Quasi-Static Approach for Retry Traffic with Different Service Time

Yoshiyuki ISHII, Kohei WATABE and Masaki AIDA
Graduate School of System Design, Tokyo Metropolitan University
6–6, Asahigaoka, Hino-shi, Tokyo 191-0065, Japan
Email: {ishii-yoshiyuki,watabe-kouhei,maida}@sd.tmu.ac.jp

***Abstract*—We proposed the "Quasi-static approach" as a method that can analyze the stability of a telecommunication system experiencing retry traffic. This method considers human interaction with the system, and use the difference of timescale between humans and system. Our recent studies evaluated the stability when all traffic including retry traffic has the same holding-time distribution as calls in the data plane. However, it is not natural to assume that all the calls including retries have the same holding-time distribution as those in the data plane. In this report, we consider a generalization of the quasi-static approach for retry traffic with different holding-time distributions.**

I. Introduction

In the Internet, congestion due to overload of the control plane has become an important issue in addition to congestion in the data plane. Problems with commercial IP telephony systems have recently been reported one after the other. One of the important causes is congestion in the control plane. In particular, retry traffic by the user greatly influences the performance of the control plane. Therefore, it is important to develop a mechanism to avoid congestion caused by retry traffic.

Retry traffic is generated by congestions in both the data plane and the control plane as described below.

- Retry traffic generated due to a shortage of data plane resources. A shortage of resources in the data plane causes requests to establish a connection or to secure bandwidth to be discarded. This will induce retries.
- Retry traffic generated due to a shortage of control plane resources. A shortage of processing resources causes the response time of the processing system to increase. This will induce impatient users to repeat their requests.

Since retry traffic itself consumes network resources, it is not appropriate to handle congestion on the data plane and that on the control plane separately. Therefore, it is important to manage network resources appropriately in order to prevent retry traffic from increasing. We proposed the "Quasi-static approach" as a method that can analyze the stability of a telecommunication system experiencing retry traffic [1].

Our previous studies assumed that all traffic, including retry traffic, had the same holding-time distribution of calls. However, this assumption is not realistic since retry traffic generated by a shortage of control plane resources overlap since calls that have been already input are not immediately cancelled. Therefore, it is necessary to assume that retry traffic generated due to a shortage of control plane resources has a different holding-time distribution. In this report, we consider a generalization of the quasi-static approach to handle retry traffic with different holding-time distributions.

This paper is organized as follows. Section 2 introduces the original quasi-static approach. In Sect.3, we consider the realistic behavior of retry traffic in data plane system. In addition, we discuss a generalization of the quasi-static approach to handle retry traffic with different holding-time distributions. Sect.4 evaluates the generalized quasi-static approach. Finaly, Section 5 shows our conclusions.

II. Outline of quasi-static approach

A. Basic model

In order to describe the retry traffic generated by resource shortages in the data and control planes, we have developed a model that incorporates a serial combination of a data plane model and a control plane model (Fig.1).

The data plane model describes the behavior of data transmission processing, such as securing link bandwidth; for this we use the M/G/s/s retrial queue model. If the system fails to secure bandwidth due to resource shortage in the data plane, it means that all s servers are busy. In this case a service request is kept waiting in the retrial queue for a period of time (exponential), and then is re-entered into the system as a new request.

The control plane model describes the system behavior related to routing and protocol processing; for this we adopt the M/M/1 model. M/M/1 treats the service requests as being generated in a Poisson manner, entering the system, receiving service from one server with exponential service time, and then leaving the system. Service requests are queued if the server is busy. The reasons why we apply M/M/1 to the control plane model are that it makes the analysis of the stability of the retry system more simple and tractable. Note that the use of a single server is an acceptable assumption. Although the accuracy of the M/M/1-based model with respect to the service time of the control plane model is unknown, the results obtained from this model will provide a foundation for analyzing more complex models in future studies.

