase. Whereas there are many variants of rate adaptation algorithms and it is pointed out that a simple AIMD suffers from several issues, it is yet simple and effective. Therefore, as a benchmark used for comparison purposes, we consider a pseudo rate control mechanism with a delay-based AIMD algorithm.

In the delay-based AIMD rate control, a sender obtains the average d_{i-1} of one-way-delay of RTP packets by exchanging RTCP packets at regular control intervals. Then, it derives the new sending rate a_{new} Mbps by using the following equation.

$$a_{new} = \begin{cases} a + A & (d_{i-1} < T) \\ B \times a & (d_{i-1} \geq T), \end{cases} \tag{17}$$

where T is the target delay, A $(A > 0)$ is the additive increase factor, and B $(0 < B < 1)$ is the multiplicative decrease factor. Here, both A and B are constant values.

6.1.2 Network topology and traffic

We used the dumbbell topology depicted in Fig. 6 and set a UDP session same as Sec. 4.2. At the beginning of a simulation run, session 2 emits UDP packets at the sending rate of 9 Mbps. Then, it increases the sendig rate to 10.5 Mbps at 200 s and keeps the rate until the end of the simulation run, i.e. 400 s. The increase in the load is from 0.6 to 0.7. We employ our proposal on session 1 established between nodes S_1 and D_1. The size of a RTP packet including RTP, UDP, and IP headers is set at 1000 bytes. The sizes of an SR packet and a RR packet including an IP header are 64 and 72 bytes, respectively. The maximum sending rate a_{max} and the minimum sending rate a_{min} of our proposal are 15.0 Mbps and 0.1 Mbps, respectively. The interval I of SR packet transmission is 10 s. The target delay is set at 8.2 ms, which is the one-way delay observed in the simulation experiments of the case of $\rho = 0.8$ in Sec. 4.2. Parameters used in evaluations are summarized in Table 3.

In order to evaluate the influence of the coefficient b, we conduct simulation experiments with b=300, 407.63, and function $b(\rho)$. $b(\rho)$ enables dynamic adaptation of b with respect to the load condition. In the case of $b(\rho)$, we assume that a sender always knows the current load ρ of a network to derive the appropriate rate change Δa, whereas it is not possible to have the accurate and up-to-date information about the load condition of a network in an actual situation. More specifically, at t_i s, when a sender sends the i-th SR packet, the average load ρ_i on the bottleneck link from t_{i-1} s to t_i s is given and substituted into $b(\rho_i)$.

For comparison purposes, we additionally conduct simulation experiments for the cases of CBR traffic in Sec. 7.1. In those cases, session 1 generates CBR traffic at 3 Mbps or 0.8 Mbps using RTP and RTCP. Note that a pair of SR and RR packets is sent every 10 s for fair comparisons, but they are not used for rate control. We denote a case of CBR traffic with sending rate of 3 Mbps as "CBR 3 Mbps" and that of 0.8 Mbps as "CBR 0.8 Mbps".

Furthermore, we compare our proposal with a delay-based AIMD rate control mechanism explained in Sec. 7.2. The value of B is fixed at 1/2, but the value of A is set at 1 Mbps, 0.8 Mbps, 0.4 Mbps, 0.2 Mbps, or 0.1 Mbps. We denote a case of delay-based AMD rate control with the value of A as "AIMD A" like "AIMD 0.1 Mbps". Since the delay-based AIMD rate control mechanism has a long transient state, we consider the behavior of the delay-based AIMD rate control at a steady state.

Table 3: Parameter settings

parameter	value
a_{min}	0.1 [Mbps]
a_{max}	15 [Mbps]
Interval I of SR packets	10 [s]
T	8.2 [ms]
b	$300, 407.63$

6.2 Evaluation criteria

To evaluate how our proposal achieves and maintains the target delay, we introduce the mean square error, the coefficient of variation, and the delay jitter defined in the following. We consider the first control interval after the initial transient state as the 0-th interval.

