IEICE TRANS. COMMUN., VOL.E95–B, NO.7 JULY 2012

PAPER

Unsupervised Ensemble Anomaly Detection Using Time-Periodic Packet Sampling*

Masato UCHIDA[†a)], **Shuichi NAWATA**[††b)], ***Members***, **Yu GU**[†††c)], ***Nonmember***, **Masato TSURU**[††††d)], ***Member***, ***and*** **Yuji OIE**[††††e)], ***Fellow***

SUMMARY We propose an anomaly detection method for finding patterns in network traffic that do not conform to legitimate (i.e., normal) behavior. The proposed method trains a baseline model describing the normal behavior of network traffic without using manually labeled traffic data. The trained baseline model is used as the basis for comparison with the audit network traffic. This anomaly detection works in an unsupervised manner through the use of time-periodic packet sampling, which is used in a manner that differs from its intended purpose — the lossy nature of packet sampling is used to extract normal packets from the unlabeled original traffic data. Evaluation using actual traffic traces showed that the proposed method has false positive and false negative rates in the detection of anomalies regarding TCP SYN packets comparable to those of a conventional method that uses manually labeled traffic data to train the baseline model. Performance variation due to the probabilistic nature of sampled traffic data is mitigated by using ensemble anomaly detection that collectively exploits multiple baseline models in parallel. Alarm sensitivity is adjusted for the intended use by using maximum- and minimum-based anomaly detection that effectively take advantage of the performance variations among the multiple baseline models. Testing using actual traffic traces showed that the proposed anomaly detection method performs as well as one using manually labeled traffic data and better than one using randomly sampled (unlabeled) traffic data.

key words: *anomaly detection, packet sampling*

1. Introduction

Anomaly detection is the process of finding patterns in the audit network traffic that do not conform to legitimate (i.e., normal) behavior. The nonconforming patterns are called "anomalies." Anomalies reflect the existence of malicious activities such as worms, port scans, denial of service attacks, and spoofing, and can seriously affect the operation and normal use of the network and can cause an enormous waste of network resources and economic loss. Consequently, anomaly detection has become an important issue in network monitoring and network security [2]–[7].

The design of an anomaly detection method usually relies on a baseline model describing the normal behavior of network traffic. An alarm is raised if a pattern in the traffic deviates from the model. The model is generally trained using data extracted from traffic for which all instances (i.e., packets) are labeled in advance as either normal or anomalous. However, labeling traffic data is prohibitively expensive and time-consuming because it is usually done by human experts.

In this paper, we have developed a method for training a baseline model without using manually labeled traffic data. The proposed method employs packet sampling for the purpose of extracting normal packets from the unlabeled original traffic data: the purpose is different from which it was intended. The proposed method uses this extracted traffic data to train a baseline model. This enables anomaly detection to be executed in an unsupervised manner, unlike a conventional supervised method that uses manually labeled traffic data to train a baseline model.

The efficiency and ease of use of packet sampling for network monitoring have led it its widespread use on high-speed backbone routers to minimize resources used [8]–[10]. However, since packet sampling is inherently lossy, the sampled traffic is simply an incomplete approximation of the actual network traffic. We look at these drawbacks of packet sampling from a different perspective. That is, we expect that the sampled (unlabeled) traffic would be favorably biased to the normal traffic by skipping the periods in which burst anomalies occur. This approach differs from that of other research on packet sampling in which the intent was to reduce the bias of the sampled traffic. We confirmed that this conjecture holds true by analyzing actual traffic traces. We found that a baseline model trained using time-periodically sampled (unlabeled) traffic data is comparable in performance for detecting anomalies regarding TCP SYN packets to one trained using manually labeled traffic data and is better in performance than a baseline model trained using randomly sampled (unlabeled) traffic data. Since the performance of time-periodic sampling varies from trial to trial because of the probabilistic nature of the sampling procedure, we devised an ensemble anomaly detection method that collectively exploits multiple baseline models that are trained independently using the time-periodically sampled

Manuscript received January 14, 2011.
Manuscript revised December 27, 2011.

[†]The author is with the Department of Electrical, Electronics and Computer Engineering, Faculty of Engineering, Chiba Institute of Technology, Narashino-shi, 275-0016 Japan.
[††]The author is with KDDI R&D Laboratories Inc., Fujimino-shi, 356-8502 Japan.
[†††]The author is with Amazon Web Services, 410 Terry Ave. North Seattle, WA, 98109 USA.
[††††]The authors are with Network Design Research Center, Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.

*The present paper is an extended version of a previous paper presented at the 13th IEEE Global Internet Symposium 2010 [1].

a) E-mail: masato.uchida@ieee.org
b) E-mail: nawata@kddilabs.jp
c) E-mail: yuguamz@amazon.com
d) E-mail: tsuru@cse.kyutech.ac.jp
e) E-mail: oie@cse.kyutech.ac.jp

DOI: 10.1587/transcom.E95.B.2358

traffic data. This mitigates the variation and improves the overall performance. Since the desired alarm sensitivity depends on the intended use, we also devised maximum- and minimum-based anomaly detection methods that effectively take advantage of the variations among the multiple baseline models.

This paper is organized as follows. Section 2 provides a brief review of related work on anomaly detection and packet sampling. Section 3 explains the fundamental idea behind the proposed method. Section 4 describes the experimental results obtained using actual traffic traces. Section 5 concludes the paper with a summary of the key points.

2. Related Work

2.1 Intrusion Detection

The process of securing a network infrastructure by scanning the network for suspicious activities is generically referred to as intrusion detection. The approaches to intrusion detection can be roughly classified into two categories: signature detection and anomaly detection.

2.1.1 Signature Detection

In signature detection, the most widely deployed and commercially viable approach to detecting intrusions, the detection system identifies specific traffic patterns by matching the audited traffic data against the signatures of known attacks. The signatures are usually provided by human experts who investigate from the port number in the packet header to a specific byte sequence in the payloads of a series of packets. Snort [11] and Bro [12] are well-known open source systems that use signature detection. One of the benefits of this approach is that, once a signature database has been established, known attacks can be reliably detected with a low false positive rate. However, an alarm is not raised for attacks not in the database. For complete protection, the detection system must have a signature database containing all possible attacks, and the database must be manually updated whenever a new type of attack is discovered. Before such an update is made, the system is vulnerable to the new attack, meaning that the database must be frequently updated.

2.1.2 Anomaly Detection

In anomaly detection, a baseline model is built for describing the normal behavior of network traffic. An alarm is raised if a pattern identified in the audited traffic data deviates from the baseline model. Unknown attacks can thus be detected because their behavior will deviate from the baseline model. Another benefit is that it is potentially easier to maintain than the approach based on the signature detection because we do not need to update any signature records. Although false alarms are inevitable, the two benefits make the anomaly detection approach a promising area of research, and a number of methods based on this approach have been proposed [2], [3]. A number of these methods are variations of the change detection method; they include adaptive threshold [13], cumulative sum [14], wavelets [15], and maximum entropy [16]. In addition, a method exploiting multiple existing anomaly detection algorithms in parallel has been used to increase the accuracy of anomaly detection [17].