In addition to the ordinary inputs of the M/M/1 model, some of the users who have been kept waiting due to increased processing time may attempt to retry (reenter) their request. Therefore, we have modified this model slightly to take

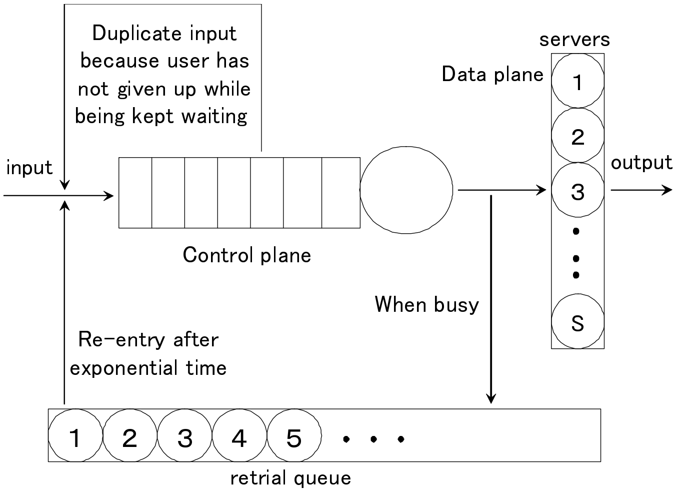

Fig. 1. Model incorporating retry traffic in both the control plane model (M/M/1) and the data plane model (M/G/s/s)

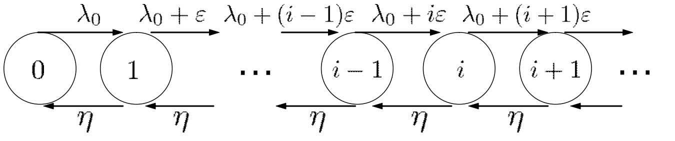

Fig. 2. State transition diagram: retry traffic generation is proportional to the queue length of the control plane system

account of retry traffic that will arise depending on the length of the queue in the M/M/1 model. Since such retry traffic will be generated without cancellation of the original service requests, our model assumes that new service requests arise without any of the service requests waiting in the queue in the M/M/1 model being withdrawn.

Note that we assume the input to the data plane follows a Poisson process. If there is no retry traffic, this assumption is true because M/M/1 is chosen for the control plane model. This is because service requests that leave the control plane and enter the data plane follow a Poisson process. As shown in subsequent sections, we also model the input of retry traffic as a Poisson process. Thus, the input to the data plane follows a Poisson process even when there is retry traffic.

B. Quasi-static retry traffic model

This subsection considers how retry traffic arises from the control plane system, see Fig.1. If the retry traffic is dependent on the length of the queue in the control plane, one possibility is that retry traffic volume is proportional to the length of the queue in the control plane. If we assume that service requests waiting in the queue generate retry traffic at a certain rate ε, the state transition rate with respect to the queue length of the control plane system is as shown in Fig.2. In this figure, λ_0 is the arrival rate of service requests, excluding retry traffic, per unit time, while $1/\eta$ is the average service time of the control plane system. For this system to have a steady-state probability, the infinite sum on the right-hand side of the following equation must be finite.

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

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 control plane under normal operating conditions, we can conclude that models that make retry traffic generation proportional to the queue length of the control plane are not realistic. In general, state transitions of the control plane system occur on a timescale much shorter than humans can perceive. It is natural to assume that users grow impatient and reenter service requests is not in response to the queue length at the present time but in response to the average waiting time that occurs on a longer timescale, a timescale long enough for humans to perceive. Since the rate of user reenters changes very slowly, the change of the mean of system behavior occurs slowly on a timescale long enough for humans to perceive. Let T be the minimum timescale perceptible to humans. We assume that the control plane 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 assume a quasi-static steady state for the generation of retry traffic from the control plane. The key assumptions are as follows:

- The system can be assumed to be in a steady state on a timescale shorter than T.
- Changes in the system are observed at discrete times occurring at an interval of T.
- Retry traffic from the control plane system at $t = k$ is proportional to the average queue length determined by the steady-state probability of the control plane system at $t = k - 1$
- Retry traffic from the data plane system at $t = k$ is proportional to the loss ratio determined by the steady-state probability of the data plane system at $t = k - 1$.

Since this paper deals with retry traffic generated when users respond to an increase in waiting time, we treat the behavior of the control plane system occurring on a timescale shorter than T separately, and consider the system as being in a quasi-static steady state on such a timescale. To do so, it is necessary to determine the appropriate value of T. The value of T must satisfy the following requirements:

- T must be such that humans can actually perceive an increase in the waiting time.
- T must be sufficiently longer than the timescale at which state transitions occur in the system so that it is possible to assume that the system is in a steady state at timescales under T.

Reference [5] classifies the timescales according to human perception. From this, one second is a reasonable value for T.

