

IEICE **TRANSACTIONS**

on Communications

VOL. E102-B NO. 1
JANUARY 2019

The usage of this PDF file must comply with the IEICE Provisions on Copyright.

The author(s) can distribute this PDF file for research and educational (nonprofit) purposes only.

Distribution by anyone other than the author(s) is prohibited.

A PUBLICATION OF THE COMMUNICATIONS SOCIETY



The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Measuring Lost Packets with Minimum Counters in Traffic Matrix Estimation*

Kohei WATABE^{†a)}, Member, Toru MANO^{††}, Nonmember, Takeru INOUE^{††}, Kimihiro MIZUTANI^{††},
Osamu AKASHI^{††}, and Kenji NAKAGAWA[†], Members

SUMMARY Traffic matrix (TM) estimation has been extensively studied for decades. Although conventional estimation techniques assume that traffic volumes are unchanged between origins and destinations, packets are often lost on a path due to traffic burstiness, silent failures, etc. Counting every path at every link, we could easily get the traffic volumes with their change, but this approach significantly increases the measurement cost since counters are usually implemented using expensive memory structures like a SRAM. This paper proposes a mathematical model to estimate TMs including volume changes. The method is established on a Boolean fault localization technique; the technique requires fewer counters as it simply determines whether each link is lossy. This paper extends the Boolean technique so as to deal with traffic volumes with error bounds that requires only a few counters. In our method, the estimation errors can be controlled through parameter settings, while the minimum-cost counter placement is determined with submodular optimization. Numerical experiments are conducted with real network datasets to evaluate our method.

key words: traffic matrix, packet drop, submodular optimization, failure localization, passive measurement

1. Introduction

Traffic Matrices (TMs), which specify the traffic volumes between origin-destination pairs in a network, are used by many network engineering tasks, such as traffic engineering [1], [2], capacity planning [3], and anomaly detection [4]. These tasks rely heavily on the availability and accuracy of TMs. Due to their importance, there has been a substantial body of work on TM estimation [5]–[11]. Conventional TM estimation techniques assume the strict flow conservation, i.e., traffic volumes are unchanged along a path from the origin to the destination. Traffic, however, can be lost due to packet drops in a network. For example, bursty traffic could be dropped by the overflow of a switch's queue, even if the long-term average throughput is lower than the link capacity; silent packet drops [12] often happen, i.e., switches for unknown reasons drop packets without showing information about the drops.

This paper studies a mathematical model to estimate

Manuscript received March 12, 2018.

Manuscript revised June 13, 2018.

Manuscript publicized June 2, 2018.

[†]The authors are with the Graduate School of Engineering, Nagaoka University of Technology, Nagaoka-shi, 940-2188 Japan.

^{††}The authors are with NTT Network Innovation Laboratories, Yokosuka-shi, 239-0847 Japan.

*An earlier version of this paper was presented at the 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017) Poster Track.

a) E-mail: k_watabe@vos.nagaokaut.ac.jp
DOI: 10.1587/transcom.2018EBP3072

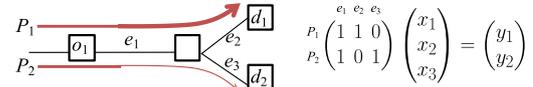


Fig. 1 (Left) Traffic volumes with change. There are three links, e_1 , e_2 , and e_3 , and two paths, P_1 and P_2 , in the network. The volumes are indicated by the arrow width. (Right) Identification of failed link with a Boolean measurement matrix. The rank of this matrix ($m = 2$) is lower than the number of variables ($n = 3$), but we can obtain the solution \mathbf{x} for any \mathbf{y} , assuming a single link failure; e.g., for $\mathbf{y} = (0 \ 1)^T$, we have $\mathbf{x} = (0 \ 0 \ 1)^T$.

TMs with lost packets as shown in Fig. 1(Left). What we want to estimate in this study differs from the traditional traffic matrix. Whereas the volume of traffic on each path is estimated in traditional traffic matrix estimation, our method estimates where and how much traffic on each path was lost/passed. Of course, counting every path at every link, we could easily get the traffic volumes with their change. The counters are, however, usually maintained on an expensive memory like a SRAM to match to the speed of broadband links, so the number of counters should be minimized to reduce the measurement cost. This paper specifically discusses the tradeoff between the measurement cost and estimation errors. There are two baseline approaches; one is the linear algebra approach which determines an upper bound for the number of counters, while the other is Boolean algebra approach which gives a lower bound.

Linear algebra approach. For the upper bound, we come up with a naive combination of TMs with link loss-ratio estimation, as follows. Let m be the number of paths and n be the number of links. Let T be an $m \times m$ traffic matrix where element t_{ii} is a traffic volume of i -th path ($i = 1, 2, \dots, m$) and the other elements are zero, let A be an $m \times n$ measurement matrix where element a_{ij} indicates whether i -th path includes j -th link ($j = 1, 2, \dots, n$), let \mathbf{x} be an n -dimensional column vector whose j -th element is the packet loss ratio on j -th link, and let \mathbf{y} be an m -dimensional column vector whose i -th element indicates volume of lost traffic on i -th path (i.e., the difference of traffic volume between the ingress and the egress on the path). Here, T and \mathbf{y} are measured using counters placed at ingress and egress switches, while A is determined from forwarding configurations. Given T , A , and \mathbf{y} , equation $TA\mathbf{x} = \mathbf{y}$ can be uniquely solved for \mathbf{x} if TA satisfies $\text{rank}(TA) = n \leq m$. This approach can exactly determine the traffic volumes with change, but it requires a full-rank matrix as TA ; if TA is

rank deficient, some paths have to be divided at intermediate switches and another counters would be put on them.