We are interested in the anomaly detection approach, in which a baseline model is conventionally trained using normal traffic data extracted from labeled traffic data, where the label associated with an instance (i.e., a packet) denotes whether the instance is normal or anomalous. The basic problem with this is that obtaining labeled traffic data is usually prohibitively expensive and time-consuming because it is labeled by human experts. We have thus developed a method for training a baseline model without using labeled traffic data. The use of only unlabeled traffic data means that this anomaly detection method is unsupervised. The fundamental idea is to take advantage of the lossy nature of packet sampling for the purpose of extracting normal packets from given unlabeled traffic data. Although methods using the clustering approach without data labeling are efficient [18], consideration of a different approach based on a new idea should promote the development of unsupervised anomaly detection.

2.2 Packet Sampling

Packet sampling has been attracting more and more attention as a way to minimize the resources needed for monitoring traffic passing through high-speed backbone routers [19]. Modern routers already incorporate this technique, e.g., sFlow [8] and NetFlow [9]. Moreover, the Packet Sampling (PSAMP) Working Group [10] of the Internet Engineering Task Force (IETF) has standardized packet sampling techniques. Although packet sampling provides greater scalability for network measurements [19]–[22], it makes inferring the original traffic characteristics much more difficult because it is inherently lossy. Therefore, accuracy is degraded if sampled traffic is used as input for anomaly detection [23]–[25].

We take advantage of this potential drawback of packet sampling; i.e., the sampled packets do not represent the actual characteristics of the underlying traffic. This is in contrast to research in which the intent was to mitigate sampling-induced information loss in order to analyze the actual characteristics of the underlying traffic. That is, by skipping the periods in which burst anomalies occur, we should be able to obtain sampled packets biased towards normal packets. By analyzing actual traffic traces, we showed that this conjecture holds for time-periodically sampled packets that are selected at periodic instants separated by a sampling interval. On the other hand, we also showed that this conjecture does not hold for randomly sampled packets that are selected independently with a fixed sampling probability per packet called sampling rate. Although the bias of time-periodically sampled traffic is problematic

for inferring the characteristics of the underlying traffic, this bias is not a drawback in our approach. That is, we use time-periodically sampled traffic, which contains a lower ratio of anomalous packets than the original traffic, not to infer the characteristics of the underlying traffic but to train a baseline model for anomaly detection.

3. Proposed Method

3.1 Time-Periodic Packet Sampling

We use time-periodic packet sampling in order to extract normal packets from given unlabeled packet data. As shown in Fig. 1, in time-periodic packet sampling, triggers fire at times $T_n = \sum_{i=1}^{n} t_i$ [sec] for $n = 0, 1, \ldots$, where $T_0 = 0$, and t_i represents a sampling interval that follows an independent and identical exponential distribution with expectation t. That is, triggers fire in accordance with a Poisson process with rate $\tau = 1/t$. The packet arriving immediately after each trigger is sampled while the other packets in the sampling interval are not sampled. Note that a packet is not sampled if a packet does not arrive before the next trigger is fired. The time-periodic packet sampling described in [21] estimates the flow rate with a constant sampling interval, t_i. Since a constant sampling interval might bias the sampling towards a flow in which packets arrive in a precisely periodical manner, we use a variable sampling interval to avoid synchronization effects. We use the time-periodically sampled packet data to train a baseline model. This makes manual labeling before training the model unnecessary. As shown elsewhere [26], time-periodic packet sampling tends to ignore events that happen in a burst, unlike random sampling. Therefore, under the assumption that anomalous traffic is very bursty, time-periodic packet sampling should extract traffic data containing a higher ratio of normal packets than the original traffic data. This assumption is supported by previous studies of anomalous traffic, including traffic generated during a distributed denial-of-service (DDoS) or worm attack [27].

The expectation that time-periodic packet sampling should extract traffic data containing a higher ratio of normal packets than the original traffic data is theoretically supported by a simplified model. Although the simplified model may be controversial and inaccurately reflect the actual situation, we believe that it provides useful information for understanding a typical aspect of time-periodic packet sampling. Let us consider a case in which two flows are multiplexed. The packets in flow 1 arrive in accordance with a Poisson process at rate λ_1. Those in flow 2 also arrive in accordance with a Poisson process at rate Λ_2, which is continuously and uniformly distributed over $(0, 2\lambda_2)$. Note that the expected value of Λ_2 is $\mathbb{E}[\Lambda_2] = \lambda_2$. We regard flows 1 and 2 as normal and anomalous (i.e., bursty) traffic, respectively. In addition, let $A_j^{(i)}$ denote the arrival time of the j-th packet of flow i, and let $p_1^{(i)}$ denote the probability that the first packet arriving after time $T_0 = 0$ is in flow i. The probability is given as follows. The derivation of Eqs. (1) and (2) are given in Appendix A.

$$\begin{aligned} p_1^{(1)} &= \Pr\{A_1^{(1)} < T_1, A_1^{(1)} < A_1^{(2)}\} \\ &= \frac{\lambda_1}{2\lambda_2} \log\left(1 + \frac{2\lambda_2}{\tau + \lambda_1}\right), \end{aligned} \tag{1}$$

$$\begin{aligned} p_1^{(2)} &= \Pr\{A_1^{(2)} < T_1, A_1^{(2)} < A_1^{(1)}\} \\ &= 1 - \frac{\tau + \lambda_1}{2\lambda_2} \log\left(1 + \frac{2\lambda_2}{\tau + \lambda_1}\right). \end{aligned} \tag{2}$$

We can show that $p_1^{(2)}/p_1^{(1)}$ is a monotone increasing function with respect to rate τ (i.e., a monotone decreasing function with respect to the average sampling interval t) and that it satisfies

$$\frac{p_1^{(2)}}{p_1^{(1)}} < \sup_{\tau>0} \frac{p_1^{(2)}}{p_1^{(1)}} = \lim_{\tau\to\infty} \frac{p_1^{(2)}}{p_1^{(1)}} = \frac{\lambda_2}{\lambda_1}. \tag{3}$$

This means that the ratio of packets from an anomalous traffic flow (flow 2) to those from a normal traffic flow (flow 1) in time-periodically sampled packets is smaller than the corresponding ratio for the underlying traffic. The detailed proof of this is given in Appendix B.