6.2.1 Mean square error

We evaluate the closeness to the target delay by the mean square error. First, we calculate the average delay T_i of successfully received RTP packets that are sent in the i-th control interval from t_i s to t_{i+1} s. Note that T_i is not equal to d_i, which is the average delay defined in Sec. 5. Then, we obtain the mean square error M as follows.

$$M = \frac{1}{n+1}\sum_{i=0}^{n}(T_i - T)^2 \tag{18}$$

Here T is the target delay, n is the number of SR packets sent in the whole simulation time. Therefore, T_n is the average delay of RTP packets that are sent from t_n s to the end of the simulation. A small M means that the average delay is close to the target delay in most of cases.

6.2.2 Coefficient of variation

We evaluate the stability of the average delay by the coefficient of variation. We calculate the mean $\bar{T}$ and the standard deviation σ^2 of the average delay in the simulation as below.

$$\bar{T} = \frac{1}{j+1}\sum_{i=0}^{n} T_i \tag{19}$$

$$\sigma^2 = \sqrt{\frac{1}{n+1}\sum_{i=0}^{n}(T_i - \bar{T})^2} \tag{20}$$

Then we obtain the coefficient C of variation as follows.

$$C = \frac{\sigma^2}{\bar{T}} \tag{21}$$

A small C means that the average delay is kept constant and stable.

6.2.3 Delay jitter

We define the delay jitter J as follows.

$$J = \max_{0 \leq i \leq n} \{|T_i - T|\} \tag{22}$$

The delay jitter is the maximum difference between the target delay T and the average delay T_i.

7 Evaluation results

7.1 Basic evaluation

First we show an example of temporal variations in Fig. 10 and 11. In Fig. 10, variations of average delay T_i against the simulation time are depicted. In Fig. 11, variations of averaged sending rate per control interval are depicted. All results in the figures are obtained from simulation experiments with the identical background traffic pattern.

As shown in Fig. 10, CBR 3.0 Mbps results in the average delay close to the target delay at the beginning, but the delay becomes larger after the increase of background traffic. On the contrary, the average delay of CBR 0.8 Mbps is as low as the target delay from 200 s, whereas it is smaller than the target delay in the first half on the simulation run. Regarding our proposal, independently of the setting of coefficient b, the average end-to-end delay stays close to the target delay except for the period right after the sudden load increase. In the case of $b = 300$ for example, a sender node tries to decrease the sending rate on reception of a RR packet from a receiver node at 201 s. However, the decrease is only 0.33 Mbps at that time as shown in Fig. 11. It is because the delay and variance informed by the RR packet are derived from RTP packets sent before the load increase. At the next timing of rate control at 211 s, delay and variance have grown much to 9.07 ms and 3.23 ms^2 respectively . Then, the amount of decrease derived at the sender node becomes 0.90 Mbps. As a result of drastic rate reduction, the obtained end-to-end delay approaches the target delay again. The instantaneous increase of delay is basically unavoidable, but the duration can be shorten by a shorter control interval, that is, frequent rate control. However, too short control interval decreases the accuracy of variance derivation as a statistic and a sender node cannot precisely capture the fluctuation of a network. We should emphasize here that setting of constant value b does not affect the performance of our proposal very much. It suggests that even when a network condition changes, the sender can keep using the same b without unacceptable performance deterioration.

Figure 12 summarizes results of all simulation experiments conducted 30 times for each settings. Fig. 12(a) and Fig. 12(b) show the relationship between the coefficient of variation and the mean square error, and that between the coefficient of variation and the delay jitter, respectively. In both figures, the closer the point is to the origin, the more stable the end-to-end delay is around the target delay. Figure 12 shows that points of our proposal overlap with each other independently

