Hierarchical Performance Evaluation Method for Describing the Interactions between Networks and Users

Sota Hatakeyama[1], Chisa Takano[2], and Masaki Aida[1]
[1] Graduate School of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan
[2] Graduate School of Information Sciences, Hiroshima City University, Hiroshima 731-3194, Japan
Email: sota-hatakeyama@sd.tmu.ac.jp; takano@hiroshima-cu.ac.jp; aida@tmu.ac.jp

Abstract —Information networks are an important social infrastructure, and we should ensure their stable and sustainable operation. Since retry traffic greatly impacts network stability, we should, in particular, consider retry traffic when designing and controlling communication network systems. We previously proposed a hierarchical method that uses the quasi-static approach to evaluate system performance in the presence of retry traffic. This approach is based on the fact that system response times are much shorter than the users' perceivable time-scales. For the sake of simplicity, previous work considered a specific model that assumed that all users' behaviors were synchronized with respect to a certain time interval even though users behave independently. This paper introduces a performance evaluation method, based on the framework of the quasi-static approach, for networks in which retry traffic is generated by users autonomously.

Index Terms—Retry traffic, IP telephony, system stability, quasi-static approach

I. Introduction

Signaling systems play a crucial role in supporting all virtual channel (VC) communication services, including IP telephony and conventional telephone networks. The congestion in the current Internet is caused by overloading of not only the communications links but also the processing resources in the signaling systems. Recently, problems with commercial IP telephony systems have been reported in Japan. One of the key causes of such problems is overloading of the signaling functions provided by the call-processing system. In particular, the retry traffic generated by users has a serious negative impact on system stability. Here, retry means multiple user attempts to set up a connection.

In general, retries are generated by two different factors as described below.

- Retry traffic generated due to a shortage of link bandwidth.

This shortage (e.g., resource for VCs) triggers rejection of requests for connection setup or for bandwidth reservation. If a request is rejected, the corresponding user might generate retry traffic.

- Retry traffic generated due to a shortage of call-processing resources.

A shortage of processing resources causes the response time of the call-processing system to increase. Psychological factors will induce impatient users to retry their requests. Because prior requests are not cancelled, one user can create duplicate requests.

A variety of queuing models with retry traffic have been well studied [1], [2]. Conventionally, the relationship between the link bandwidth and the input rate of call setup requests has been investigated using the M/G/s/s model. Let us consider the M/G/s/s based retry model called the multi-server model in [1]. The expression M/G/s/s represents a model in which service (call-setup) requests arise in a Poisson manner, enter the system, receive service from one of s servers, and then leave the system, and in which service requests are discarded if all s servers are busy. An M/G/s/s retrial queue model represents an M/G/s/s model in which discarded service requests are stored in a retry queue and re-enter the system after a certain elapsed time determined by an exponential distribution. This model is stable if the length of the retry queue does not diverge [2]; its stability condition is known to be

$$\frac{\lambda_0}{\eta} < s. \tag{1}$$

where λ_0 is the arrival rate of primary service requests, excluding retry requests, per unit time, and $1/\eta$ is the average service time. The above model incorporates retry traffic that arises due to a resource shortage in the link bandwidth, but it does not consider retry traffic that arises due to resource shortages in the call-processing system. However, such traffic is expected to serious impact system stability. This is because that the prior requests are not cancelled when impatient users generate retry traffic, duplicate requests will exist in the system. In this paper, we focus on a call-processing system that is configured using a VC-based communication service model (e.g., IP telephony), and consider the stability of the system under the impact of retries. We consider how to evaluate the probability that the system will become unstable.

Our previous research focused on an IP telephony system, and discussed the properties of retry traffic

Manuscript received July 24, 2014; revised October 16, 2014.
This work was supported by a Grant-in-Aid for Scientific Research (B) No. 26280032 (2014-2016) from the Japan Society for the Promotion of Science.
Corresponding author email: aida@tmu.ac.jp
doi:10.12720/jcm.9.10.737-744

caused by resource shortages in the link bandwidth and/or the call-processing system [3]–[6]. To understand the dynamics of retries, we should understand the interactions of the users and the system. A fundamental approach for describing interactions is the decomposition of timescales. However, the decomposition models in earlier papers were primitive. In [3], we introduced the quasi-static approach to describe fluctuations in traffic. This approach assumes that state transitions of the system occur on a timescale significantly (infinitesimally) shorter than the timescale of human perception. This means that the system might as well be working at infinite speed. This assumption cannot take account of fluctuations in traffic. In [4]–[6], it is assumed that all users' behaviors are synchronized with respect to a certain time interval, but this assumption is unrealistic.

This paper focuses on retry traffic caused by resource shortages in the call-processing system and introduces a retry traffic model in which users' behaviors are mutually independent. That is, we develop a new realistic model of retry traffic, and clarify that system stability can be evaluated by using the conventional quasi-static approach. First, we briefly explain time scale decomposition for describing the retry traffic in Sec. II. Next, we show the conventional quasi-static approach as the conventional retry traffic model and its issues, in Sec. III. Then, we introduce a new retry traffic model that can describe users' behaviors properly as an extension of the quasi-static approach in Sec. IV. The validity of the proposed model is verified through an evaluation of system stability in Sec. V. Finally, we conclude our discussion in Sec. VI.