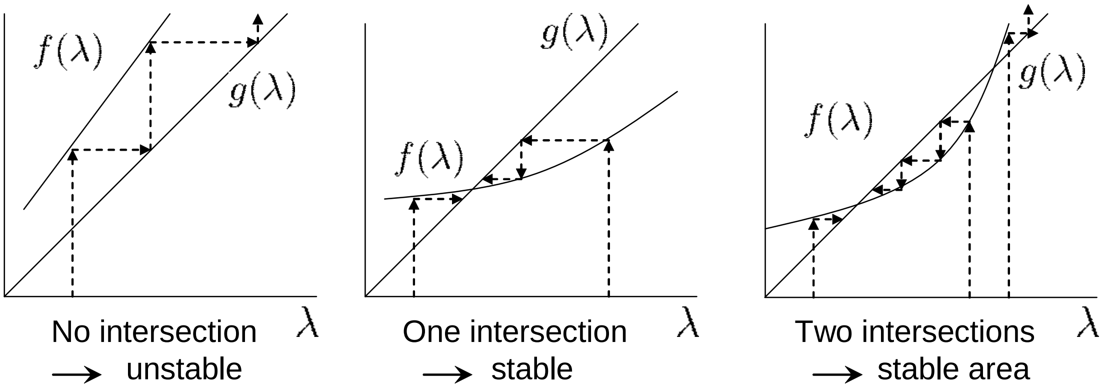

Fig. 3. System stability depends on the presence of intersections between $f(\lambda)$ and $g(\lambda)$

C. Input of traffic(including retry traffic) and stability

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

$$\lambda_{k+1} = \lambda_0 + \lambda_k B(\rho_k, s) + \varepsilon \frac{\rho_k/a}{1-\rho_k/a}, \quad (2)$$

where λ_0 is the traffic input rate without retry traffic, $\rho_k = \lambda_k/\mu$, $1/\mu$ is the average service time of the data plane system, a is the ratio of $1/\mu$ to the processing time of the control plane server, i.e., the ratio between the relative processing powers of the two servers ($a = \eta/\mu$ where $1/\eta$ is the average service time of the control plane system), $B(\rho_k, s)$ is an Erlang B formula, and $varepsilon$ is a positive constant indicating the intensity of the retry traffic generated due to control plane resource shortage. The structure of Eq.(2) is as follows. The first term on the right-hand side is the input rate of original traffic (excluding retry traffic). The second term corresponds to the rate of retries from the data plane, and denotes the number of losses occurring at M/G/s/s per unit time at time k. The last term corresponds to the rate of retries from the control plane system. The quantity $(\rho_k/a)/(1-(\rho_k/a))$ denotes the average number of service requests in the M/M/1 system.

Next, we consider the stability of the system. We assume that the requirement for the system to be stable is that the volume of traffic, including retry traffic, does not diverge after sufficient elapsed time. Namely, $\lim_{k\to\infty} \lambda_k < \infty$. This can be verified by defining two functions of λ, $f(\lambda)$ and $g(\lambda)$ as follows, and examining whether they intersect.

$$f(\lambda) = \lambda_0 + \lambda B(\rho, s) + \varepsilon \frac{\rho/a}{1-\rho/a}, \quad g(\lambda) = \lambda, \quad (3)$$

where $\rho = \lambda/\mu$. Since $\lambda_0 > 0$, $f(0) > g(0)$. The relationship between $f(\lambda)$ and $g(\lambda)$ in some typical cases is as shown in Fig.3. If the input traffic at a certain time is λ, the input traffic after one unit time becomes $\{g^{-1} \circ f\}(\lambda)$, followed by $\{g^{-2} \circ f\}(\lambda)$, etc. In general, the input traffic after the elapse of n units of time becomes $\{g^{-1} \circ f\}^n(\lambda)$. If the retry traffic from the control plane system is not negligible ($\epsilon > 0$ and $a < \infty$), the third term on the right-hand side of Eq.(2) always yields $\lim_{n\to\infty}\{g^{-1} \circ f\}^n(\lambda) > 1$. Therefore, $f(\lambda)$ and $g(\lambda)$ may intersect at two points or at one point, or may not intersect at all. As shown in the left-hand chart of Fig.3,if $f(\lambda)$ and $g(\lambda)$ do not intersect, $\lim_{n\to\infty}\{g^{-1} \circ f\}^n(\lambda) = \infty$, the system is instable. If, on the other hand, the two functions intersect as shown in the middle chart, and if $\lim_{n\to\infty}\{g^{-1} \circ f\}^n(\lambda) < 1$, then the system is stable, and if $\lim_{n\to\infty}\{g^{-1} \circ f\}^n(\lambda) > 1$, then the system is stable for all λ to the left of the right-hand intersection.