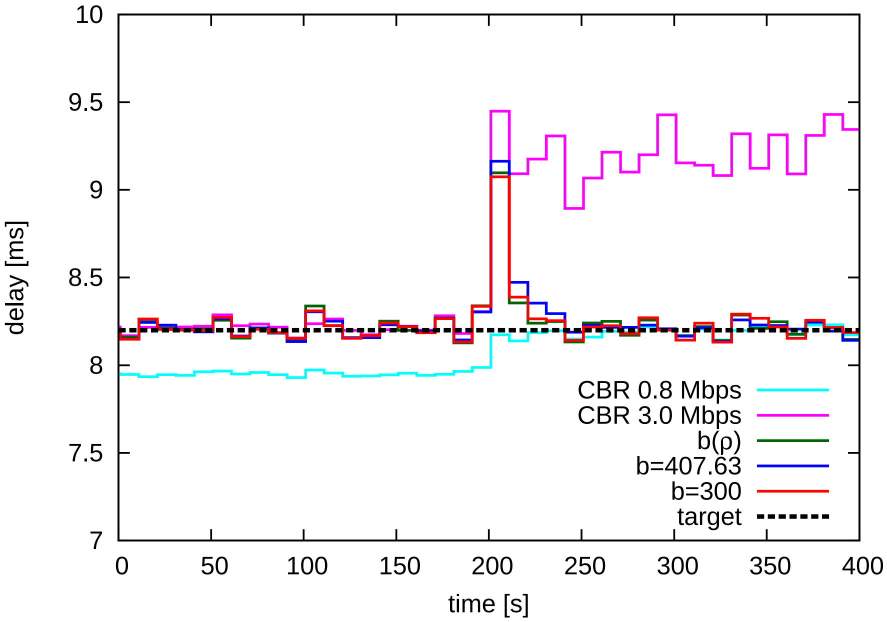


Figure 10: Comparison of average delay

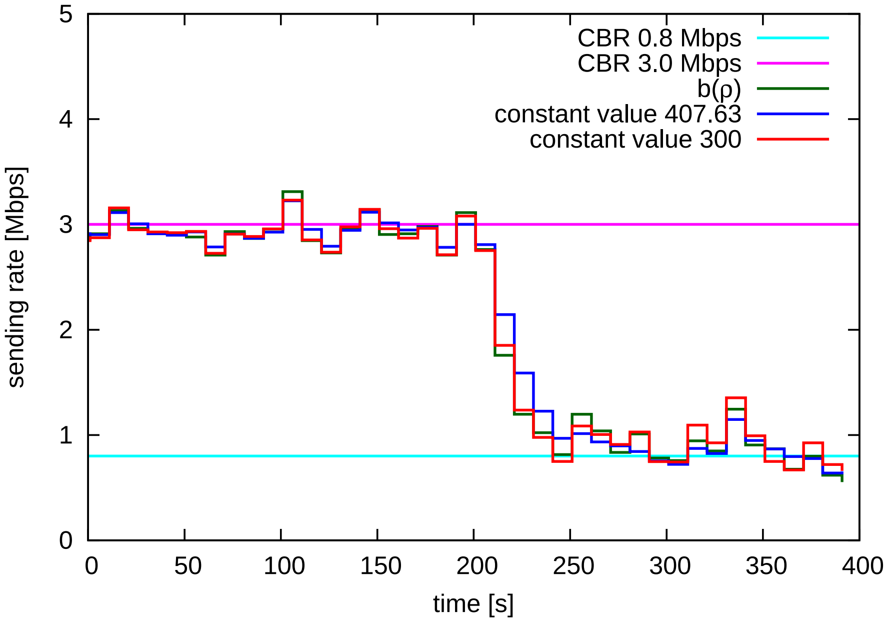
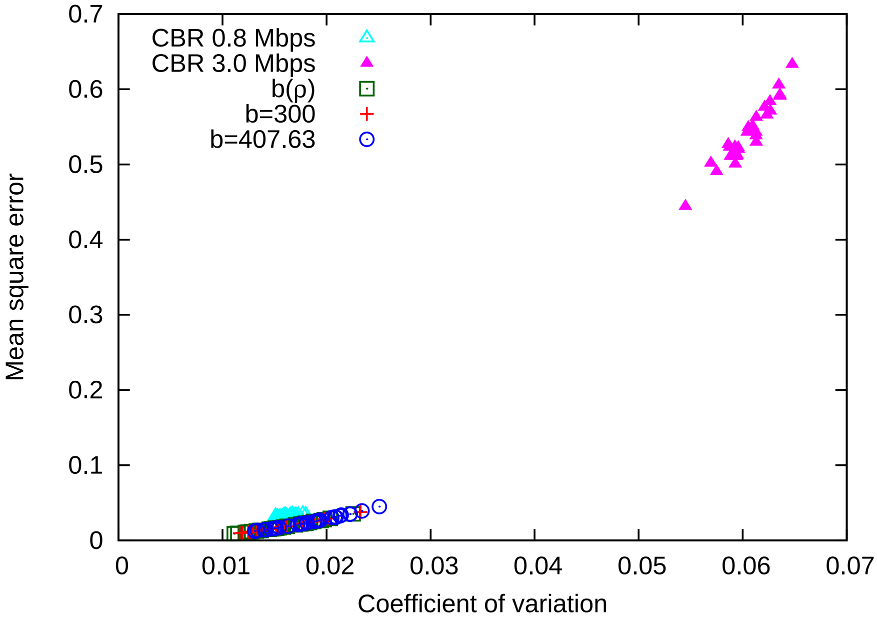
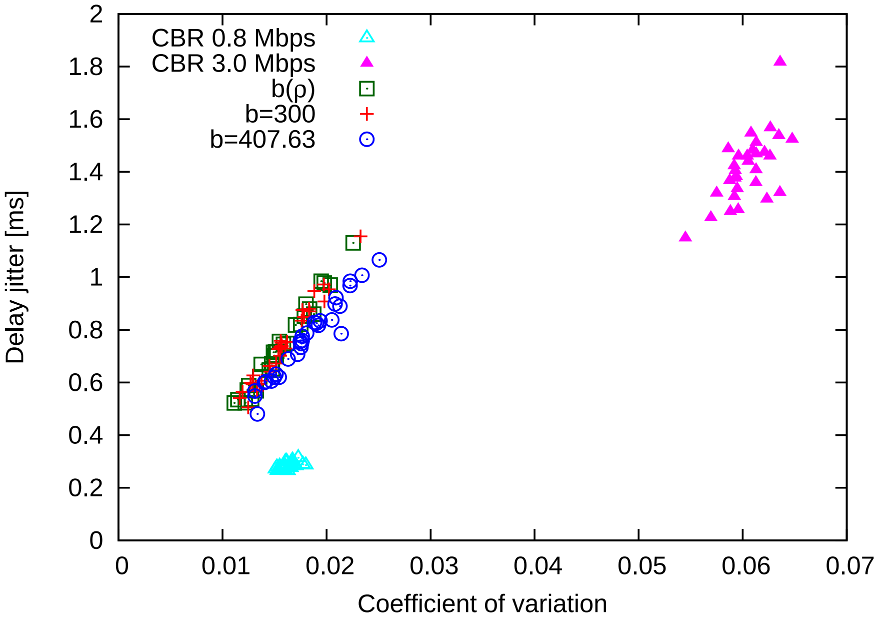


Figure 11: Comparison of sending rate



(a) Coefficient of variation and mean square error



(b) Coefficient of variation and delay jitter

Figure 12: Performance comparisons

setting of the coefficient b and they are in the lower left region. MSE of our proposal is smaller than that of CBR 0.8 Mbps and much smaller than that of CBR 3.0 Mbps. On the other hand, our proposal results in larger coefficient of variation in some cases and larger delay jitter in all cases than CBR 0.8 Mbps. A reason is that the average delay does not change much before and after the load increase due to the low sending rate with CBR 0.8 Mbps. In contrast, our proposal suffers from the instantaneous increase of delay after the load increase. It makes the delay jitter larger than that of CBR 0.8 Mbps and affects the coefficient of variation as well. From a practical view point, the delay jitter of as much as 1.2 ms on a session of the propagation delay of 15 ms is small enough.