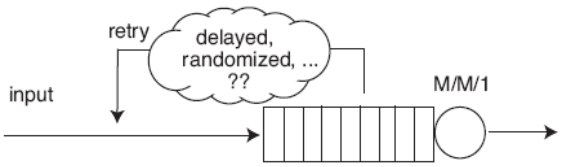


Fig. 1. System model

II. Time Scale Decomposition for Describing Retry Traffic

This section describes time scale decomposition that can describe a retry traffic model that can take account of resource shortages in call-processing systems. The decomposition gives a foundation to the quasi-static approach shown in the next section. The significant features of the decomposition include macroscopic system behavior on a human perceptible timescale.

The call-processing system model should describe the system behavior related to call setup at a server, and for this we use the M/M/1-based model. The M/M/1-based model is a combination of a simple M/M/1 model and a retry traffic model that includes resource shortages in the call-processing system, with retry traffic generated by impatient users (Fig. 1). That is, some of the users who have been kept waiting for execution of call setup request a new call setup. Such retry traffic will be generated without canceling the existing call setup requests.

Let us consider how retry traffic arises. It is natural to assume that the intensity of retry traffic depends on the length of the queue in the M/M/1 model. It is logical to consider that retry traffic dependent on the number of queued contents in the call-processing system. If we assume that each call setup request waiting in the call-processing system generates retry traffic at a certain rate ε, then a diagram depicting the state transition rate with respect to the number of requests in the M/M/1 model will be like that shown in Fig. 2. In this figure, λ_0 is the primary arrival rate of call setup requests, excluding retry traffic, per unit time, while $1/\mu$ is the average service time of the call-processing system. For this system to have a steady-state probability, the infinite sum on the right-hand side of the following equation must exist.

$$p_0 = \left[1 + \sum_{i=1}^{\infty} \prod_{j=0}^{i-1} \left(\frac{\lambda_0 + j\epsilon}{\mu}\right)\right]^{-1}$$

Therefore, if $\varepsilon > 0$, the system is unstable. Since an increase in retry traffic does not result in the divergence of the waiting time of an actual call-processing system under normal operating conditions, we can conclude that a model in which retry traffic is generated in proportion to the number of call setup requests in the call-processing system is not realistic.

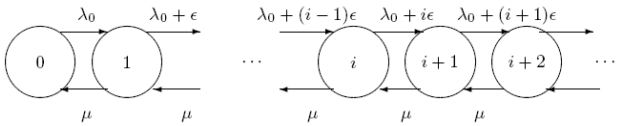


Fig. 2. State transition diagram for the case where retry traffic is generated in proportion to the current number of call setup requests in the call-processing system

Let us identify what is inappropriate in the above model. In general, state transitions of the call-processing system occur on a timescale much shorter than humans can perceive. It is natural to assume that the intensity of the retry traffic is not proportional to the number of call setup requests in M/M/1 at the present time, but proportional to the average number of call setup requests in M/M/1 on a longer timescale. The timescale should be long enough for humans to perceive. Let *T* be the minimum timescale perceptible to humans. We assume that the call-processing system is in a steady state on a timescale smaller than *T*, and that retry traffic affects the system on a timescale larger than *T*. In the following, we summarize the assumptions of such a quasi-static state for the generation of retry traffic.

- The system can be assumed to be in a steady state on a timescale smaller than *T*.
- Changes in the system are observed at discrete times occurring at an interval of *T*.
- Retry traffic generated from users at a certain discrete time, $t = k$, is determined by the steady-state probability of the system at $t = k - 1$.

Note that the condition that the steady state is achieved on a finite timescale T implies that transitions of the system occur on a timescale much (infinitesimally) shorter than the human perceptible timescale.

In other words, the system works at essentially infinite speed.

The value of T must satisfy the following requirements:

- T must be a timescale on which humans can actually perceive an increase in the waiting time for their call requests.
- T must be sufficiently longer than the timescale for which it is possible to assume that the system is in a steady state on a timescale smaller than T.

It is well known that the tolerable waiting time to display a website is about 8 seconds; a longer wait exceeds the typical user's patience Many webpages are designed so as to satisfy this so-called 8-second rule [7]. Measurements of waiting time have verified that a user's interest fades after 10 seconds or so [8]. In our case, however, the tolerable waiting time must be smaller than that for webpages. Reference [9] classifies the key time intervals according to human perception as follows:

- Delay of 0.1 second is perceived as instantaneous access.
- Delay of 1.0 second is the limit for a user's thought flow to remain uninterrupted.
- Delay of 10 seconds is the limit for keeping a user's attention/focus on the dialogue.

The last category coincides fairly well with the 8-second rule for websites in terms of both the duration and the explanation. Since the generation of retry traffic seems to correspond to the second category, one second is a reasonable value for T.