Boolean algebra approach. This approach requires fewer counters, but it only locates failed links without volume change. Instead of TA used in the above approach, we utilize the Boolean measurement matrix A with another Boolean vectors \mathbf{x} and \mathbf{y} , where j th element x_j of \mathbf{x} indicates whether j -th link is failed while i th element y_i of \mathbf{y} indicates whether i -th path is failed. Note that the norm of \mathbf{x} , $\|\mathbf{x}\|$, indicates the number of failed links. Boolean matrices have an interesting property: given $K \in \mathbb{N}$, there exists a rank deficient Boolean matrix A that yields an injective mapping, $\mathbf{x} \mapsto A\mathbf{x}$, if $\|\mathbf{x}\| \leq K$ (i.e., for sparse \mathbf{x}) [13], [14]. This property allows us to *uniquely* determine \mathbf{x} from \mathbf{y} subject to $A\mathbf{x} = \mathbf{y}$ though $m < n$. This is because, roughly speaking, the Boolean space restricted by K can be small enough to be covered by a rank deficient matrix. Therefore, at the expense of knowledge of volume change, this approach reduces the number of counters since smaller m leads to a smaller number of counters. In Fig. 1(Right), we measure only two paths for the three links, but failed links are identifiable assuming a single link failure ($K = 1$).

Our method proposed in this paper intervenes between the two approaches; the volume change is estimated with error bounds, while it requires counters fewer than the linear algebra approach and close to the Boolean algebra approach. Since exact volumes are usually not needed if their errors are bounded, this is a good tradeoff. Our method is designed as a passive scheme, which does not disrupt production traffic. Our contributions are summarized as follows.

- A new theory about the *measurability* of packet/byte counts is established (our theory handles both packet counts and byte counts in the same way, so we only address packet counts in the rest of paper). The TM estimation is reformulated for packet drops with Boolean algebra. The error bounds can be controlled by threshold parameters.
- Our optimization method that minimizes the number of counters is developed based on the measurability theory. The counter placement problem is relaxed to a submodular optimization problem, and an approximation algorithm is presented.
- Experiments with three real network datasets show that the size of the measurement matrix in our method is close to that of the Boolean algebra approach. Additionally, it is confirmed that the accuracy of our method is highly guaranteed with tight error bounds for these datasets.

The remainder of this paper is organized as follows. Section 2 clarifies the problem of conventional TM estimation techniques. Section 3 gives the problem statement. Section 4 discusses our method based on “per-path” measurement, while Sect. 5 unifies some measurement paths into a tree (some rows are merged into a single row in the matrix), thereby the number of counters is further decreased. After describing practical modeling issues in Sect. 6, Sect. 7

shows experimental results. Section 8 summarizes related work and Sect. 9 concludes the paper.

2. Conventional TM Estimation Techniques

First of all, we make the problem of conventional TM estimation techniques clear. To evaluate conventional TM estimation techniques under packet drops, we conducted preliminary experiments using Sparsity Regularized Singular Value Decomposition (SRSVD) [9]. Figure 1(Left) shows the network used in the experiments. We assume that path P_1 has 10× more traffic than P_2 . Packets are counted at nodes o_1 , d_1 , and d_2 , and traffic volumes are estimated for each path using the packet counts. Without packet drops, the estimation is very accurate. However, with packet drops, it is not; e.g., if 10% of traffic on link e_1 drops, estimated volumes are in error by up to 40% on P_2 , since the estimators do not distinguish volumes on e_1 and e_3 . These large errors are also inevitable in other conventional estimation techniques, which assume the strict flow conservation. These large estimation errors are, unfortunately, considered to be unacceptable for most applications. Accurate TMs including volume changes, i.e., where and how much packets are dropped on the path, would be very useful for advanced network engineering; e.g., capacity planning could consider lost traffic in addition to the volume, while traffic engineering could avoid lossy links and switches.

3. Model and Problem

3.1 Network Model

A network is represented by a directed graph, $G = (V, E)$ where V and E represent a set of vertexes and arcs. Given a packet, the forwarding path is determined by the packet header (e.g., addresses and ports) based on routing protocols and/or switch configurations. A path along which there exist some packets forwarded is called a *feasible* path $P \in \mathcal{P}$, where \mathcal{P} is the set of feasible paths in the network. A path is regarded as a set of arcs[†]. We do not consider multi-path forwarding nor multicasting.

Let C be a set of counters placed in the network. A counter is placed at the *tail* of directed arc $e_j \in E$ ($j = 1, 2, \dots, n$) and maintains the number of packets transmitted into the arc along the associated path (or the set of associated paths, which forms a tree). Each counter is specified by the pair of arc and path, e.g., $(e_j, P) \in C$ and $C \subseteq \tilde{C} = E \times \mathcal{P}$, and the value of the counter is denoted by κ_j . Packets are classified to the corresponding counters based on their path using path-oriented packet classification techniques [15], [16]. Counter clocks are supposed to be synchronized [17], [18]. We assume that all packets are observed without sampling some of them; our method could be extended to a sampling version using hash functions [19]. We assume that packets

[†]The word “arc”, $e \in E$, is used as an abstraction of transmission media in this paper, since arcs will be used for links as well as switch internals, as will be described in Sect. 6.

Table 1 Notation table.