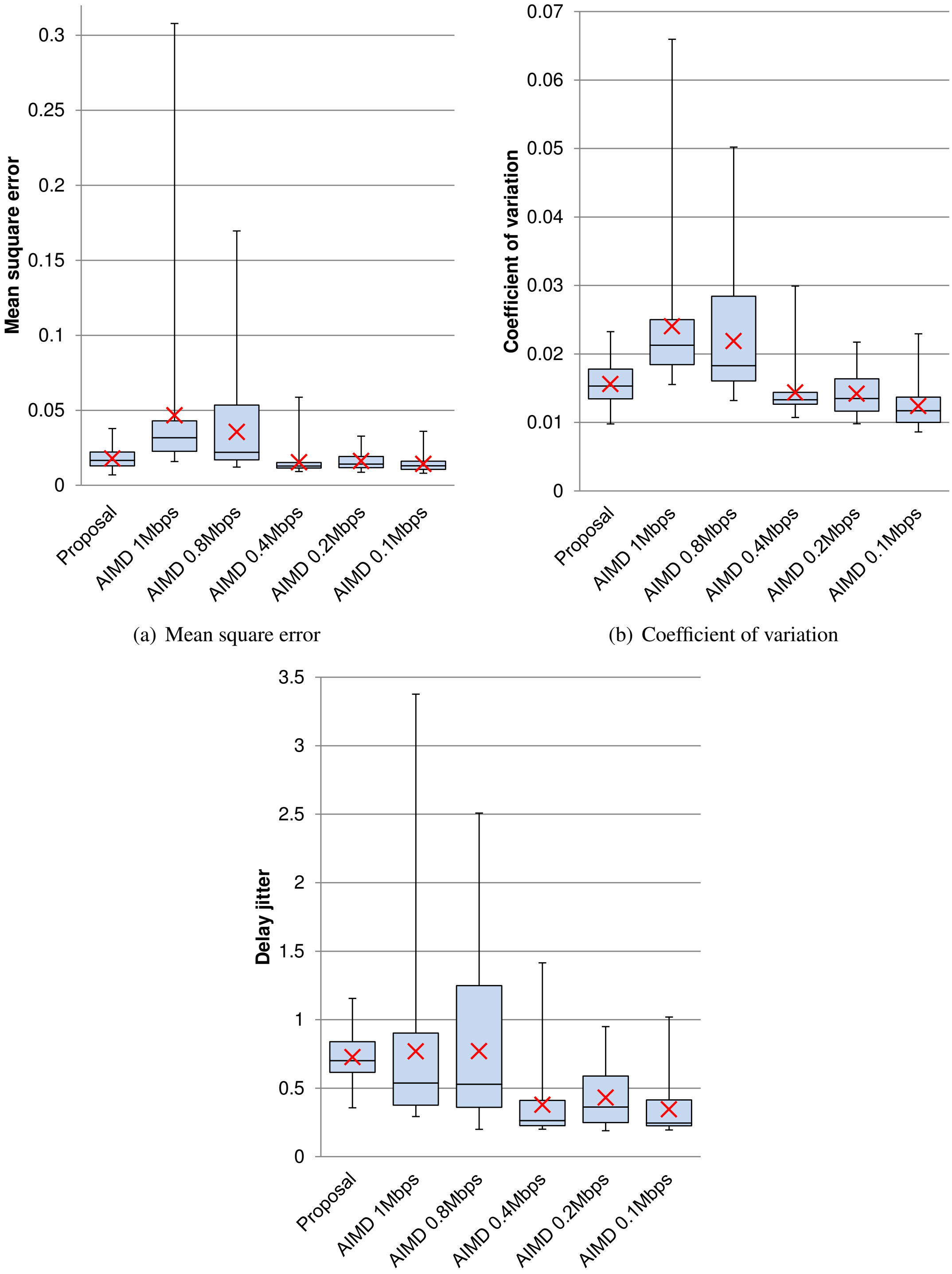
In summary, we can conclude that our proposal can accomplish the stable end-to-end delay facing to the sudden load increase except for the instantaneous growth of delay right after the increase. We further showed that the setting of coefficient b did not influence rate control very much, which supports our motivation not to rely on the detailed knowledge about a network and its dynamics. That is, our proposal is insensitive to parameter setting as can be seen in the flexibility and robustness of biological systems.