Changes in the state of the system are examined at discrete intervals T. The retry traffic at time $t = k$ is determined by the steady-state probability of the system at time $t = k - 1$ (see Fig. 3).

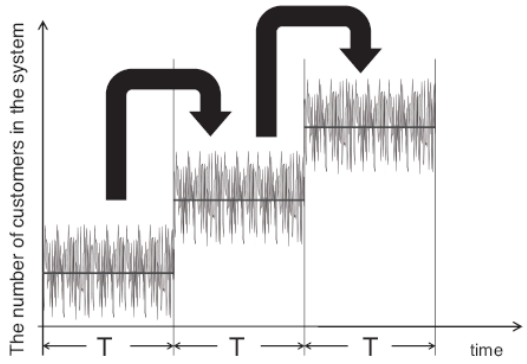


Fig. 3. Temporal evolution of the input rate in discrete interval model.

Let λk be the input rate, including retry traffic, at time k. We assume that the input rate at time $k + 1$ is

$$\lambda_{k+1} = \begin{cases} \lambda_0 + \epsilon \dfrac{\lambda_k/\mu}{1-\lambda_k/\mu}, & (\lambda_k/\mu < 1) \\ \infty, & (\lambda_k/\mu \geq 1) \end{cases} \quad (2)$$

where $1/\mu$ is the average service time of the call-processing server, and ε is a positive constant indicating the intensity of the retry traffic generated due to a call-processing resource shortage.

Next, let us consider the stability of the system. We assume that the requirement for the system to be stable is that traffic, including retry traffic, does not diverge after sufficient elapsed time. Namely,

$$\lim_{k\to\infty} \lambda_k < \infty \quad (3)$$

This can be verified by defining the functions of λ, $f(\lambda)$ and $g(\lambda)$ as follows, and determining if they intersect.

$$f(\lambda) = \lambda_0 + \epsilon \frac{\lambda/\mu}{1-\lambda/\mu} \quad (4)$$

$$g(\lambda) = \lambda \quad (5)$$

Since $\lambda_0 > 0$, $f(0) > g(0)$. The relationship between $f(\lambda)$ and $g(\lambda)$ in some typical cases is shown in Fig. 4. If the input traffic at a certain time is λ, the input traffic at the next time instant becomes $\{g^{-1} \circ f\}(\lambda)$, followed by $\{g^{-1} \circ f\}^2(\lambda)$, etc. In general, the input traffic after the passage of n unit times becomes $\{g^{-1} \circ f\}^k(\lambda)$. As shown in the left chart in Fig. 4, if $f(\lambda)$ and $g(\lambda)$ do not intersect, $\{g^{-1} \circ f\}^k(\lambda) \to \infty$ $(k \to \infty)$. If, on the other hand, the two intersect in the manner shown in the right chart of Fig. 4, then the system is stable for all λ to the left of the right-most intersection.

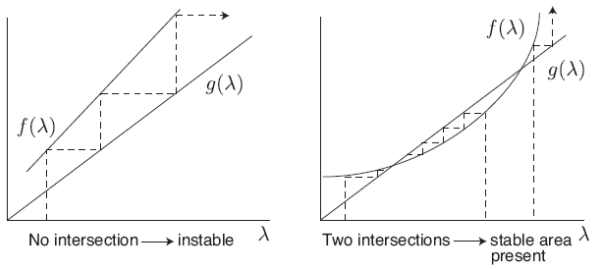


Fig. 4. System stability possible if $f(\lambda)$ and $g(\lambda)$ intersect.

III. Quasi-Static Approach: Concept and Issues

In the previous sections, we assumed that a steady state was achieved on finite timescale T. This implied that the system works at an infinitely high speed. However, since real systems have finite speeds, the steady state is never achieved in a finite time interval. In addition, synchronization of users' behaviors with respect to interval T, shown in Fig. 3, is unrealistic. The quasi-static approach tries to avoid these problems.

A. Concept of Quasi-Static Approach

In this subsection, we briefly explain the concept of the quasi-static approach. Let us consider how to examine system stability to input traffic under the condition that the system works at a finite speed. First, we clarify certain problems in the traditional approaches to examining stability. Then, we clarify the concept of the quasi-static approach can avoid the problems.

Let n be the number of call setup requests arising in interval T. If retry traffic from the call-processing system is proportional to the average number of call requests waiting in M/M/1 at time k (in interval T), the second term on the right-hand side of (2) is replaced with

$$\epsilon \frac{1}{n} \sum_{i=1}^{n} Q_i^k \tag{6}$$

where Q_i^k denotes the number of call requests waiting in the M/M/1 system immediately before the i-th request arrives. If the system works at infinite speed, that is, $n \to \infty$, we have

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Q_i^k = \epsilon \frac{\lambda_k/\mu}{1-\lambda_k/\mu} \quad \text{a.s.} \tag{7}$$

Fig. 5 shows the appropriate approaches with respect to the system speed. The horizontal axis plots the system speed from slow to infinitely high. Let us consider the case where the system works at a very low speed, for example $n = 1$. This case corresponds to the situation in which a human can detect the systems present state, meaning that the human's perception is exceptionally sensitive or the system speed is very slow. In this case, the next value of input traffic depends only on the latest value of Q_1^k. That is, the system can be described by the Markov model of Fig. 2, which corresponds to *slow* in Fig. 5.