Notation	Definition
G	A directed graph (V, E) that represent a network. V and E represent a set of vertexes and arcs.
P	A feasible path: a path along which there exist some packets forwarded.
\mathcal{P}	The set of feasible paths in the network.
e_j	An arc.
e_a	An abnormal arc.
(e_j, P)	A counter placed at arc e_j on path P .
C	A set of counters placed in the network.
\tilde{C}	The set of all potential counters $E \times \mathcal{P}$.
C_0	The set of ingress counters in the network.
X	A set of additional counters $C \setminus C_0$.
X_0	An intermediate counter set assuming $\epsilon = 0$.
X_{ALG}	A counter set that outputted by Algorithm 3.
X^*	The optimal counter set of (10) for given thresholds ϵ and δ .
X_0^*	The optimal counter set of (10) for given thresholds $\epsilon = 0$.
κ_j	The value of the counter e_j .
τ_j	The number of packets transmitted into e_j along P in a measurement period.
$\hat{\tau}_j$	An estimated value of τ_j .
ℓ_j	The packet loss ratio through arc e_j .
ϕ	An estimator of the pass ratio on the abnormal arc.
ϵ	The threshold of loss ratio for a normal state. Loss ratio of a normal state arc is smaller than ϵ .
δ	The threshold of loss ratio for an abnormal state. Loss ratio of a abnormal state arc is larger than δ . It also used as a loss threshold for a separable path.
d	The maximum path length $\max\{ P : P \in \mathcal{P}\}$.
Q_i	A separable subpath: a part of a feasible path between a pair of counters that satisfies $(1 - \delta) < (1 - \epsilon)^{k-j}$ (see Fig. 2).
$A(C)$	A measurement matrix whose (i, j) element a_{ij} indicates that a separable path Q_j includes an arc e_i (see (4)).
\mathbf{a}_j	The j -th column vector of a measurement matrix $A(C)$.
\mathbf{x}	A n -dimensional column vector whose j -th element indicates that j -th arc is an abnormal state.
\mathbf{y}	A m -dimensional column vector whose i -th element indicates that subpath Q_i includes an abnormal arc.
\mathbf{x}^*	The unique solution of $A\mathbf{x} = \mathbf{y}$.
g	A coverage function that represents how close the measurement matrix is to 1-independent.
g_0	A coverage function for $\epsilon = 0$.
$H(x)$	x -th harmonic number $\sum_{i=1}^x 1/i$.

are not fragmented in a network.

The packet loss ratio is defined by ratio of lost packets to input packet of an arc, and the packet loss ratio through arc e_j is denoted by $\ell_j \in [0, 1]$. We assume that every arc has *either* a normal state $\ell_j \leq \epsilon$ or an abnormal state $\ell_j \geq \delta$, where $\epsilon < \delta$; it is worth noting that our model works without the strong assumption, $\epsilon \ll \delta$, used in [14]. We assume that every path observes the equal loss ratio for the same abnormal arc; e.g., in Fig. 1, if arc e_1 were abnormal and the other arcs were loss-less, the both paths (P_1 and P_2) would experience the equal loss ratio. For the sake of simplicity, this paper assumes a single arc failure, i.e., $K = 1$. However, our theory could be easily extended for multiple failures. Packets are dropped only at arcs, not at vertexes; this issue will be elaborated in Sect. 6.

For reference, we list our notation throughout the paper in Table 1.

3.2 Problem Statement

Let $\tau_j \in \mathbb{R}$ be the number of packets transmitted into e_j along P in a measurement period; our model has $\kappa_j = \tau_j$ if a counter is placed at (e_j, P) . The loss ratio ℓ_j is expressed by $1 - \tau_{j+1}/\tau_j$ for path P shown in Fig. 2. Let $\hat{\tau}_j$ be an estimated value of τ_j (Note that counters are not necessarily placed at all arcs, and τ_j cannot be obtained directly from counter

values if a counter is not placed at (e_j, P)). The counter set C is α -measurable if,

$$\forall P \in \mathcal{P}, \forall e_j \in P, \forall \tau_j : \alpha \cdot \tau_j \leq \hat{\tau}_j \leq \frac{1}{\alpha} \cdot \tau_j, \quad (1)$$

where $0 < \alpha \leq 1$. Our problem is defined as follows: for given G and \mathcal{P} ,

$$\min_{C \subseteq E \times \mathcal{P}} \{|C| : C \text{ is } \alpha\text{-measurable}\}, \quad (2)$$

where $|C|$ is the number of counters and α is a constant that satisfies $0 < \alpha \leq 1$ (this optimization problem will be redefined rigorously in Sect. 4.3 after establishing the theory).

There is a trivial solution of 1-measurable; if we put counters at every arc on every feasible path, i.e., $C = \{(e, P) : P \in \mathcal{P}, e \in P\}$, it is clearly 1-measurable, but it requires many counters, $|C| = \sum_{P \in \mathcal{P}} |P| \leq d|\mathcal{P}|$, where $d = \max\{|P| : P \in \mathcal{P}\}$. Our target is an α -measurable counter set whose size is close to the number of feasible paths, since this paper tries to estimate volume change at each arc given ingress volumes for each feasible path, $|\mathcal{P}|$ is the lower bound for $|C|$.

4. Per-Path Measurement

This section proposes a polynomial-time algorithm that finds a counter set that is α -measurable, based on ‘‘per-path’’ traffic measurement; per-path measurement means that every counter is associated with a single feasible path (Sect. 5 associates some counters with a set of paths).