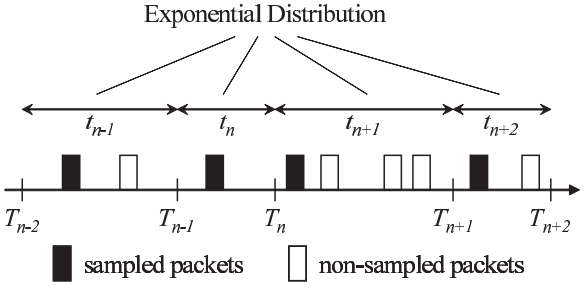


Fig. 1 Example of time-periodic packet sampling.

3.2 Training of Baseline Model Using Time-Periodically Sampled Traffic Data

We use the maximum entropy-based method to train the baseline model [16]. This method is good at detecting anomalies regarding TCP SYN packets, which make up the majority of today's significant operational threat [4]–[6]. The baseline model is defined using the generalized Gibbs distribution over a set Ω of packet class

$$P_\Xi(\omega) = \frac{1}{Z} \exp\left\{\sum_{i=1}^{|\mathcal{F}|} \xi_i f_i(\omega)\right\}$$

for $\forall\omega \in \Omega$, where Ω is defined as a set of two-dimensional classes based on the protocol information and the destination port number in the packet header. $\mathcal{F}$ denotes a set of indicator functions $f_i : \Omega \mapsto \{0, 1\}$, $i = 1, 2, \ldots, |\mathcal{F}|$, which are called feature functions. For each feature function f_i, a parameter ξ_i determines its weight in the baseline model. $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{|\mathcal{F}|}\}$ denotes the set of weight parameters. Z is a normalization constant that ensures that the sum of probabilities over Ω is 1. We use a conjugate gradient technique

on the time-periodically sampled packet data in order to obtain the maximum likelihood estimator of the set of weight parameters Ξ. Note that the original method [16] requires manually labeled packet data for the estimation.

3.3 Ensemble Anomaly Detection Using Multiple Baseline Models

We use the sliding-window-based anomaly detection approach proposed in [16]. In this approach, each time period is divided into slots with a fixed length δ [sec]. Suppose the audited traffic in a time slot contains packet sequences $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$. The empirical distribution of the packet classes in this time slot is defined as

$$\tilde{P}_\Pi(\omega) = \frac{\sum_{\pi \in \Pi} \mathbf{1}(\pi \in \omega)}{n},$$

where $\mathbf{1}(\pi \in \omega)$ takes the value 1 if packet π belongs to packet class ω and 0 otherwise. The relative entropy of packet class ω between $\tilde{P}_\Pi$ and P_Ξ is defined as

$$D_{\tilde{P}_\Pi \| P_\Xi}(\omega) = \tilde{P}_\Pi(\omega) \log \frac{\tilde{P}_\Pi(\omega)}{P_\Xi(\omega)}.$$

If, for a certain packet class ω,

$$D_{\tilde{P}_\Pi \| P_\Xi}(\omega) > d \tag{4}$$

holds for more than h times in a window with W time slots, an alarm is raised together with the packet class, ω, where d is a predefined threshold.

However, as shown in the following section, the anomaly detection performance of the baseline model, P_Ξ, trained with sampled traffic data varies from trial to trial. As mentioned, to mitigate this variation, we also devised an unsupervised ensemble anomaly detection method that exploits the multiple baseline models trained using multiple sets of sampled data. A similar idea proposed in [17] exploits multiple existing anomaly detection systems in parallel. Our method performs simultaneous time-periodic packet samplings. It independently trains multiple baseline models by using these time-periodically sampled traffic data. The trained distributions are then integrated to enable unified judgment using sliding-window-based anomaly detection. That is, in place of Eq. (4), we use

$$\frac{1}{M} \sum_{i=1}^{M} D_{\tilde{P}_\Pi \| P_{\Xi_i}}(\omega) > d, \tag{5}$$

where P_{Ξ_i}, $i = 1, 2, \ldots, M$ denotes the baseline model trained using the time-periodically sampled packet data obtained in the i-th trial, and M denotes the number of baseline models to be integrated.

The performance variations among the trained multiple baseline models can be used for other purposes. That is, rather than using Eq. (5) to mitigate the variation in anomaly detection performance, to mitigate missing anomalies, we can make use of a sensitive judgment rule:

$$\max_{i=1,2,\ldots,M} D_{\tilde{P}_\Pi \| P_{\Xi_i}}(\omega) > d. \tag{6}$$

In addition, to mitigate incorrect identification as anomalies, we can make use of an insensitive judgment rule:

$$\min_{i=1,2,\ldots,M} D_{\tilde{P}_\Pi \| P_{\Xi_i}}(\omega) > d. \tag{7}$$

4. Experimental Results

4.1 Traffic Data

We used two-way traffic traces provided by the UMass Trace Repository [28]. The traces were measured at the UMass Internet gateway router. The UMass campus is connected to the Internet through Verio, a commercial ISP, and Internet 2. Both of these connections are Gigabit Ethernet links. In particular, we used the "Gateway Link 3 Trace," the data of which was measured every morning from 9:30 to 10:30 from July 16 to 22, 2004. All the data were manually labeled, but we did not use the labels with the proposed method. A detailed description of the traces is available elsewhere [28].

4.2 Ratio of Anomalous Packets

First, we confirmed our conjecture that time-periodically sampled traffic contains a higher ratio of normal packets than the original traffic. Table 1 shows the ratio of anomalous packets for the original traffic, the time-periodically sampled traffic, and the randomly sampled traffic. The sampling intervals for the time-periodically sampled traffic followed an independent and identical exponential distribution with expectation t [sec]. The sampling rate for the randomly sampled traffic was r.

As shown in the table, the ratio of anomalous packets for the time-periodically sampled traffic was much smaller than that for the original traffic. In addition, the ratio was almost the same for two orders of magnitude of sampling intervals. This result can be intuitively understood by focusing on the timescales of the sampling and packet-arrival intervals. As mentioned above, we used traffic traces measured at the UMass Internet gateway router, where the UMass campus is connected to the Internet via two commercial Gigabit Ethernet links. The sampling intervals for the time-periodically sampled traffic followed an independent and identical exponential distribution with expectation $t = 0.1$, 0.01, 0.001 [s] (i.e., 100, 10, 1 [ms]). Therefore, the sampling intervals were sufficiently longer than the packet arrival intervals; i.e., the probability of no packets arriving during a sampling interval was sufficiently small. Since, in time-periodic packet sampling, the packet arriving immediately after each trigger is sampled while the other packets in the sampling interval are not sampled, the probability that the sampled packet is anomalous strongly depends on the rate of anomalous traffic (in terms of packets per second, or pps) at the time the trigger fires. It does not strongly depend on the (burst) duration of anomalous traffic. This relationship corresponds to $\tau\ (= 1/t) \ll \lambda_1,\ \lambda_2$ in Eqs. (1) and (2).

Table 1 Ratio of anomalous packets (%).