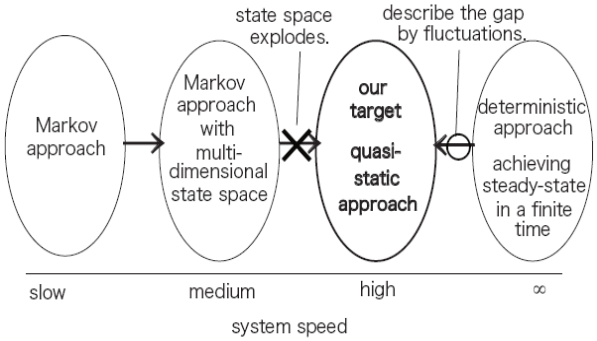


Fig. 5. Concept of quasi-static approach.

In general, if $n > 1$, we can use a Markov model with an n-dimensional state space that is constructed by the past n states $\{ Q_1^k, Q_2^k, ..., Q_n^k \}$. That is, it corresponds to medium in Fig. 5. However, if the system works at high speed, $n \gg 1$, the state space explodes and becomes intractable.

Next, we consider the applicability of simulation techniques. If we require the probability of the system becoming unstable, for example, to be less than 10^{-6} in the long-term operation of an IP telephony system, more than 10^6 (usually 10^8 to 10^9) runs of the corresponding long-term simulation are required. Such large-scale simulations are not realistic.

The quasi-static approach proceeds follows. In order to analyze the stability of the high (but finite) speed system, the quasi-static approach first examines the sys- tem behavior at infinite speed, and considers the gap between that behavior and the system behavior at finite speed as fluctuations (see Fig. 5). In other words, the change factor in the state of the system is decomposed into two parts, one being the deterministic change factor which is described by system behavior at infinite system speed and the other factors being described as stochastic fluctuations. This approach is effective for a model with a scale that cannot be realistically handled by both the Markov model and the simulation technique.

B. Conventional Quasi-Static Approach

We define input rate $\Lambda(t;T)$ at time t as

$$\Lambda(t;T) = \lambda_0 + \epsilon \langle Q_T \rangle_t \tag{8}$$

where $<Q_T>_t$ is a kind of the average number of call requests in the call-processing M/M/1 system during interval $[t - T, t]$ and ε is a positive constant. This is an extension of (2) written using continuous time t. The input rate $\Lambda(t;T)$ is given by the sum of the rate λ_0 for primary traffic and the rate for retry traffic, which is proportional to the average number of call requests.

In the conventional quasi-static approach, $<Q_T>_t$ is defined as the average during interval T immediately before the present time t, that is,

$$\Lambda(t;T) = \lambda_0 + \frac{\epsilon}{T} \int_{t-T}^{t^-} Q(s)\mathrm{d}s \tag{9}$$

where $Q(t)$ is the number of call requests in the system at time t. This model makes the rate of retry traffic rate at time t proportional to the average number of call-setup requests in $[t - T, t]$. If users were able to react to the variation of the system state immediately, $T \to 0$ then

$$\lim_{T\to+0} \Lambda(t;T) = \lambda_0 + \epsilon Q(t^-) \tag{10}$$

This corresponds to the system model described by the state transition diagram shown in Fig. 2.

As a preliminary for describing the quasi-static approach, we show the Langevin equation and the Fokker- Planck equation [10]. Let $X(t)$ be a random variable. Let us investigate the temporal evolution of $X(t)$ as follows. We assume that the temporal evolution of $X(t)$ is given by

$$\mathrm{d}X(t) = g_1(X)\,\mathrm{d}t + g_2(X)\,\mathrm{d}W(t) \tag{11}$$

where $g_1(X)$ and $g_2(X)$ are functions of $X(t)$, and $W(t)$ is the Wiener process. The first and second terms on the right-hand side of (11) are deterministic and stochastic parts of the temporal evolution, respectively. Equivalently, this equation can be written in the following form

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = g_1(X) + g_2(X)\,\xi(t) \tag{12}$$

where $\xi(t)$ is Gaussian white noise such that $E[\xi(t)] = 0$

and E $[\xi(t)\xi(t')] = \delta(t - t')$. This equation is called the Langevin equation. Let the probability density function of X (t) be p(x, t). Then, the temporal evolution function of p(x, t) is given by the following Fokker-Planck equation,

$$\frac{\partial}{\partial t}p(x,t) = \left(\frac{\partial}{\partial x}g_1(x) + \frac{\partial^2}{\partial x^2}g_2(x)\right)p(x,t) \tag{13}$$

The first and second terms of (13) are called the drift term and the diffusion term, respectively. The drift term describes the deterministic motion of $p(x,t)$ and the diffusion term describes the fluctuation around the drift motion. The values of $g_1(X)$ and $g_2(X)$ govern the strength of drift and diffusion motions, and they denotes variation of the mean value and its standard deviation, respectively. These mechanisms are easily recognized through the following explanation. Let us introduce potential function $U(x)$ of the drift motion as

$$U(x) = -\int g_1(x)\,\mathrm{d}x \tag{14}$$

Fig. 6 shows an example of the potential function. The drift motion occurs toward the direction that the value of the potential decreases. The minimum point and the wall of the potential correspond to the left and right intersections in the right panel of Fig. 4, respectively. In addition to the drift motion described as Fig. 6, the quasi-static approach can describe some fluctuations by the diffusion term.