Before going into detail, we overview this section; the relationship among the measurement matrix $A(C)$, the measurability condition, and the algorithm. Intuitively, each column of the matrix corresponds to the arc and each row corresponds to the path on which we can distinguish whether it contains abnormal arc (Fig. 1). If any column vector of $A(C)$ is different from each other and none of them equal to zero, the matrix $A(C)$ is called *1-independent*. When $\alpha = (1 - \epsilon)^{d-2}$, 1-independence is a sufficient condition for making a counter set C α -measurable. Hence, we reformulate our problem as follows,

$$\min_{C \subseteq E \times \mathcal{P}} \{|C| : A(C) \text{ is 1-independent}\} \quad (3)$$

and leverage *submodularity* [20], which is a discrete analog of convexity, of the measurement matrix $A(C)$ and propose polynomial-time algorithm with a theoretically guaranteed approximation ratio $O(\log(n))$.

In the rest of this section, Sect. 4.1 defines the measurement matrix and other related notions, Sect. 4.2 induces the sufficient condition, and Sect. 4.3 proposes an approximation algorithm.

4.1 Definitions

To define the matrix, we define a *separable subpath* $Q \subseteq E$. A separable subpath $Q = \{e_j, e_{j+1}, \dots, e_{k-1}\}$ is a part of a

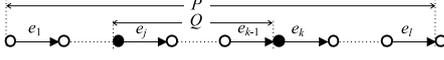


Fig. 2 A subpath Q of feasible path P . Counters, (e_j, P) and (e_k, P) , are placed at the black vertices.

feasible path $P = \{e_1, e_2, \dots, e_l\}$ between a pair of counters $\{(e_j, P), (e_k, P)\}$ (Fig. 2) and satisfies $(1 - \delta) < (1 - \epsilon)^{k-j}$. Inspecting the counter values $\{\kappa_j, \kappa_k\}$, we can tell the subpath contains an abnormal arc or not; The subpath contains an abnormal arc if and only if $\kappa_k / \kappa_j \leq 1 - \delta$. The condition $(1 - \delta) < (1 - \epsilon)^{k-j}$ is needed in order to make subpaths containing an abnormal arc separable. The minimum loss ratio of a subpath containing an abnormal arc is δ . Conversely, the maximum loss ratio of a subpath in which all arc is normal state is $1 - (1 - \epsilon)^{k-j}$. If $(1 - \epsilon)^{k-j} \leq (1 - \delta)$, we cannot judge whether the subpath contains an abnormal arc or not.

A *measurement matrix*, $A(C)$, is defined as follows (we sometimes omit the argument C if it is clear from the context). Let $Q = \{Q_1, Q_2, \dots, Q_m\}$ be a set of separable subpaths under a counter set C . Measurement matrix $A = (a_{ij})$ is defined as the $m \times n$ Boolean matrix of

$$a_{ij} = \begin{cases} 1 & e_j \in Q_i \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

The j -th column vector is denoted by \mathbf{a}_j ($j = 1, 2, \dots, n$).

A matrix A is *1-independent* if any column vector of A is different from each other and none of them equal to zero; i.e., $\forall j \neq j' : \mathbf{a}_j \neq \mathbf{a}_{j'}$ and $\forall j : \mathbf{a}_j \neq \mathbf{0}$.

Vectors \mathbf{x} and \mathbf{y} in the introduction are rigorously re-defined as follows. \mathbf{x} is an n -dimensional column vector whose j -th element indicates that j -th arc is an abnormal state. \mathbf{y} is a m -dimensional column vector whose i -th element indicates that subpath Q_i includes an abnormal arc.

4.2 Measurability Theorem and Estimation Algorithm

4.2.1 Theorem

When $\alpha = (1 - \epsilon)^{d-2}$, we can relate the sufficient conditions of α -measurable to the measurement matrix. Given the measurement matrix $A(C)$, sufficient conditions of measurability are described by the following theorem.

Theorem 1. *When τ_j is estimated by Algorithm 1, a counter set C is $(1 - \epsilon)^{d-2}$ -measurable, if the measurement matrix $A(C)$ is 1-independent and every feasible path P has at least one counter on it.*

In the following, we explain an algorithm that is used to estimate $\hat{\tau}$. Then, we prove its correctness together with the proof of Theorem 1.

For the sake of simplicity, we assume that the maximum length of feasible path is at least 3, $d \geq 3$, and every feasible path P has an ingress counter (e_1, P) to satisfy the second condition of the Theorem 1 where e_1 is the ingress arc of feasible path P . Also, we write the set of ingress counters as $C_0 = \{(e_1, P) : P \in \mathcal{P}\}$ and a set of additional

Algorithm 1: Transmitted Packets Estimation (Per-Path)

Input: Paths \mathcal{P} , Separable subpaths $\{Q_1, \dots, Q_m\}$, and counter values κ .
Output: Estimated number of packets $\hat{\tau}$ at every arc for each feasible path.

```

1 forall  $i \in \{1, \dots, m\}$  do
2   if  $Q_i$  contains an abnormal arc then  $y_i \leftarrow 1$  else  $y_i \leftarrow 0$ ;
3    $e_a \leftarrow \text{NULL}$ ;
4    $\phi \leftarrow 0$ ;
5   if  $y_i \neq 0$  then // Find an abnormal arc
6      $e_a \leftarrow$  Find an arc  $e_j$  such that  $\mathbf{y} = \mathbf{a}_j$ ;
7     Find a separable subpath  $Q = \{e_{j'}, \dots, e_{k'}\}$  that contains  $e_a$ ;
8      $\phi \leftarrow \kappa_{k'} / (\beta_{k'-j'-1} \kappa_{j'})$ ;
9 return { ESTIMATE( $P, e_a, \kappa$ ) :  $P \in \mathcal{P}$  };

```

counters as $X = C \setminus C_0$.