Since our previous studies assumed that all traffic including retry traffic have the same holding-time distribution of calls, average service rate per server in the data plane was considered to be constant. However, average service rate per server in the data plane is not constant when retry traffic has a different service time distribution, and it always changes under the influence of the system condition. Therefore, we need to generalize the quasi-static approach.

III. Generalization of quasi-static approach for retry traffic with different service Time

This section considers the behavior of retry traffic generated due to a shortage of control plane resources in detail, and describes why retry traffic from the control plane has different service time in the data plane. In addition, in order to extend the quasi-static approach, we discuss the change in average service rate per server when the retry traffic has different service time.

A. Behavior of retry traffic in data plane

As shown in Sect.1, retry traffic generated due to a shortage of control plane resources overlaps if calls that have been already input are not cancelled. Even if some of the users who have been kept waiting due to increased processing time attempt to initiate a new service request and establish some connection, the call for the user is only once, and other call requests is considered to be no intention to call. Therefore, behavior of retry traffic from the control plane in the data plane is clearly different from others; a broad classification is as follows

1) Data plane resources are released at the request of a user after the cal is finished.

2) System detects that there is no intention to call when the connection is established, and data plane resource is released automatically by system at timeout.

In addition, if all servers in data plane are busy and service requests by retry traffic from control plane are discarded, it is not common for the user to re-enter the call as a new service requests. In this paper, we assume that service requests created by retry traffic from the control plane are not returned.

B. The temporal evolution of average service rate per server in data plane

In this subsection, we consider the process underlying the change in average service rate per server when the retry traffic has different service time. We consider state transitions in unit time; let t $(0 \leq t < T)$ denote the start of discrete time period k just before the start of discrete time period $k+1$. Therefore, time t in discrete time period k is denoted as $k \cdot T + t$.

First, we set two service rates in the data plane; they reflect the difference in data plane server response to two types of input traffic to the data plane. Let μ_1 be the service rate of the data plane server in handling original traffic and retry traffic from data plane, and let μ_2 be the service rate of the data plane server in handling retry traffic from the control plane. Next, at time t in discrete time period k, let $n_k(t)$ be the total of service requests with service rate μ_1 in the data plane. Similarly, let $m_k(t)$ be the total of service requests with service rate μ_2 in the data plane. Next, let $\mu_k(t)$ be the overall average data plane service rate at time t in discrete time period k as follows.

$$\mu_k(t) = \frac{\mu_1 \cdot n_k(t) + \mu_2 \cdot m_k(t)}{n_k(t) + m_k(t)}. \tag{4}$$

We consider that the service request rate is more than or less than data plane system capacity.

C. Service request rate is less than data plane system capacity

In this subsection, we consider the case that the service request rate is less than data plane system capacity, $\lambda_k < s\mu_k(t)$. Here, we assume that λ_k is constant at discrete time T. When $\lambda_k < s\mu_k(t)$, since the loss yielded by resource shortages in the data plane is very small, retry traffic from data plane can be disregarded.

The temporal evolution of total service requests with service rate μ_1 in data plane $n_k(t)$ is as follows,

$$\frac{\mathrm{d}}{\mathrm{d}t} n_k(t) = \lambda_0 - n_k(t)\,\mu_1. \tag{5}$$

The first term on the right-hand side is the input rate with service rate μ_1 in order to receive service in the data plane system. The second term is output rate with service rate μ_1 in order to complete service in the data plane system. The general solution of $n_k(t)$ is obtained as follows.

$$n_k(t) = C_k e^{-\mu_1 \cdot t} + \frac{\lambda_0}{\mu_1}, \tag{6}$$

where,

$$C_k = C_{k-1} e^{-\mu_1 \cdot T}. \tag{7}$$

Next, we consider the temporal evolution of total service requests with service rate μ_2 in data plane $m_k(t)$. That is,

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k(t) = \varepsilon \cdot \frac{\lambda_{k-1}/\eta}{1 - \lambda_{k-1}/\eta} - m_k(t)\mu_2. \tag{8}$$

The first term on the right-hand of Eq.(8) is input rate with service rate μ_2 in order to receive service in the data plane system. The second term is output rate with service rate μ_2 in order to complete service in the data plane system. The general solution of $m_k(t)$ is obtained as follows.

$$m_k(t) = C_k e^{-\mu_2 \cdot t} + \frac{\varepsilon \cdot \frac{\lambda_{k-1}/\eta}{1-\lambda_{k-1}/\eta}}{\mu_2}, \tag{9}$$

where,

$$C_k = C_{k-1} e^{-\mu_2 \cdot T} + \frac{\varepsilon \cdot \frac{\lambda_{k-2}/\eta}{1-\lambda_{k-2}/\eta} - \varepsilon \cdot \frac{\lambda_{k-1}/\eta}{1-\lambda_{k-1}/\eta}}{\mu_2}. \tag{10}$$

Therefore, we obtained $n_k(t)$ and $m_k(t)$ when the service request rate is less than data plane system capacity. By entering these terms into Eq.(4), we get the overall average data plane service rate at time Zt in discrete time period k.