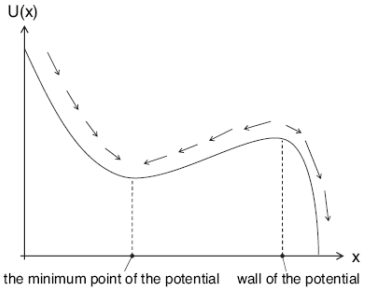


Fig. 6. Example of potential function and the corresponding drift motion.

Next, we consider the temporal evolution of retry traffic by the quasi-static approach. We define the actual number of customers arriving during $[t - T, t]$ as $X(t,T)$. Under the limit of higher system speed, $\lim_{T\to\infty} X(t,T)/T = \Lambda(t;T)$ a.s. In general, $X(t,T)$ is not stationary but the variation occurs very slowly. Here, we define the infinitesimal variation of $X(t,T)$ as

$$\begin{aligned}\mathrm{d}X(t,T) &= X(t+\mathrm{d}t,T) - X(t,T)\\ &= X(t+\mathrm{d}t,\mathrm{d}t) - X(t-T,\mathrm{d}t)\end{aligned} \tag{15}$$

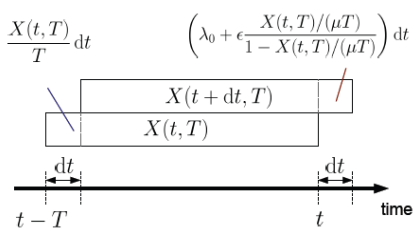


Fig. 7. Variation of input rate

This is composed of the difference between increment $X(t+\mathrm{d}t, \mathrm{d}t)$ and decrement $X(t-T, \mathrm{d}t)$ (see Fig. 7).

Here, since the input traffic follows a Poisson process, the variance of the input traffic is equal to the input rate. In addition, since we consider large-scale systems many call requests, the Poisson distribution can be approximated by a Gauss distribution with no correlation. Therefore, by the quasi-static approach, the number of arriving customers can be expressed as

$$\begin{aligned}&X(t+\mathrm{d}t,\mathrm{d}t)\\ &\quad= \left(\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)}\right)\mathrm{d}t\\ &\qquad+\sqrt{\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)}}\,\mathrm{d}W(t)\end{aligned} \tag{16}$$

$$\begin{aligned}&X(t-T,\mathrm{d}t)\\ &\quad= \frac{X(t,T)}{T}\mathrm{d}t + \sqrt{\frac{X(t,T)}{T}}\,\mathrm{d}W(t)\end{aligned} \tag{17}$$

In the form of Langevin equation, we have

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}X(t,T)\\ &\quad= \left(\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} - \frac{X(t,T)}{T}\right)\mathrm{d}t\\ &\qquad+\sqrt{\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} + \frac{X(t,T)}{T}}\,\xi(t)\end{aligned} \tag{18}$$

This equation has been verified by comparison against simulation results in a certain situation that all the users are synchronized as described in Fig. 3 [6]. The temporal evolution equation (18) has been improved from the deterministic model (2) in following two ways.

- (18) is stochastic model and it takes fluctuations into consideration.
- The time parameter is improved from discrete variable k to continuous variable t.

However, synchronization of users' behaviors is not completely avoided. The users' behaviors in the time interval of T affect only the next time interval. It is natural that the behavior of individual users should be independent of the specific periodic timing of T introduced by us. To avoid this problem, we introduce an extension of the conventional quasi-static approach in the next section.

IV. Extension of Quasi-Static Approach

A. Randomness of Retry Users

In order to avoid the fact that an individual user depends on a specific periodic interval of T, we assume that the call-setup requests (customers) waiting for service by a M/M/1 call-processing system generate retry traffic after randomized intervals. Since we assume the input traffic obeys a Poisson process, the randomized interval follows an exponential distribution.