4.2.2 Algorithm

Algorithm 1 estimates the number $\hat{\tau}$ of transmitted packets for every arc and every path, given a per-path counter set C and their values κ , where β_j equals to $\sqrt{(1 - \epsilon)^j}$ for all integer j . The basic idea is as follows. If there is no abnormal arc between e_1 and e_{j-1} , for any arc e_j on a feasible path $P = \{e_1, \dots, e_l\}$, the number of transmitted packets τ_j falls between $(1 - \epsilon)^{j-1} \tau_1$ and τ_1 , $(1 - \epsilon)^{j-1} \tau_1 \leq \tau_j \leq \tau_1$, since a loss ratio of normal arc is less than or equal to ϵ and there are $j - 1$ arcs between e_1 and e_{j-1} . Therefore, we define the estimator $\hat{\tau}_j = \beta_{j-1} \kappa_1$ as the midpoint between $(1 - \epsilon)^{j-1} \kappa_1$ and κ_1 in a logarithmic scale since $\kappa_1 = \tau_1$ from the definition. Consequently, the estimated value $\hat{\tau}_j$ satisfies the condition $\beta_{j-1} \tau_j \leq \hat{\tau}_j \leq \tau_j / \beta_{j-1}$ for measurability. If there is an abnormal arc between e_1 and e_{j-1} , τ_j satisfies $(1 - \ell_a)(1 - \epsilon)^{j-2} \tau_1 \leq \tau_j \leq (1 - \ell_a) \tau_1$, since a loss ratio of normal arc is less than or equal to ϵ and there are $j - 2$ normal arcs between e_1 and e_{j-1} . By replacing $(1 - \ell_a)$ with the estimator ϕ of the pass ratio $(1 - \ell_a)$, we define the estimator $\hat{\tau}_j = \phi \beta_{j-2} \kappa_1$. Rigorous arguments, including the definition of ϕ are given in later in this section. The estimation algorithm that shown in Algorithm 1 runs as follows.

First, Algorithm 1 constructs the system of Boolean equations $A\mathbf{x} = \mathbf{y}$ from measurement results κ (Lines 1 to 2). Algorithm 1, then, solves the equations and identifies an abnormal arc e_a and estimates its loss ratio if it exists (Lines 3 to 8). Finally, the transmitted packets $\hat{\tau}_j$ are estimated through the subroutine ESTIMATE (Line 9 in Algorithm 1 and Algorithm 2). In the subroutine, the estimator $\hat{\tau}_j$ of loss ratio of arc e_j is $\beta_{j-1} \kappa_1$ if $j \leq a$, and $\phi \beta_{j-2} \kappa_1$ otherwise. Correctness of Algorithm 1 and 2 is provided in the following along with the proof of the Theorem 1.

4.2.3 Proof

We show that if $A(C)$ is 1-independent then the output of the Algorithm 1, $\hat{\tau}$, satisfies

$$\forall P \in \mathcal{P}, \forall e_j \in P : \beta_{2(d-2)} \cdot \tau_j \leq \hat{\tau}_j \leq \frac{1}{\beta_{2(d-2)}} \cdot \tau_j. \quad (5)$$

Algorithm 2: ESTIMATE (Per-Path)

Input: A feasible path $P = \{e_1, \dots, e_l\}$, an abnormal arc e_a , and counter values.

Output: Estimated number of packets $\hat{\tau}_j$ at each arc e_j on the path P .

```

1  $T \leftarrow \emptyset$ ;
2 forall  $j \in \{1, \dots, l\}$  do
3   if  $e_a \notin P \vee j \leq a$  then
4      $\hat{\tau}_j \leftarrow \beta_{j-1}\kappa_1$ ;
5   else
6      $\hat{\tau}_j \leftarrow \phi\beta_{j-2}\kappa_1$ ;
7    $T \leftarrow T \cup \{\hat{\tau}_j\}$ ;
8 return  $T$ ;
```

This proves the correctness of Algorithm 1 and Theorem 1 since $(1 - \epsilon)^{d-2} = \beta_{2(d-2)}$.

First, we show that the Algorithm 1 identifies an abnormal arc if it exists. Next, we evaluate accuracy of estimated value $\hat{\tau}$.

The following Lemma 1 shows that the abnormal arc is uniquely determined at Line 6 in Algorithm 1. Note that \vee denotes Boolean OR operator and Boolean OR operator is used instead of the ordinary sum in Boolean equations.

Lemma 1. *Let x_j be a binary variable ($j = 1, \dots, n$). Then the system of Boolean equations $A\mathbf{x} = \mathbf{y}$,*

$$a_{i1}x_1 \vee a_{i2}x_2 \vee \dots \vee a_{in}x_n = y_i \quad (i = 1, 2, \dots, m), \quad (6)$$

have unique solution \mathbf{x}^* among to $\|\mathbf{x}^*\| \leq 1$ and $x_j^* = 1$ if and only if the arc e_j is abnormal.