D. Service request rate is more than data plane system capacity

In this subsection, we consider the case that the service request rate is more than data plane system capacity, $\lambda_k > s\mu_k(t)$. Here, we assume that λ_k is constant at discrete time T. When $\lambda_k < s\mu_k(t)$, since all servers of the data plane are always busy, retry traffic from data plane and all s servers can be approximated as,

$$\lambda_k B(\rho, s) \approx \lambda_k - s\mu_k(t), \tag{11}$$

$$n_k(t) + m_k(t) \approx s, \tag{12}$$

and we have

$$\frac{s\mu_k(t)}{\lambda_k}. \tag{13}$$

This approximation represents the probability that input traffic can enter the data plane system and receive service.

The temporal evolution in total service requests with service rate μ_1 in the data plane is

$$\frac{\mathrm{d}}{\mathrm{d}t} n_k(t) = A_k \cdot \frac{s\mu_k(t)}{\lambda_k} - n_k(t)\,\mu_1, \tag{14}$$

where the first term on the right-hand side is the input rate with service rate μ_1 that receives service in the data plane system. It is given by the product of input rate with service rate μ_1, A_k, and Eq.(13). Input rate with service rate μ_1 is

$$A_k = \lambda_0 + (\lambda_{k-1} - s\mu_{k-1}(t)) \left(1 - \frac{\varepsilon \cdot \frac{\lambda_{k-2}/\eta}{1-\lambda_{k-2}/\eta}}{\lambda_{k-1}} \right), \tag{15}$$

where the first term on the right-hand side of A_k is the traffic input rate without retry traffic, and the second term is retry traffic from data plane except retry traffic from control plane. The second term on the right-hand side of Eq(14) is the output rate with service rate μ_1 in order to finish service in the data plane system.

Next, the temporal evolution in total service requests with service rate μ_2 in data plane is

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k(t) = \varepsilon \cdot \frac{\lambda_{k-1}/\eta}{1-\lambda_{k-1}/\eta} \cdot \frac{s\mu_k(t)}{\lambda_k} - m_k(t)\mu_2. \tag{16}$$

The first term on the right-hand of Eq.(16) is the retry traffic from control plane and has the input rate with service rate μ_2. The second term is the output rate with service rate μ_2.

Here, Eq.(14) and Eq.(16) have two functions of t. However, considering Eq.(4) and Eq.(12) shows that there is basically only one function of t. Therefore, we consider $\mu_k(t)$. First, using both Eq.(4) and Eq.(12), $n_k(t)$ and $m_k(t)$ are as follows

$$n_k(t) = \frac{s\mu_k(t) - s\mu_2}{\mu_1 - \mu_2}, \tag{17}$$

$$m_k(t) = \frac{s\mu_k(t) - s\mu_1}{\mu_2 - \mu_1}. \tag{18}$$

Next, we substitute differentiated Eq.(4) with respect to t in Eq.(17), (18), (14) and (16), $\mu_k(t)$ as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu_k(t) + \alpha_k \cdot \mu_k(t) = \mu_1\mu_2, \tag{19}$$

where,

$$\alpha_k = \mu_1 + \mu_2 - \frac{\mu_1 \cdot A_k + \mu_2 \cdot \varepsilon \frac{\lambda_{k-1}/\eta}{1-\lambda_{k-1}/\eta}}{\lambda_k}. \tag{20}$$

The general solution of $\mu_k(t)$ is obtained as follows.

$$\mu_k(t) = C_k e^{-\alpha_k \cdot t} + \frac{\beta}{\alpha_k} \tag{21}$$

$$C_k = C_{k-1} e^{-\alpha_{k-1}\cdot T} + \frac{\mu_1\mu_2}{\alpha_{k-1}} - \frac{\mu_1\mu_2}{\alpha_k}. \tag{22}$$

Therefore, we have obtained the overall service rate in the data plane, regardless of whether the input rate is more than or less than data plane system capacity.