Each customer in the system at an arbitrary point of time retries the request at a constant rate ε per unit time. The randomized time interval, which is from the arbitrary

chosen time point to the time of the corresponding retry traffic arrival, obeys an exponential distribution with mean of T. Fig. 8 illustrates the above retry traffic model. Let $Q(s)$ be the number of customers in the system at time s. The contribution to the input rate of retry traffic at current time t ($s < t$) from $Q(s)$ at the point of the past s is $(\varepsilon Q(s)/T)\exp((t-s)/T)\mathrm{d}t$. Then input rate at time t is expressed as

$$\Lambda(t;T) = \lambda_0 + \frac{\epsilon}{T}\int_{-\infty}^{t^-} Q(s)\, e^{-\frac{1}{T}(t-s)}\,\mathrm{d}s. \tag{19}$$

This is an example of (8) and an extension of (9) [11]. The second term on the right hand side of (19) denotes retry traffic rate. This is the exponentially weighted moving average of the number of customers of the past. Figure 9 shows examples of the temporal evolution of customers and the weight. The quantity of this weights decays exponentially by time constant T. Similar to (10), if users were able to react to variation of the system state immediately, $T \to 0$ then

$$\lim_{T\to+0} \Lambda(t;T) = \lambda_0 + \epsilon\, Q(t^-) \tag{20}$$

This means the new model (19) is also an extension of the Markov model described in Fig. 2.

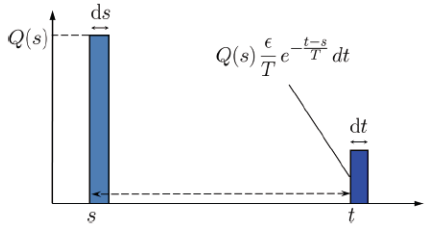


Fig. 8. Relationship between the number of requests $Q(s)$ at s and the retry at t caused by $Q(s)$

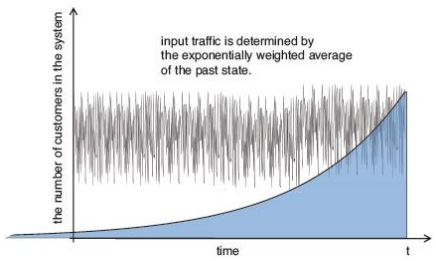


Fig. 9. Weighted average.

B. Temporal Evolution Equation for the Extended Quasi-Static Approach

The extended input traffic model (19) derived in the previous subsection fundamentally differs from the conventional model (9). In the extended model of (19), individual users autonomously generate retry traffic where generation is independent of any specific interval, T. In this subsection, we consider the temporal evolution equation of the input traffic based on the extended model (19).

In order to measure the rate of input traffic actually generated, we need to count the input traffic generated during a certain time interval and derive the number of input traffic per unit time. That is, it is impossible to know the input traffic rate of (19) by measurement at any one moment. Let $X(t,T)$ be the number of actual input traffic generated in time period $[t - T, t]$ whose length is T. If $T \to \infty$, the observed input traffic rate $X(t)/T$ approaches the theoretical value,

$$\frac{X(t)}{T} = \lambda_0 + \frac{\epsilon}{T}\int_{-\infty}^{t^-} Q(s)\, e^{-\frac{1}{T}(t-s)}\,\mathrm{d}s \quad \text{a.s.} \tag{21}$$

However, as seen in Fig. 5, (21) is not correct for a finite T and we should additionally consider fluctuations.

The infinitesimal variation of $X(t,T)$ is expressed as

$$\begin{aligned}\mathrm{d}X(t,T) &= X(t+\mathrm{d}t,T) - X(t,T)\\ &\overset{\text{def}}{=} A(t;\mathrm{d}t) - D(t;\mathrm{d}t)\end{aligned} \tag{22}$$

After the infinitesimal time $\mathrm{d}t$ has elapsed, X is increased by $A(t;\mathrm{d}t)$ and is decreased by $D(t;\mathrm{d}t)$. Because tran- sitions in system state are attributed to user behavior, stochastic process X varies slowly at rate T. In addition, since we consider large-scale system ($\lambda_0 \gg 1$) and there are a lot of call requests, the input Poisson process can be approximated by a Gauss distribution with no correlation. Therefore, the infinitesimal variations A and D are expressed as

$$\begin{aligned}A(t;\mathrm{d}t) = &\left(\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)}\right)\mathrm{d}t\\ &+\sqrt{\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)}}\,\mathrm{d}W(t)\end{aligned} \tag{23}$$

$$D(t;\mathrm{d}t) = \frac{X(t,T)}{T}\mathrm{d}t + \sqrt{\frac{X(t,T)}{T}}\mathrm{d}W(t) \tag{24}$$

Thus, we can obtain

$$\begin{aligned}&\mathrm{d}X(t,T)\\ &= \left(\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} - \frac{X(t,T)}{T}\right)\mathrm{d}t\\ &\quad + \sqrt{\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} + \frac{X(t,T)}{T}}\,\mathrm{d}W(t)\end{aligned} \tag{25}$$

In the form of Langevin equation, (25) is expressed as

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}X(t,T)\\ &= \left(\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} - \frac{X(t,T)}{T}\right)\\ &\quad + \sqrt{\lambda_0 + \epsilon\frac{X(t,T)/(\mu T)}{1-X(t,T)/(\mu T)} + \frac{X(t,T)}{T}}\,\xi(t)\end{aligned} \tag{26}$$

and the result, in the form of the Fokker-Planck equation, is

$$\begin{aligned}&\frac{\partial}{\partial t}p(x,t)\\ &= \frac{\partial}{\partial x}\left(\lambda_0 + \epsilon\frac{x/(\mu T)}{1-x/(\mu T)} - \frac{x}{T}\right)p(x,t)\\ &\quad + \frac{\partial^2}{\partial x^2}\sqrt{\lambda_0 + \epsilon\frac{x/(\mu T)}{1-x/(\mu T)} + \frac{x}{T}}\,p(x,t)\end{aligned} \tag{27}$$

The above results are the same as that of the conventional quasi-static approach (18).