Proof. The mapping $\mathbf{x} \mapsto A\mathbf{x}$ is injective if the domain is $\{\mathbf{x} \in \{0, 1\}^n : \|\mathbf{x}\| \leq 1\}$. This is because the measurement matrix A is 1-independent. \square

Note that if the j -th arc is abnormal, $x_j^* = 1$, then $A\mathbf{x}^* = \mathbf{a}_j$. Hence, Algorithm 1 can identify the abnormal arc at Line 6. As we mentioned in Section 3.1, we assume that every arc has either a normal state $\ell_j \leq \epsilon$ or an abnormal state $\ell_j \geq \delta$, i.e., there is no arc whose loss ratio is $\epsilon < \ell_j < \delta$. If loss ratios of some arcs are $\epsilon < \ell_j < \delta$, it is difficult to judge that a subpath contains an abnormal arc, since a subpath may not contain an abnormal arc even if $\kappa_k/\kappa_j \leq 1 - \delta$. Hence, (6) cannot be solved or the solution does not corresponds to the abnormal arc due to unreliable y_i .

Lemma 2 and 4 evaluate accuracy of the transmitted packets on normal and abnormal paths.

Lemma 2. *Let $P = \{e_1, e_2, \dots, e_l\}$ be a feasible path that does not include abnormal arcs. Then*

$$\beta_{j-1}\tau_j \leq \hat{\tau}_j \leq \frac{1}{\beta_{j-1}}\tau_j \quad (j = 1, 2, \dots, l), \quad (7)$$

where $\hat{\tau}_j = \beta_{j-1}\kappa_1$ is the estimator of loss ratio of arc e_j .

Proof. This is because $(1 - \epsilon)^{j-1} \cdot \kappa_1 \leq \tau_j \leq \kappa_1$ for all $j = 1, 2, \dots, l$. \square

Lemma 3. *Let ℓ_a and $Q = \{e_j, \dots, e_{k'-1}\}$ be the loss ratio of*

the abnormal arc e_a and a separable subpath that contains e_a , respectively. Then

$$\beta_{k'-j-1} \cdot (1 - \ell_a) \leq \phi \leq \frac{1}{\beta_{k'-j-1}} \cdot (1 - \ell_a), \quad (8)$$

where $\phi = \kappa_{k'} / (\beta_{k'-j-1}\kappa_j)$ is an estimator of the pass ratio $1 - \ell_a$ on the abnormal arc.

Proof. This is because the separable subpath $Q = \{e_j, \dots, e_{k'-1}\}$ contains e_a and $(1 - \epsilon)^{k'-j-1} \cdot (1 - \ell_a) \cdot \kappa_j \leq \kappa_{k'} \leq (1 - \ell_a) \cdot \kappa_j$ since $\kappa_{k'} = \tau_{k'}$. \square

Lemma 4. *Let $P = \{e_1, e_2, \dots, e_l\}$ be a feasible path that contains an abnormal arc, e_a ($1 \leq a \leq l$). And let $Q = \{e_j, \dots, e_{k'-1}\}$ be a separable subpath that contains the abnormal arc. Then*

$$\beta_{k'-j'+j-3} \cdot \tau_j \leq \hat{\tau}_j \leq \frac{1}{\beta_{k'-j'+j-3}} \cdot \tau_j \quad (j = 1, 2, \dots, l), \quad (9)$$

where the estimator $\hat{\tau}_j$ of loss ratio of arc e_j is $\beta_{j-1}\kappa_1$ if $j \leq a$, and $\phi\beta_{j-2}\kappa_1$ otherwise.

Proof. If $j \leq a$ then use Lemma 2. Otherwise, $j > a$, use the next Lemma 3 and the fact that $(1 - \epsilon)^{j-2}(1 - \ell_a) \cdot \kappa_1 \leq \tau_j \leq (1 - \ell_a)\kappa_1$ where ℓ_a is the loss ratio at the abnormal arc e_a . \square

Hence, we have (5) because in Lemma 2 we have $j - 1 \leq d - 1 \leq 2(d - 2)$ from $d \geq 3$, and in Lemma 4 we have $k' - j' + j - 3 \leq 2(d - 2)$ from $1 \leq j, j', k' \leq d$.

4.3 Counter Set Minimization Algorithm

To describe the counter placement algorithm, we reformulate our problem (2) based on the Theorem 1 as follows

$$\min_{\mathbf{x} \in \tilde{\mathcal{C}}} \left\{ |\mathbf{X}| : g(\mathbf{X}) = g(\tilde{\mathcal{C}}) \right\}, \quad (10)$$

where $\tilde{\mathcal{C}}$ is the set of per-path counters, $E \times \mathcal{P}$, and g is a function that represents how close the measurement matrix $A(C_0 \cup X)$ is to 1-independent. The function g is defined as

$$g(\mathbf{X}) = \left| \{(j, k) : 0 \leq j < k \leq n, \mathbf{a}_j \neq \mathbf{a}_k\} \right|, \quad (11)$$

where \mathbf{a}_j is j -th column vector of the measurement matrix $A(C_0 \cup X)$ and \mathbf{a}_0 is the zero vector $\mathbf{0}$. We call this function g coverage function. Note that the coverage function $g(\mathbf{X})$ equals to $\binom{n+1}{2}$ if and only if the matrix $A(C_0 \cup X)$ is 1-independent. We leverage submodularity of this coverage function g for algorithm design.

In this subsection, we give a constructive proof of Theorem 2 by using Algorithm 3.