Date	Original Traffic	Time-Periodically Sampled Traffic			Randomly Sampled Traffic		
		$t = 0.1$	$t = 0.01$	$t = 0.001$	$r = 0.001$	$r = 0.01$	$r = 0.1$
July 16	8.48	4.20	4.32	4.67	8.43	8.47	8.48
July 17	8.71	7.49	7.59	7.86	8.46	8.71	8.71
July 18	18.18	14.68	14.80	15.35	18.31	18.19	18.18
July 19	11.02	6.13	6.19	6.57	11.07	11.02	11.02
July 20	8.36	3.50	3.50	3.79	8.37	8.36	8.36
July 21	6.62	3.17	3.20	3.39	6.63	6.61	6.62
July 22	2.97	1.36	1.35	1.47	3.00	2.97	2.97

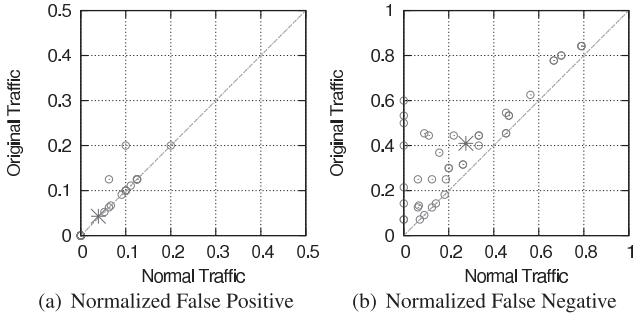
That is, the probability that the first packet arriving after a trigger is in flow i, $p_1^{(i)}$, does not depend on the sampling interval. Here, we regard flows 1 and 2 to be normal and anomalous (i.e., bursty) traffic, respectively.

In contrast, the ratio of anomalous packets for the randomly sampled traffic was almost identical to that for the original traffic. The randomly sampled traffic was not biased because each packet was simply sampled with a fixed probability regardless of whether it was anomalous or normal. Therefore, the ratio between anomalous and normal packets for the randomly sampled traffic was approximately the same as that for the original traffic.

These results indicate that time-periodic packet sampling is useful for extracting normal packets from unlabeled original traffic which may include anomalous packets. However, the time-periodically sampled traffic might be biased towards a specific aspect of normal traffic. Therefore, we investigated the performance of baseline models trained with time-periodically sampled traffic data.

4.3 Performance of Time-Periodic Packet Sampling

To evaluate the performance of time-periodic packet sampling, we trained baseline models by using different types of traffic data for a certain measurement day: normal traffic data, original traffic data, ten sets of time-periodically sampled traffic data, and ten sets of randomly sampled traffic data. We then used traffic data for other measurement days to evaluate the number of normal behaviors incorrectly identified as anomalies (false positives, FPs) and the number of missed anomalies (false negatives, FNs) regarding TCP SYN packets. For example, when traffic data for July 16 were used to train the baseline models, those for July 17 to 22 were used for the evaluation. All possible combinations of measurement days used for training and of days used for evaluation were treated. Consecutive FPs were considered to be a single FP. That is, the performance of anomaly detection was evaluated on a flow-by-flow basis. Unless otherwise noted, we used the average FP and FN for ten sampling trials for the sampling results because the results varied with each trial. We also used the normalized FP and FN numbers, which are defined as the original numbers of FPs and FNs divided by the number of anomalous incidents. Throughout this paper, we used $d = 0.005$, $W = 60$, and $h = 30$ for fair comparison.

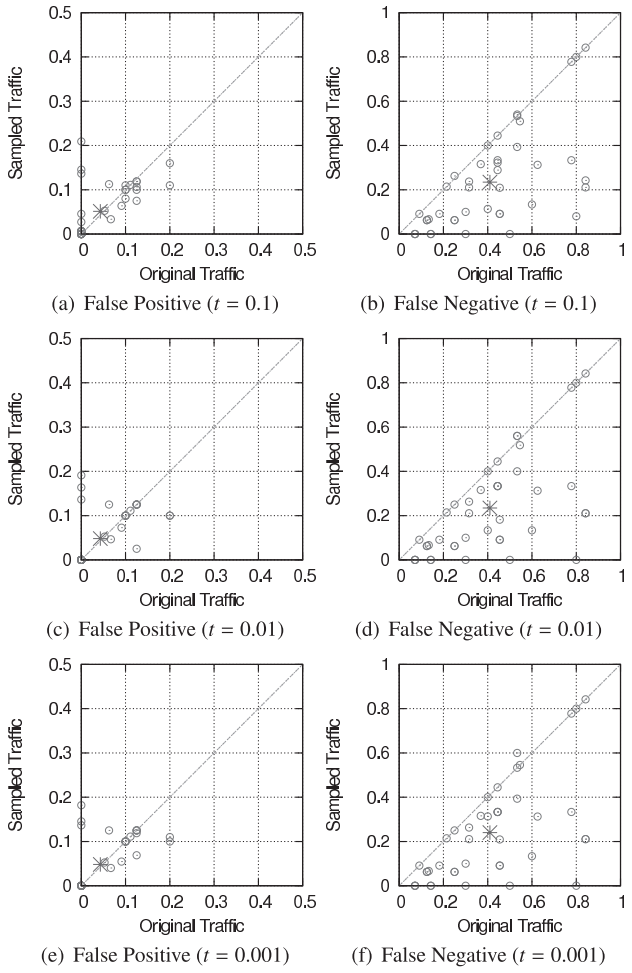


(a) Normalized False Positive (b) Normalized False Negative

Fig. 2 Original traffic vs. normal traffic.

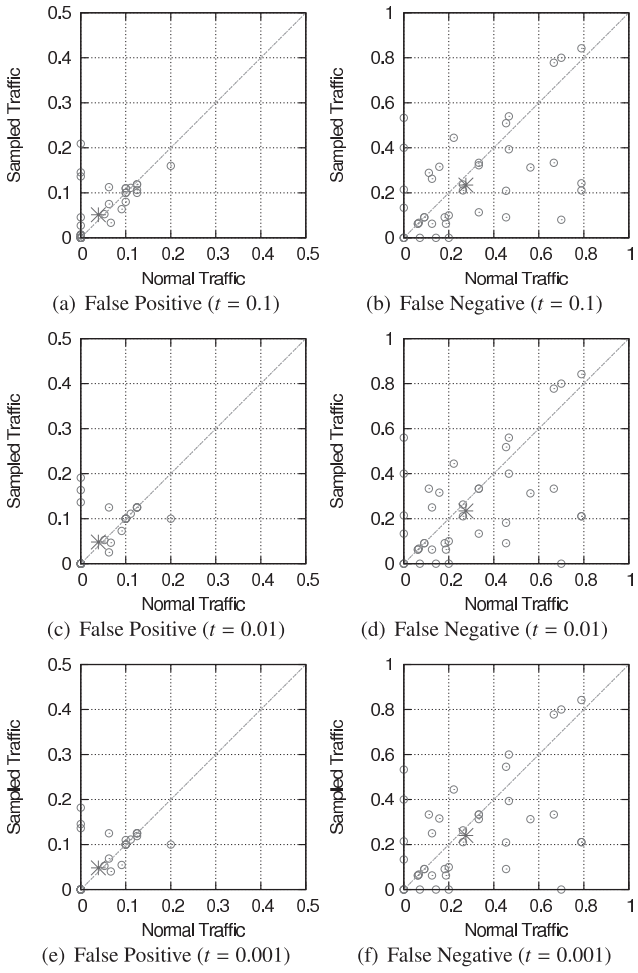
To provide a basis for comparison, we first present the comparative performance between baseline models trained using normal and original traffic data. The open circles in Figs. 2(a) and 2(b) respectively show the FPs and FNs for the baseline models trained using original traffic data against those for the baseline models trained using normal traffic data. All possible combinations of training and evaluation days were considered, as mentioned above, and each circle represents one such combination. For reference purposes, the average position of the circles is shown by the asterisk in each figure. These results indicate that both FP and FN were worse for the original traffic data than for the normal traffic data. Given this result, in the following, we show that the performance of a baseline model trained using time-periodically sampled traffic data is better than that of one trained using original traffic data (see Fig. 3) and almost identical to that of one trained using normal traffic data (see Fig. 4), while a baseline model trained using randomly sampled traffic data is not (see Fig. 5).