IV. Evaluation of generalized quasi-static approach

A. Reproducibility of temporal change of service rate

We evaluated the service rate obtained in the previous section through simulations. In simulation, system work finite speed and continuous time. Fig.(4) shows the reproducibility of temporal change of service rate when retry traffic from the control plane has longer service time in the data plane than the other traffic. We assume that $\lambda_0 = 15000$, $\varepsilon = 50$, $\eta = 20000$, $s = 300$, $\mu_1 = 100$ and $\mu_2 = 1$. The horizontal axes denote time and the vertical axes denote service rate in the data plane. The points indicate simulation results and the solid line is predictive value obtained by the generalized quasi-static approach. These result show that generalized quasi-static approach can reproduce temporal change of service rate.

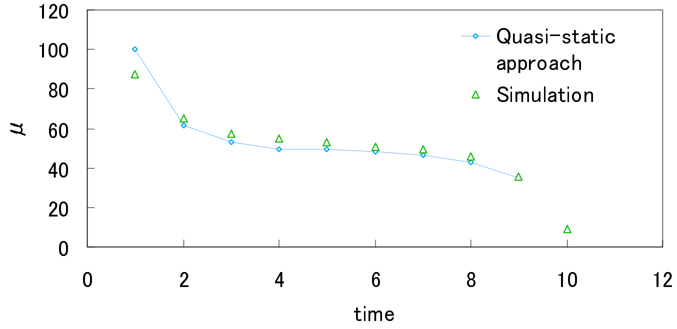

Fig. 4. Temporal change of service rate

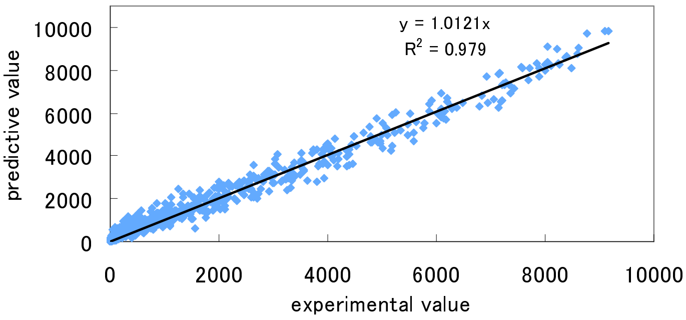

Fig. 5. Predicted number of retries from data plane

B. Predicting the number of retries from data plane

This subsection uses the genralized quasi-static approach to predict the number of retries from data plane. Eq.(2) indicates that service rate μ is related only to the retry traffic from the data plane (the second term on the right-hand side). First, it is necessary to appropriately set the service rate. The situation in which the service rate change constantly, service rate at discrete time k is influenced by service rate at discrete time $k-1$. Therefore, we employ the average service rate per $T = 1$ at discrete time k in order to predict the number of retries from data plane at discrete time $k+1$.

$$\int_0^1 \mu_k(t)dt. \tag{23}$$

Figure5 plots the predicted number of retries from the data plane. The horizontal axes plot simulation value and the vertical axes plot the value from the generalized quasi-static approach. This result shows that the average service rate per $T = 1$ can predict the number of retries from data plane.

V. Conclusions

This paper has introduced a generalization of the quasi-static approach to handle the realistic situation that retry traffic from the control plane has different holding-time distributions. This required an assessment of the temporal evolution in service rate in the data plane. In addition, we have confirmed that the average service rate per $T = 1$ can predict the number of retries from data plane.

We plan to evaluate the situation when retry traffic from the control plane has a shorter service time in the data plane than the other traffic.

ACKNOWLEDGMENT

This work was partly supported by a Grant-in-Aid for Scientific Research (B) No. 21300027 (2009–2011) from the Japan Society for the Promotion of Science, and Early-concept Grants for Exploratory Research on New-generation Network from NICT.

REFERENCES

[1] Masaki Aida, Chisa Takano, Masayuki Murata and Makoto 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, February 2008.

[2] J.R. Artalejo, Accessible bibliography on retrial queues, *Mathematical and Computer Modeling*, 30, 1–4 (1999).

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

[4] Ren-Hung Hwang, Chia-Yi, James F. Kurose and Don Towsley, On-call processing delay in high speed networks, *IEEE/ACM Transactions on Networking*, 3(6), 628–639 (1995).

[5] Jakob Nielsen “Response times : the three important limits.”Excerpt from Chapter 5 of Usability Engineering by J.Nielsen,Academic Press,1993