Theorem 2. *There exists a polynomial-time approximation algorithm of (10) and its approximation ratio is*

$$\left(1 + \frac{d-1}{\log_{1-\epsilon}(1-\delta)-1} \right) H\left(\frac{n(n+1)}{2} \right), \quad (12)$$

Algorithm 3: Counter Placement (Per-Path)

Input: A set of feasible paths \mathcal{P} and loss thresholds ϵ, δ .
Output: A set of additional counters X_{ALG} that satisfies $g(X_{\text{ALG}}) = g(\tilde{C})$.

- 1 $X_0 \leftarrow \emptyset$;
- 2 **while** $\exists x \in \tilde{C} : g_0(x|X_0) > 0$ **do**
- 3 $x^* \leftarrow \arg \max_{x \in \tilde{C}} g_0(x|X_0)$;
- 4 $X_0 \leftarrow X_0 \cup \{x^*\}$;
- 5 $Q' \leftarrow$ Irreducible separable subpaths of X_0 at $\epsilon = 0$;
- 6 **forall** $Q \in Q'$ **do** $X_{\text{ALG}} \leftarrow \text{SEPARATE}(Q, X_0, \epsilon, \delta)$;
- 7 **return** X_{ALG} ;

Algorithm 4: SEPARATE (Per-Path)

Input: A subpath $Q = \{e_j, \dots, e_{k-1}\} \subseteq P$, a counter set X_0 , and loss thresholds ϵ, δ .
Output: A counter set X_{ALG} that divides the subpath Q into separable subpaths.

- 1 $k' \leftarrow j - 1 + \lceil \log_{1-\epsilon}(1 - \delta) \rceil$;
- 2 $X_{\text{ALG}} \leftarrow X_0$;
- 3 **if** $k' < k$ **then**
- 4 $X_{\text{ALG}} \leftarrow X_{\text{ALG}} \cup \{(e_{k'}, P)\}$;
- 5 $Q \leftarrow \{e_{k'}, \dots, e_{k-1}\}$;
- 6 $X_{\text{ALG}} \leftarrow \text{SEPARATE}(Q, X_{\text{ALG}}, \epsilon, \delta)$;
- 7 **return** X_{ALG} ;

where $H(x)$ is x -th harmonic number $\sum_{i=1}^x 1/i$.

In the following Sect. 4.3.1 introduces some notations, Sect. 4.3.2 explains the algorithm, and Sect. 4.3.3 proves the Theorem 2.

4.3.1 Notations

Here we introduce some notations to explain Algorithms 3 and 4.

A function $g(x|X)$ represents the gain of counter $x \in \tilde{C}$ added to X : $g(x|X) = g(X \cup \{x\}) - g(X)$. The function g_0 is the coverage function for $\epsilon = 0$, and we have $g_0(x|X) = g_0(X \cup \{x\}) - g_0(X)$.

An *irreducible* separable subpath $Q = \{e_j, \dots, e_{k-1}\} \subseteq P$ of a counter set $C_0 \cup X$ is a separable subpath of $C_0 \cup X$ and it does not have any counter in the inside it: $\forall e_{j'} \in Q \setminus \{e_j, e_{k-1}\} = \{e_{j+1}, \dots, e_{k-2}\} : (e_{j'}, P) \notin X$. Intuitively, if a subpath is separable but not irreducible, then it is unnecessary to identify abnormal arcs. A separable but not irreducible subpath does not contribute to make a measurement matrix 1-independent since $a_{i_1 j} \neq a_{i_1 j'} \Rightarrow a_{i_2 j} \neq a_{i_2 j'} \vee a_{i_2 j} \neq a_{i_2 j'}$ for all j and j' , where Q_{i_1} is a separable but not irreducible subpath that includes two irreducible subpaths Q_{i_2} and Q_{i_3} .

An *irreducible* measurement matrix of a counter set $C_0 \cup X$ is the matrix constructed from the irreducible separable subpaths of $C_0 \cup X$ along with the definition (4). Note that the measurement matrix is 1-independent if and only if the irreducible one is 1-independent.

Let X_0 be the intermediate counter set assuming $\epsilon = 0$ at Line 5 in Algorithm 3, X_{ALG} be the output of Algorithm 3, X^* be the optimal counter set of (10) for given thresholds ϵ and δ . We define a feasible counter set as a counter set that makes a measurement matrix 1-independent. Both X_0 and

X_{ALG} are feasible counter sets.

4.3.2 Algorithm

To approximately solve the problem (10), we solve the problem for $\epsilon = 0$ and convert its solution X_0 into a solution X_{ALG} for given ϵ and δ . This is because, when the lower threshold ϵ equals to 0, the coverage function g_0 becomes a submodular function (Proof sketch in Appendix A.1) and a simple greedy algorithm yields an approximate solution with a theoretical guarantee in polynomial-time [21]. Also, the counter set conversion runs in linear time and the counter increase is theoretically bounded. Note that the solution X_{ALG} for given ϵ and δ can differ from the solution X_0 for $\epsilon = 0$ since the measurement matrix $A(X)$ can be changed depending on ϵ and δ . The measurement matrix is defined by separable subpaths under the given counters, and whether or not a subpath is separable depends on ϵ and δ . If $\epsilon = 0$, all subpaths are separable from the definition. Conversely, for example, subpaths with 3 or more arcs are not separable when $\epsilon = 0.1$ and $\delta = 0.2$.

First, from Line 1 to Line 4, Algorithm 3 finds a feasible counter set for $\epsilon = 0$. In the while loop, we can find the counter x^* that maximizes the gain by calculating the gain $g(x|X)$ for all counters $x = (e, P) \in E \times \mathcal{P} \setminus X$. After the while loop, the counter set X_0 is a feasible counter set for $\epsilon = 0$: $g_0(X_0) = g_0(\tilde{C})$ (Lemma 6).