7.2 Comparison with delay-based AIMD rate control

In this section, we show summarized results of 100 simulation experiments to compare our proposal with $b = 300$ and the delay-based AIMD control mechanisms. The simulation settings are the same as in Sec. 7.1.

Figure 13 illustrates box-and-whisker plots showing statistical properties of evaluation criteria. Cross marks represent the average of evaluation criteria. The bottom and top of a box are the 25th and 75th percentile, respectively. The band near the middle of a box is the 50th percentile (the median). The ends of the whiskers represent the minimum and maximum values, respectively.

Regarding the delay-based AIMD rate control, as the additive increase factor A becomes large, evaluation criteria have wider distribution and the average becomes larger. That is, the performance deteriorates. When A is large, a sender increases the sending rate by the large amount. It is likely that the increase is too much for the load condition and the delay exceeds the target. Then the sender drastically decreases the sending rate by half at the next control timing. Consequently, the sending rate considerably fluctuates and it results in the unstable delay. On the contrary, the delay-based AIMD rate control with small A regulates the sending rate more moderately. Al-

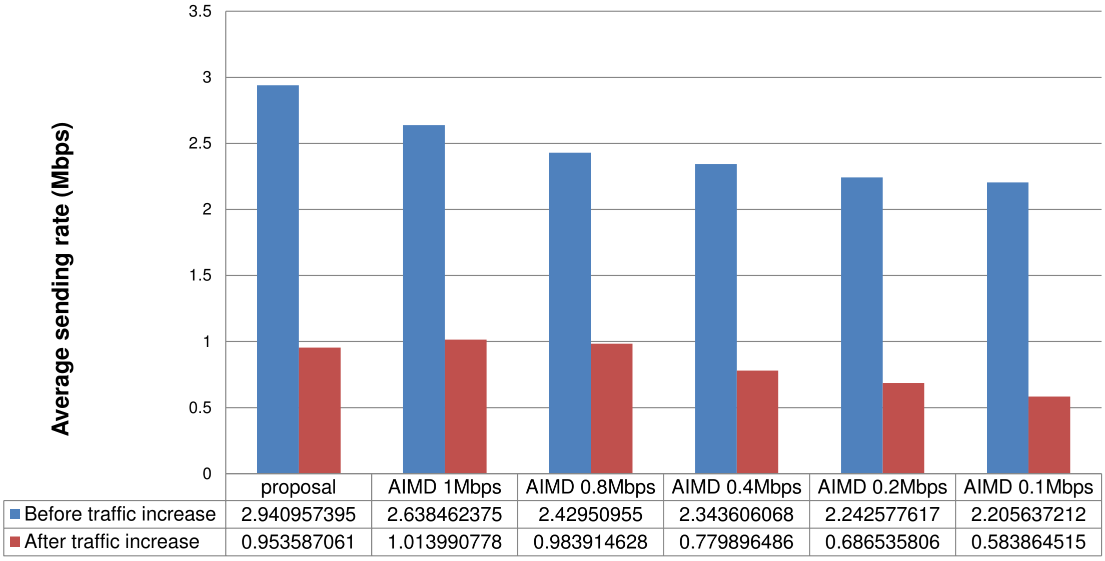


(a) Mean square error

(b) Coefficient of variation

(c) Coefficient of variation and delay jitter

Figure 13: Average and distribution range



	proposal	AIMD 1Mbps	AIMD 0.8Mbps	AIMD 0.4Mbps	AIMD 0.2Mbps	AIMD 0.1Mbps
🟦 Before traffic increase	2.940957395	2.638462375	2.42950955	2.343606068	2.242577617	2.205637212
🟥 After traffic increase	0.953587061	1.013990778	0.983914628	0.779896486	0.686535806	0.583864515

Figure 14: Comparison of average sending rate of evaluation criteria

though it takes time to accomplish the stable sending rate even under the static load condition, the observed delay has small variation. From Fig. 13, AIMD 0.4 Mbps, 0.2 Mpbs, and 0.1 Mbps outperforms our proposal. However, the small variation is achieved at the sacrifice of the sending rate, i.e. throughput, as will be shown next.