Figure 3 shows the comparative performance between baseline models trained using time-periodically sampled traffic data and original traffic data for $t = 0.1$, 0.01, and 0.001. As shown in Figs. 3(a), 3(c), and 3(e), the individual FP comparative performances varied (as shown by the spread of the circles). However, the average FP performance with time-periodically sampled traffic data was almost identical to that with original traffic data (as shown by the asterisk). In contrast, as shown in Figs. 3(b), 3(d), and 3(f), the FN performance with time-periodically sampled traffic data was better than that with original traffic data. Note that these



(a) False Positive (t = 0.1) (b) False Negative (t = 0.1)
(c) False Positive (t = 0.01) (d) False Negative (t = 0.01)
(e) False Positive (t = 0.001) (f) False Negative (t = 0.001)

Fig. 3 Comparative performance between time-periodically sampled traffic and original traffic (t = 0.1, 0.01, and 0.001).



(a) False Positive (t = 0.1) (b) False Negative (t = 0.1)
(c) False Positive (t = 0.01) (d) False Negative (t = 0.01)
(e) False Positive (t = 0.001) (f) False Negative (t = 0.001)

Fig. 4 Comparative performance between time-periodically sampled traffic and normal traffic (t = 0.1, 0.01, and 0.001).

results do not depend on the value of t.

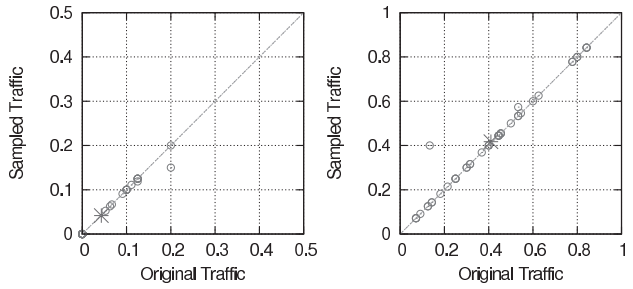
Figure 4 shows the comparative performance between baseline models trained using time-periodically sampled traffic data and normal traffic data for t = 0.1, 0.01, and 0.001. As shown in Figs. 4(a) to 4(f), the comparative performance varied for both FP and FN. However, the average performance for both FP and FN with time-periodically sampled traffic data was almost identical to that with normal traffic data.

The results shown in Figs. 3 and 4 indicate that, on average, using time-periodic packet sampling is especially effective for improving FN performance while not degrading FP performance. On the other hand, as shown in Fig. 5, the FP and FN performances with the randomly sampled traffic data were nearly identical to those with the original traffic data. This means that using randomly sampled traffic data to train the baseline models is ineffective.

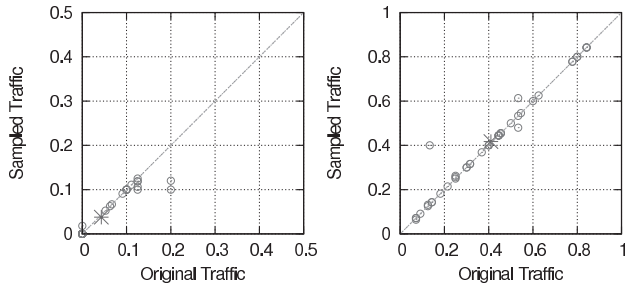
Although Fig. 4 shows that the FN performance was relatively worse than the FP performance for time-periodically sampled traffic data, the FN performance (as well as the FP performance) of the proposed method, which uses a baseline model trained with time-periodically sampled traffic, was almost identical to that of the ideal method, which uses a baseline model trained with normal traffic. This indicates that the proposed method can achieve nearly ideal performance without requiring manual labeling of the traffic data for the training of the baseline model. The FN performance could be improved by using a smaller d, but doing so would degrade FP performance. Establishing a method for adjusting the value of d so as to achieve a good balance between FP and FN performance remains for future study.

4.4 Performance of Ensemble Anomaly Detection

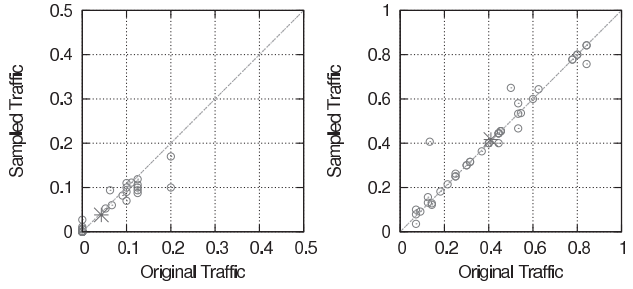
The FP and FN performances of the baseline model trained using time-periodically sampled traffic data (Figs. 3 and 4) were based on the averages over ten sampling trials. However, each individual performance before averaging would likely vary from sample to sample due to the probabilistic nature of sampled traffic data. Therefore, we compared the average performances with the best and worst individual FP and FN performances for the ten sampling trials. As shown in Fig. 6, the individual performances did indeed vary from



(a) False Positive (r = 0.001) (b) False Negative (r = 0.001)

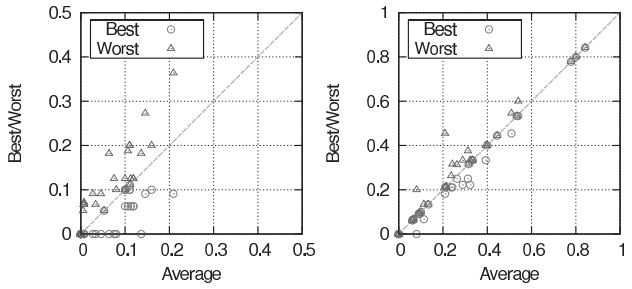


(c) False Positive (r = 0.01) (d) False Negative (r = 0.01)