Note that these input models are very different although their forms of temporal evolution are the same. The reason why we get the same equations is that the procedure of measuring the input rate is the same as the procedure of moving average of the input rate. Fortunately, we can use the same temporal evolution equation even though the input traffic model becomes complex.

V. Simulation Results

This subsection verifies the validity of (26) and (27) by comparison with Monte Carlo simulation. In this simulation model, each user generates retry traffic independently of the specific periodic interval of T. This is a major difference from the evaluation of the conventional quasi- static approach shown in [6].

Fig. 10 shows the potential function with parameters of $\lambda_0 = 800$, $\mu = 1000$, $\varepsilon = 0.5$, $T = 1.0$. The horizontal axis denotes $X(t,T)$ and vertical axis denotes the potential function $U(X)$. As the initial condition, the $p(x,0)$ is a δ-function at the minimal point of the potential function. The wall of the potential function is an absorption state (i.e., once it reaches there, it can never re- turn). Fig. 11 and Fig. 12 show the cumulative distribution function of $X(t, T)$ at $t = 1$ and $t = 50$, respectively. We can recognize that the quasi-static approach can properly describe the behavior of input traffic.

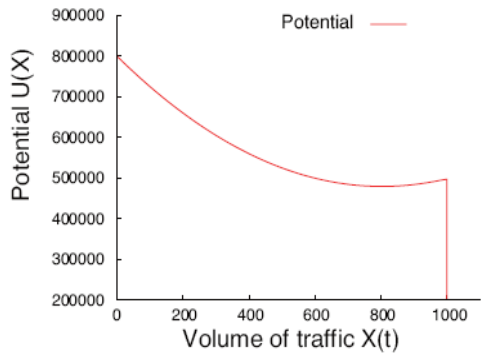


Fig. 10. Potential function.

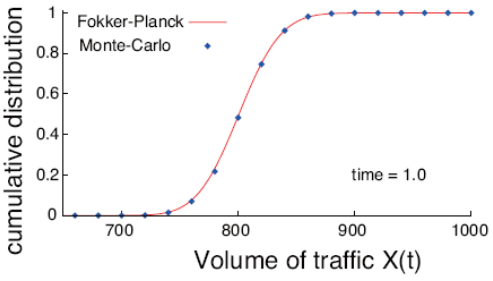


Fig. 11. Distribution of the input traffic at $t = 1.0$

Next, we investigate the situation in which the initial condition is not at the minimum point. As the initial condition, we set the delta-function to a position shifted from the minimum point of the potential function. The parameters are as follows: $\lambda_0 = 500$, $\mu = 1000$, $\varepsilon = 1.0$, $T = 1.0$. Fig. 10 shows the potential function corresponding to the above condition. The minimal point of the potential function exists at $X \simeq 500$. We set the initial distribution $p(x,0)$ as the δ-function at $X = 600$. Figures 14 and 15 show the cumulative distribution function of $X(t,T)$ at $t = 1$ and $t = 5$, respectively. We can recognize that the quasi-static approach can properly describe the change of the distribution. Through the above experiments, we can confirm that the quasi-static approach can describe the behavior of the system deviated from the stable point by perturbation.