In addition to criteria defined in Sec. 6.2, we evaluate the average sending rate before and after the background traffic increase and results are summarized in Fig. 14. Based on the result in Sec. 7.1, we learned that the delay of CBR 3 Mbps is close to the target delay before background traffic increases. It means that a sender can transmit RTP packets at approximately 3 Mbps during that period while fulfilling the delay constraint. From Fig. 14, it is apparent that the average sending rate of our proposal is mostly 3 Mbps before the background traffic increase. On the contrary, independently of a value of A the delay-based AIMD has the smaller sending rate and smaller A results in smaller sending rate. The sending rate of our proposal is higher than that of the delay-based AIMD control by 10.29 % to 25.00 %. The sending rate of our proposal after the increase of background traffic is as high as that of AIMD 1 Mbps and 0.8 Mbps, but those AIMD control have larger fluctuations in all evaluation criteria as shown in Fig. 13. In conclusion, despite for variations caused by instantaneous delay increase right after the increase of background traffic, our proposal has better performance than the delay-based AIMD control. Our proposal can achieve and maintain the target delay, keeping the sufficiently high sending rate, under the influence of traffic changes.

8 Conclusion and future work

In this paper, as an example of application of the attractor perturbation model, we propose a novel rate control mechanism to achieve and maintain the target delay in the dynamically changing environment. First we confirmed that the delay distribution of communication over wireless access networks was similar to the Gaussian from the experimental measurement. Next we proved that the attractor perturbation model held in a packet-based network as well as a general M/D/1 queuing system. Then, based on the confirmation of the attractor perturbation model in information networks, we proposed a rate control mechanism. Through simulation experiments, we confirmed that our proposal could achieve and maintain the target delay and outperformed to the delay-based AIMD rate control and CBR. More interestingly, we found the setting of coefficient b did not influence the performance of proposal very much.

However, it is an overstatement to say that our proposal does not need any parameter tuning. In this paper, our proposal used the coefficient b which had been derived by analysis and simulation in Sec. 4. The coefficient b would be influenced by characteristics of a network including the size, topology, and competing sessions. There are several strategies to refine b to fit to the actual network condition. One is to conduct preliminary experiments as we did in this thesis. Before or at the beginning of data transfer, a sender observes the end-to-end delay by sending probe packets or using the first bunch of data packets while changing the sending rate for a certain period. Next based on collected samples, the sender derives the coefficient b, which is a proportional constant of the relationship between the shift in average delay and the product of the variance and the amount of rate change. Furthermore, it also is possible to adapt b during a session, by keeping evaluating the relationship and updating the proportional constant. Such dynamic derivation and adaptation of the coefficient b remain as future work. We also plan to conduct practical experiments on a real network. Considering that a considerable number of sessions and they are multiplexed, their aggregated behavior observed by a session would be averaged and show the Gaussian fluctuation. It favors the attractor perturbation model and our proposal is expected to be effective.

Acknowledgements

This thesis would not accomplish without a lot of great supports of several people. First, I would like to express my gratitude to my supervisor, Professor Masayuki Murata of Osaka University, for his continuous support and a lot of valuable comments on my study. Furthermore, I show my deepest appreciation to Professor Naoki Wakamiya of Osaka University. He devoted a great deal of time for me and gave me an excellent guideline of my research and considerable supports.

Also, I acknowledge Professors Koso Murakami, Teruo Higashino, and Hirotaka Nakano of Osaka University, for their appropriate suggestions on my study.

I ’m greatly indebted to Specially Appointed Associate Professor Kenji Leibnitz for his advices on my study and his care.

Moreover, I would like to appreciate to Associate Professor Shin’ichi Arakawa and Assistant Professor Yuichi Ohsita for valuable comments and suggestions on this study. In addition, I would like to express sincere appreciation to my senior associates, Mr. Narun Asvarujanon, Mr. Hiroshi Yamamoto, and Mr. Takuya Iwai. I received a lot of advices from them and they kindly provided consultation for me.