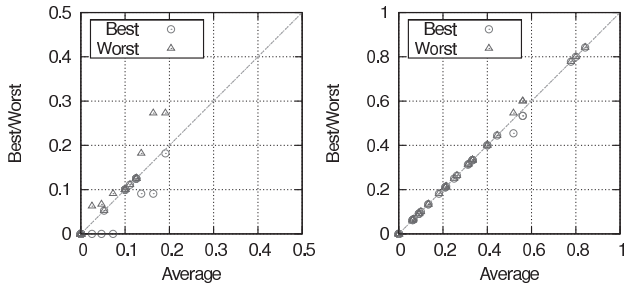


(e) False Positive (r = 0.1) (f) False Negative (r = 0.1)

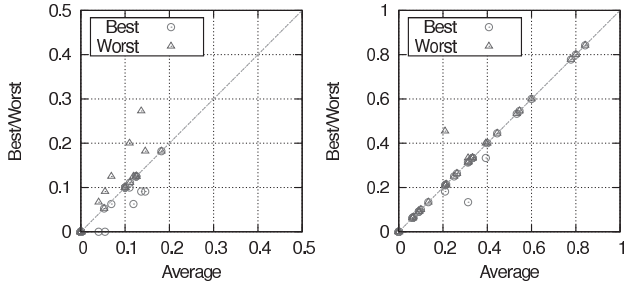
Fig. 5 Comparative Performance between Randomly Sampled Traffic and Original Traffic (r = 0.001, 0.01, and 0.1).



(a) False Positive (t = 0.1) (b) False Negative (t = 0.1)



(c) False Positive (t = 0.01) (d) False Negative (t = 0.01)



(e) False Positive (t = 0.001) (f) False Negative (t = 0.001)

Fig. 6 Comparative performance between best/worst and average (t = 0.1, 0.01, and 0.001).

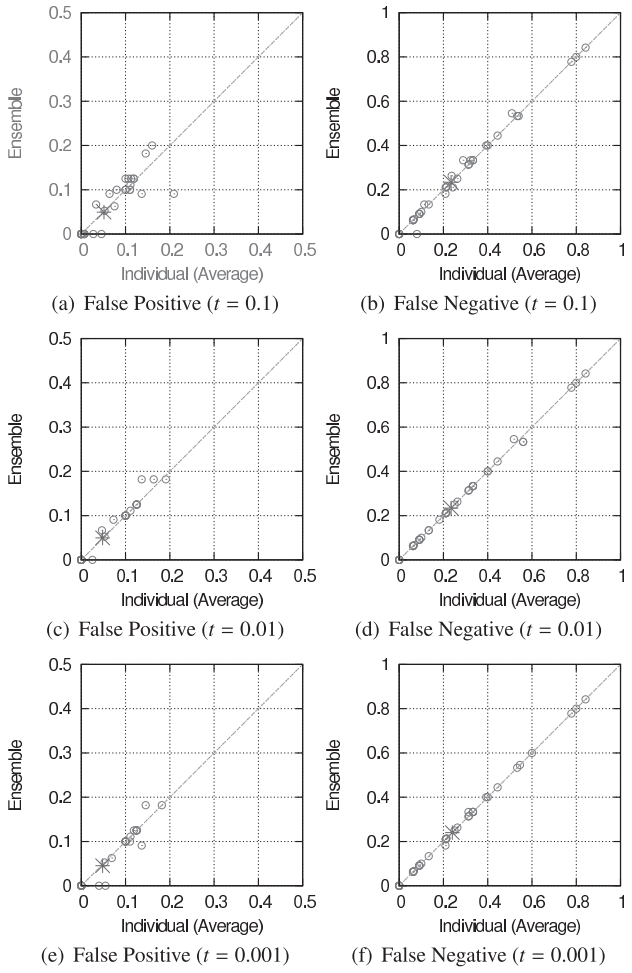
sample to sample.

As mentioned, to mitigate this performance variation, we devised an ensemble anomaly detection method. We compared the performance of individual anomaly detection (Eq. (4)) averaged over ten time-periodic sampling trials (i.e., the performances shown in Figs. 3 and 4) with that of ensemble anomaly detection (Eq. (5)) using multiple baseline models trained using ten sets of time-periodically sampled traffic data (M = 10). Figure 7 shows that the ensemble method can mitigate the performance variations among the individual baseline models for the time-periodically sampled traffic and achieve performances nearly identical to those for the average over ten sets of time-periodically sampled traffic data, which are almost identical to those for the baseline model trained using normal traffic data (Fig. 4).

We also compared the performances of ensemble anomaly detection defined by Eq. (5) with those of maximum- and minimum-based anomaly detection defined respectively by Eqs. (6) and (7). As shown in Fig. 8, the maximum-based anomaly detection improved FN performance at the expense of FP performance, while the minimum-based anomaly detection improved FP performance at the expense of FN performance. These results mean that the performance variations among trained baseline models can be used to adjust the alarm sensitivity to match the intended use. We could adjust the value of d so as to strengthen the intentions behind Eqs. (6) and (7). That is, since the intention behind Eq. (6) is to mitigate the risk of missing anomalies, it would be reasonable to use a smaller value of d to increase the alarm sensitivity. In addition, since the intention behind Eq. (7) is to mitigate the risk of incorrectly identifying anomalies, it would be reasonable to use a larger value of d to decrease the alarm sensitivity. The effect of the value of d on alarm sensitivity also remains for future study.