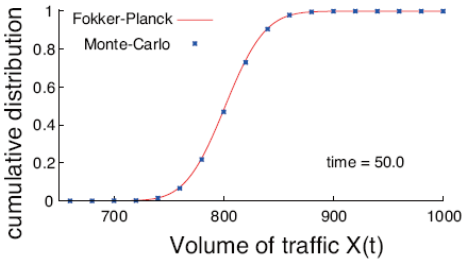


Fig. 12. Distribution of the input traffic at $t = 50.0$

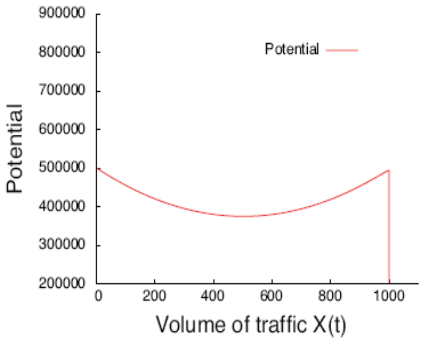


Fig. 13. Potential function

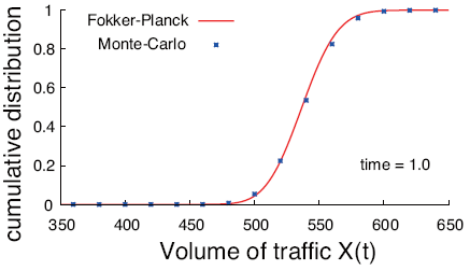


Fig. 14. Distribution of the input traffic at $t = 1.0$

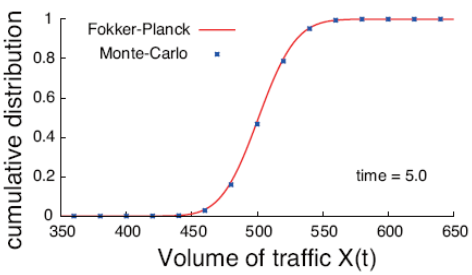


Fig. 15. Distribution of the input traffic at $t = 5.0$

VI. Conclusions

In this paper, we have shown how to properly account for retry traffic generated by interaction between users and networks. To model the interaction properly, our approach describes the retry traffic randomly generated by individual users. The model can characterize the natural behavior of users unlike the conventional model which uses a specific periodic interval.

Fortunately, although our proposed model is more complex than the conventional model, the temporal evolution equation of the input traffic can be written in the same form. Therefore, our proposed model does not need to introduce additional complex procedures. Experiments support the fact that our proposed model properly de- scribes the behavior of retry traffic.

This paper focused on retry traffic generated by inadequate resources in the call-processing system. Since retry probably significantly impacts various communication services, we need to investigate retry traffic models for the various services. In these processes, we expect that the concept of time scale decomposition and the quasi- static approach will prove to be useful tools.

ACKNOWLEDGMENT

This work was supported by a Grant-in-Aid for Scientific Research (B) No. 26280032 (2014–2016) from the Japan Society for the Promotion of Science.

REFERENCES

[1] J. R. Artalejo, "Accessible bibliography on retrial queues," *Mathematical and Computer Modeling*, vol. 30, pp. 1–4, 1999.

[2] G. I. Falin and J. G. C. Templeton, *Retrial Queues*, Chapman & Hall, London, 1997.

[3] M. Aida, C. Takano, M. Murata, and M. Imase, "A study of control plane stability with retry traffic: Comparison of hard- and soft-state protocols," *IEICE Transactions on Communications*, vol. E91-B, no. 2, pp. 437–445, 2008.

[4] M. Aida, C. Takano, M. Murata, and M. Imase, "A pro- posal of quasi-static approach for analyzing the stability of IP telephony system," in *Proc. International Conference on Networking*, 2008, pp. 363–370.

[5] M. Aida, C. Takano, M. Murata, and M. Imase, "Quasi- static approach for analyzing interactions between net- works and users based on decomposition of timescales," in *Proc. 3rd International Symposium on Multidisciplinary Emerging Networks and Systems*, 2012.

[6] K. Watabe and M. Aida, "Modeling the fluctuations in quasi-static approach describing the temporal evolution of retry traffic," *WSAES Transactions on Communications*, vol. 12, no. 9, pp. 488–498, 2013.

[7] Z. Re-search, "The need for speed II," *Zona Market Bulletin*, no. 5, pp. 1–9, April 2001.

[8] Fiona Fui-Hoon Nah, "A study on tolerable waiting time: How long are Web users willing to wait?" *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–163, 2004.

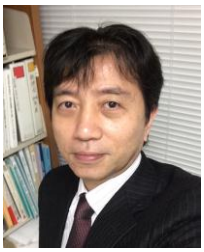
[9] J. Nielsen, "Response times: The three important limits," *Usability Engineering*, J. Nielsen, Ed., ch.5, Academic Press, 1993.

[10] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, Elsevier, 1992.

[11] M. Aida, "Using a renormalization group to create ideal hierarchical network architecture with time scale depen- dency," *IEICE Transactions on Communications*, vol. E95-B, no. 5, pp. 1488–1500, 2012.

Sota Hatakeyama received the B.E. degree from Tokyo Metropolitan University, Japan, in 2011. He is currently a graduate student working toward the M.E. degree at the Graduate School of System Design, Tokyo Metropolitan University. His primary interests include ad hoc networks.

Chisa Takano received the B.E. degree in Telecommunication Engineering from Osaka University, Japan, in 2000, and received the Ph.D. in Telecommuni-cations Engineering from Tokyo Metropolitan University, Japan, in 2008. In 2000, she joined the Traffic Research Center, NTT Advanced Technology Corporation (NTT-AT). Since April 2008, she has been an Associate Professor of the Graduate School of Information Sciences, Hiroshima City University. Her research interests are in the area of computer networks and distributed systems. She received the IEICE's Young Researchers' Award in 2003.



Masaki Aida received his B.S. degree in Physics and M.S. degree in Atomic Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, respectively, and the Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. In April 1989, he joined NTT Laboratories. From April 2005 to March 2007, he was an Associate Professor at the Faculty of System Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of System Design, Tokyo Metropolitan University since April 2007. His current interests include traffic issues in computer communication networks. He received the Best Tutorial Paper Award of IEICE Communications Society in 2013. He is a member of the IEEE, the IEICE, and the Operations Research Society of Japan.