Finally, I would like to thank all the members of the Advanced Network Architecture Laboratory at the Graduate School of Information Science and Technology, Osaka University, for support and fruitful discussion about my research and hearty encouragement.

References

[1] L. Atzori and M. Lobina, “Playout buffering in IP telephony: a survey discussing problems and approaches,” *IEEE Communications Surveys Tutorials*, vol. 8, pp. 36–46, 3rd quarter 2006.

[2] K. Fujimoto, S. Ata, and M. Murata, “Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications,” *Telecommunication Systems*, vol. 25, pp. 259–271, Mar. 2004.

[3] S. Kadur, F. Golshani, and B. Millard, “Delay-jitter control in multimedia applications,” *Multimedia systems*, vol. 4, pp. 30–39, Feb. 1996.

[4] D. Verma, H. Zhang, and D. Ferrari, “Delay jitter control for real-time communication in a packet switching network,” in *Proceedings of IEEE TRICOMM’91*, pp. 35–43, Apr. 1991.

[5] Y. Mansour and B. Patt-Shamir, “Jitter control in QoS networks,” *IEEE/ACM Transactions on Networking*, vol. 9, pp. 492–502, Aug. 2001.

[6] J. Pedrasa and C. Festin, “Value-based utility for jitter management,” in *Proceedings of IEEE Region 10 Conference*, pp. 1–5, Nov. 2006.

[7] D. Hay and G. Scalosub, “Jitter regulation for multiple streams,” *IEEE/ACM Transactions on Algorithms*, vol. 6, pp. 1–19, Dec. 2009.

[8] T. Okuyama, K. Yasukawa, and K. Yamaoka, “Proposal of multipath routing method focusing on reducing delay jitter,” in *Proceedings of IEEE Pacific Rim Conference*, pp. 296–299, Aug. 2005.

[9] I. Busse, B. Deffner, and H. Schulzrinne, “Dynamic QoS control of multimedia applications based on RTP,” *Computer Communications*, vol. 19, no. 1, pp. 49–58, 1996.

[10] Y. Sun, F. Tsou, and M. Chen, “Predictive flow control for TCP-friendly end-to-end real-time video on the internet,” *Computer Communications*, vol. 25, pp. 1230–1242, Aug. 2002.

[11] K. Sato, Y. Ito, T. Yomo, and K. Kaneko, “On the relation between fluctuation and response in biological systems,” *National Academy of Sciences*, vol. 100, pp. 14086–14090, Nov. 2003.

[12] K. Leibnitz and M. Murata, "Attractor selection and perturbation for robust networks in fluctuating environments," *IEEE Network*, vol. 24, pp. 14–18, May 2010.

[13] A. Hernandez and E. Magafia, "One-way delay measurement and characterization," in *Proceedings of the Third International Conference on Networking and Services*, pp. 114–119, June 2007.

[14] "iperf." `http://sourceforge.net/projects/iperf`.

[15] K. Leibnitz, C. Furusawa, and M. Murata, "On attractor perturbation through system-inherent fluctuations and its," in *Proceedings of the 2009 International Symposium on Nonlinear Theory and its Applications*, pp. 22–25, Oct. 2009.

[16] N. Asvarujanon, N. Wakamiya, K. Leibnitz, and M. Murata, "Noise-assisted traffic distribution over multi-path ad hoc routing," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pp. 1–5, Oct. 2011.

[17] J. Shanthikumar, "On reducing time spent in M/G/1 systems," *European Journal of Operational Research*, vol. 9, pp. 286–294, 1982.

[18] "The network simulator ns-2." `http://www.isi.edu/nsnam/ns/`.

[19] D. Sisalem and A. Wolisz, "LDA+ TCP-friendly adaptation: A measurement and comparison study," in *Procceding of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 1619–1622, June 2000.

[20] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proceeding of aIEEE conference on Computer Communication*, pp. 1337–1345, Jan. 1999.