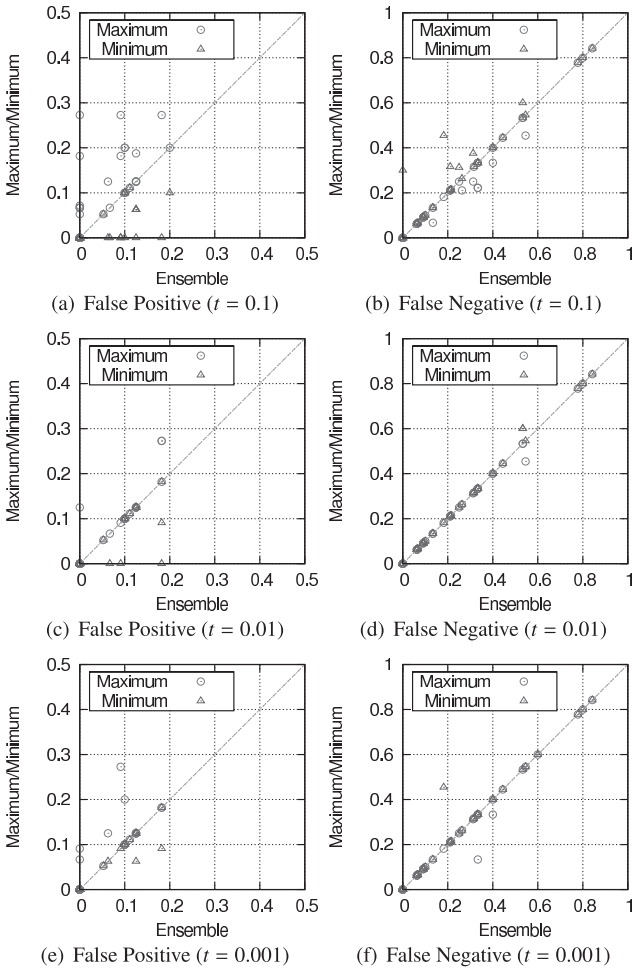
5. Conclusion

Our proposed unsupervised ensemble anomaly detection method does not require manual labeling of the traffic data used for training the baseline model. The two key ideas behind this method are (i) using packet sampling for a purpose other than that for which it was intended and (ii) improving/controlling overall anomaly detection performance



(a) False Positive (t = 0.1) (b) False Negative (t = 0.1)
(c) False Positive (t = 0.01) (d) False Negative (t = 0.01)
(e) False Positive (t = 0.001) (f) False Negative (t = 0.001)

Fig. 7 Comparative performance between ensemble anomaly detection (average) and individual anomaly detection (t = 0.1, 0.01 and 0.001).



(a) False Positive (t = 0.1) (b) False Negative (t = 0.1)
(c) False Positive (t = 0.01) (d) False Negative (t = 0.01)
(e) False Positive (t = 0.001) (f) False Negative (t = 0.001)

Fig. 8 Comparative performance between maximum/minimum-based anomaly detection and ensemble anomaly detection (t = 0.1, 0.01, and 0.001).

by using multiple baseline models in parallel. Experiments showed that (a) the use of time-periodic packet sampling results in the extraction of a higher percentage of normal packets from original unlabeled traffic, which may include burst anomalous traffic, than the use of random sampling, (b) time-periodically sampled traffic data is better than (unlabeled) original traffic data and randomly sampled traffic data for training a baseline model that is effective with respect to false positives and false negatives, (c) the performance of the proposed method can be further improved by exploiting multiple baseline models in parallel, and (d) the performance variation of multiple baseline models can be used to improved FP/FN performance at the expense of FN/FP performance. Therefore, the proposed method is preferable to one that uses manually labeled traffic data: it can save a lot of effort and is just as accurate.

Acknowledgments

This work was supported in part by the National Institute of Information and Communications Technology, the Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (21700079), and the GreenIT Project by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] S. Nawata, M. Uchida, Y. Gu, M. Tsuru, and Y. Oie, "Unsupervised ensemble anomaly detection through time-periodical packet sampling," Proc. 13th IEEE Global Internet Symposium (GI 2010), pp.1–6, San Diego, CA, USA, March 2010.
[2] J.M. Estevez-Tapiador, P. Garcia-Teodoro, and J.E. Diaz-Verdejo, "Anomaly detection methods in wired networks: A survey and taxonomy," Comput. Commun., vol.27, no.16, pp.3448–3470, Oct. 2004.
[3] A. Patcha and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Netw., vol.51, no.12, pp.3448–3470, Aug. 2007.
[4] "Worldwide infrastructure security report V, 2009," http://www.arbornetworks.com/report, Jan. 2010.
[5] "14th annual edition of the CSI computer crime and security survey, 2009," http://pathmaker.biz/whitepapers/CSISurvey2009.pdf
[6] "2010 cybersecurity watch survey — Survey results," http://www.cert.org/archive/pdf/ecrimesummary10.pdf
[7] "Defending TCP against spoofing attacks (RFC 4935)," http://tools.

ietf.org/html/rfc4953
[8] "InMon sFlow Probe," http://www.inmon.com/technology/index.php
[9] "Cisco NetFlow," http://www.cisco.com/web/go/netflow
[10] "A framework for packet selection and reporting (RFC 5474)," http://tools.ietf.org/html/rfc5474
[11] "Snort." http://www.snort.org
[12] "Bro." http://bro-ids.org
[13] V.A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting syn flooding attacks," Comput. Commun., vol.29, no.9, pp.1433–1442, May 2006.
[14] H. Wang, D. Zhang, and K.G. Shin, "Change-point monitoring for the detection of dos attacks," IEEE Trans. Dependable and Secure Computing, vol.1, no.4, pp.193–208, Oct. 2004.
[15] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," Proc. 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW 2002), pp.71–82, France, Nov. 2002.
[16] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using entropy estimation," Proc. 5th ACM SIGCOMM Conference on Internet Measurement (IMC 2005), pp.345–350, Berkeley, CA, USA, Oct. 2005.
[17] S. Shanbhag and T. Wolf, "Accurate anomaly detection through parallelism," IEEE Netw., vol.23, no.1, pp.22–28, Jan./Feb. 2009.
[18] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," ACM SIGCOMM Computer Communication Review, vol.35, no.4, pp.217–228, Oct. 2005.
[19] N. Duffield, "Sampling for passive internet measurement: A review," Statistical Science, vol.19, no.3, pp.472–498, 2004.
[20] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," IEEE/ACM Trans. Netw., vol.13, no.5, pp.933–946, Oct. 2005.
[21] S. Itou, K. Uchiyama, and S. Shioda, "Fixed-period packet sampling and its application to flow rate estimation," Proc. IEEE International Conference on Communications (ICC 2007), pp.279–286, Glasgow, Scotland, June 2007.
[22] T. Mori, T. Takine, J. Pan, R. Kawahara, M. Uchida, and S. Goto, "Identifying heavy-hitter flows from sampled flow statistics," IEICE Trans. Commun., vol.E90-B, no.11, pp.3061–3072, Nov. 2007.
[23] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," Proc. 6th ACM SIGCOMM Conference on Internet Measurement (IMC 2006), pp.159–164, Rio de Janeriro, Brazil, Oct. 2006.
[24] J. Mai, A. Sridharan, C. Chuah, H. Zang, and T. Ye, "Impact of packet sampling on portscan detection," IEEE J. Sel. Areas Commun., vol.24, no.12, pp.2285–2298, Dec. 2006.
[25] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network anomaly detection and classification via opportunistic sampling," IEEE Netw., vol.23, no.1, pp.6–12, Jan./Feb. 2009.
[26] K.C. Claffy, G.C. Polyzos, and H.W. Braun, "Application of sampling methodologies to network traffic characterization," ACM SIGCOMM Computer Communication Review, vol.23, no.4, pp.194–203, Oct. 1993.
[27] K. Lan, A. Hussain, and D. Dutta, "The effect of malicious traffic on the network," Proc. Passive and Active Measurement Workshop (PAM 2003), pp.159–164, San Diego, CA, USA, April 2003.
[28] "UMass trace repository," http://traces.cs.umass.edu/

Appendix A: Derivation of Eqs. (1) and (2)

Equation (1) was derived as follows.

$$\begin{aligned}
&\Pr\{A_1^{(1)} < T_1, A_1^{(1)} < A_1^{(2)}\} \\
&= \int_0^{2\lambda_2} \Pr\{A_1^{(1)} < T_1, A_1^{(1)} < A_1^{(2)} | \Lambda_2 = \lambda\} \frac{1}{2\lambda_2} d\lambda \\
&= \int_0^{2\lambda_2} \int_0^{\infty} \Pr\{a < T_1, a < A_1^{(2)} | A_1^{(1)} = a, \Lambda_2 = \lambda\} \\
&\quad \times \lambda_1 e^{-\lambda_1 a} \frac{1}{2\lambda_2} da d\lambda \\
&= \int_0^{2\lambda_2} \int_0^{\infty} e^{-\tau a} e^{-\lambda a} \lambda_1 e^{-\lambda_1 a} \frac{1}{2\lambda_2} da d\lambda \\
&= \frac{\lambda_1}{2\lambda_2} \int_0^{2\lambda_2} \frac{1}{\tau + \lambda_1 + \lambda} d\lambda \\
&= \frac{\lambda_1}{2\lambda_2} \log\left(1 + \frac{2\lambda_2}{\tau + \lambda_1}\right). \qquad (A \cdot 1)
\end{aligned}$$

Equation (2) was derived as follows.

$$\begin{aligned}
&\Pr\{A_1^{(2)} < T_1, A_1^{(2)} < A_1^{(1)}\} \\
&= \int_0^{2\lambda_2} \Pr\{A_1^{(2)} < T_1, A_1^{(2)} < A_1^{(1)} | \Lambda_2 = \lambda\} \frac{1}{2\lambda_2} d\lambda \\
&= \int_0^{2\lambda_2} \int_0^{\infty} \Pr\{a < T_1, a < A_1^{(1)} | A_1^{(2)} = a, \Lambda_2 = \lambda\} \\
&\quad \times \lambda e^{-\lambda a} \frac{1}{2\lambda_2} da d\lambda \\
&= \int_0^{2\lambda_2} \int_0^{\infty} e^{-\tau a} e^{-\lambda_1 a} \lambda e^{-\lambda a} \frac{1}{2\lambda_2} da d\lambda \\
&= \frac{1}{2\lambda_2} \int_0^{2\lambda_2} \frac{\lambda}{\tau + \lambda_1 + \lambda} d\lambda \\
&= 1 - \frac{\tau + \lambda_1}{2\lambda_2} \log\left(1 + \frac{2\lambda_2}{\tau + \lambda_1}\right). \qquad (A \cdot 2)
\end{aligned}$$

Appendix B: Proof of Some Properties of $p_1^{(2)}/p_1^{(1)}$

If we define x as $x = 2\lambda_2/(\tau + \lambda_1)$, the following equations hold:

$$\begin{aligned}
&\frac{p_1^{(2)}}{p_1^{(1)}} = \frac{\lambda_2}{\lambda_1} \frac{2}{x} \left(\frac{x}{\log(x+1)} - 1\right), \\
&\frac{d}{dx} \frac{p_1^{(2)}}{p_1^{(1)}} = \frac{\lambda_2}{\lambda_1} \frac{2}{x^2} \left(-\frac{1}{x+1} \frac{x^2}{(\log(x+1))^2} + 1\right).
\end{aligned}$$

In addition, the following inequalities hold:

$$\begin{aligned}
&\frac{\lambda_2}{\lambda_1} \frac{2}{x} \left(\sqrt{x+1} - 1\right) \le \frac{p_1^{(2)}}{p_1^{(1)}} \le \frac{\lambda_2}{\lambda_1}, \\
&\frac{d}{dx} \frac{p_1^{(2)}}{p_1^{(1)}} \le 0,
\end{aligned}$$

where the equalities hold if and only if $x \downarrow 0$. Here, we used that the following inequality holds for $\forall x > 0$:

$$\sqrt{x+1} \le \frac{x}{\log(x+1)} \le \frac{x+2}{2},$$

where the equalities hold if and only if $x \downarrow 0$. Note that

$$\lim_{x \downarrow 0} \frac{p_1^{(2)}}{p_1^{(1)}} = \frac{\lambda_2}{\lambda_1}$$

holds because

$$\lim_{x \downarrow 0} \frac{2}{x} \left(\sqrt{x+1} - 1 \right) = 1$$

holds. The above discussion indicates that $p_1^{(2)}/p_1^{(1)}$ is a monotonically decreasing function with respect to x (i.e., a monotonically increasing function with respect to τ) and its supremum is given by λ_2/λ_1 when $x \downarrow 0$ (i.e., $\tau \to \infty$).

Masato Uchida received B.E., M.E., and D.E. degrees from Hokkaido University, Hokkaido, Japan, in 1999, 2001, and 2005, respectively. In 2001, he joined NTT Service Integration Laboratories, Tokyo, Japan. From August 2005 to March 2012, he was an Associate Professor in the Network Design Research Center, Kyushu Institute of Technology, Fukuoka, Japan. Since April 2012, he has been an Associate Professor in the Department of Electrical, Electronics and Computer Engineering, Faculty of Engineering, Chiba Institute of Technology, Chiba, Japan. His research field is mathematical information engineering with applications to information networking and machine learning. He is a member of the IEEE and ACM.

Shuichi Nawata received B.E. and M.E. degrees from the Kyushu Institute of Technology, Fukuoka, Japan, in 2008 and 2010, respectively. He joined KDDI R&D Laboratories, Inc. in 2010 and has been engaged in research and development of analysis techniques for communication networks. He is currently a research engineer in the Communications Network Planning Laboratory of KDDI R&D Laboratories, Inc.

Yu Gu received B.S. and M.S. degrees from the Beijing University of Aeronautics and Astronautics, China, in 1998 and 2001, respectively, and a Ph.D. from the University of Massachusetts, Amherst, in 2008. He worked in NEC's research laboratory after graduation and, in 2010, he joined Amazon Web Services.

Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan, in 1983 and 1985, respectively, and a D.E. degree from the Kyushu Institute of Technology, Japan, in 2002. He worked at Oki Electric Industry Co., Ltd., Nagasaki University, and the Telecommunications Advancement Organization of Japan. In 2003, he joined the Department of Computer Science and Electronics, Kyushu Institute of Technology, as an Associate Professor and has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the IEEE, ACM, IPSJ, and JSSST.

Yuji Oie received B.E., M.E., and D.E. degrees from Kyoto University, Kyoto, Japan, in 1978, 1980, and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queuing systems. He is a fellow of the IPSJ and a member of the IEEE.