Then, Algorithm 3 converts the feasible counter set X_0 into a feasible one for given thresholds ϵ and δ by adding counters through the subroutine SEPARATE (Lines 5, 6, and Algorithm 4). Every time the subroutine adds a counter $(e_{k'}, P)$, the subpath $Q = \{e_j, \dots, e_{k-1}\}$ is divided into two subpaths $Q_1 = \{e_j, \dots, e_{k'-1}\}$ and $Q_2 = \{e_{k'}, \dots, e_{k-1}\}$. Line 1 identifies the subscript k' of the arc at which a counter should be added. $-1 + \lceil \log_{1-\epsilon}(1 - \delta) \rceil$ means the maximum integer d' that satisfies $1 - \delta < (1 - \epsilon)^{d'}$ (i.e. d' is the maximum length for a subpath to be separable). Note that, now the former subpath Q_1 is separable for given thresholds ϵ and δ . As a result, every irreducible subpath Q for $\epsilon = 0$ is divided into irreducible subpaths Q_1, Q_2, \dots, Q_l for given thresholds ϵ and δ . After Line 6, the counter set X_{ALG} is a feasible counter set: $g(X_{\text{ALG}}) = g(\tilde{C})$ (Lemma 5).

4.3.3 Proof

First, we prove the correctness of the Algorithm 3, i.e., the output counter set X_{ALG} satisfies $g(X_{\text{ALG}}) = g(\tilde{C})$. Then, we prove the Theorem 2 by evaluating the approximation ratio of Algorithm 3. Note that it is trivial to show that Algorithm 3 runs in polynomial time for $|E|$ and $|\mathcal{P}|$.

To show the correctness, it is sufficient to prove the following two lemmas.

Lemma 5. *If the intermediate counter set X_0 satisfies $g_0(X_0) = g_0(\tilde{C})$ then the output counter set X_{ALG} satisfies $g(X_{\text{ALG}}) = g(\tilde{C})$.*

Lemma 6. *The intermediate counter set X_0 satisfies*

$$g_0(X_0) = g_0(\tilde{C}).$$

Proof of Lemma 5. We will prove by contradiction. Let $A = (a_{ij})$ be the measurement matrix of X_{ALG} for given ϵ and δ , $A' = (a'_{ij})$ be the irreducible measurement matrix of X_0 for $\epsilon = 0$. As shown in Algorithm 3, the subroutine SEPARATE shown in Algorithm 4 computes the difference between X_{ALG} and X_0 . The irreducible separable subpath of X_0 for $\epsilon = 0$ is divided into separable subpaths for given ϵ and δ by X_{ALG} . Suppose that $g(X_{\text{ALG}}) < g_0(X_0)$. Then, there exist two arc e_j and e_k such that $\mathbf{a}_j = \mathbf{a}_k$ and $\mathbf{a}'_j \neq \mathbf{a}'_k$. Hence, there exists an irreducible separable subpath Q'_i for $\epsilon = 0$ at X_0 such that $\mathbf{a}'_{ij} \neq \mathbf{a}'_{ik}$. This subpath Q'_i must be divided into some subpaths Q_{i1}, \dots, Q_{il} through the subroutine SEPARATE. Thus, we have

$$\mathbf{a}'_{ij} = a_{i1j} \vee a_{i2j} \vee \dots \vee a_{ilj}, \quad (13)$$

$$\mathbf{a}'_{ik} = a_{i1k} \vee a_{i2k} \vee \dots \vee a_{ilk}, \quad (14)$$

since all arcs in subpath Q_i are included in either of Q_{i1}, \dots, Q_{il} . The right-hand sides of (13) and (14) must not be equal since $\mathbf{a}'_{ij} \neq \mathbf{a}'_{ik}$. However, this contradicts $\mathbf{a}_j = \mathbf{a}_k$ because the right-hand sides of (13) and (14) are equal if $\mathbf{a}_j = \mathbf{a}_k$. \square

Proof of Lemma 6. The problem of (10) for $\epsilon = 0$ is called the submodular cover [21], [22] because the function g_0 is monotone submodular for $\epsilon = 0$ (Sketch of proof in Appendix). The previous research [21] shows that the greedy algorithm from Lines 1 to 4 in Algorithm 3 finds a feasible solution and its approximation ratio is $H(\max_{x \in \tilde{C}} g_0(\{x\}))$. \square

To prove the approximation ratio, it is sufficient to prove the following two lemmas.

Lemma 7. For given ϵ , δ , and d , the following inequality holds.

$$|X_{\text{ALG}}| \leq \left(1 + \frac{d-1}{\log_{1-\epsilon}(1-\delta)-1}\right) |X_0|. \quad (15)$$

Lemma 8. For given n , the following inequality holds.

$$|X_0| \leq H\left(\frac{n(n+1)}{2}\right) |X^*|. \quad (16)$$

Proof of Lemma 7. We have $|Q'| = |X_0|$ since there is a one-to-one relation between the last arc e_{k-1} of an irreducible separable subpath $Q \subseteq P$ and a counter $(e_k, P) \in X_0$. For each irreducible separable subpath $Q \in Q'$, SEPARATE adds at most $\frac{d-1}{\log_{1-\epsilon}(1-\delta)-1}$ counters to X_0 . Hence, we have

$$|X_{\text{ALG}}| \leq |X_0| + \frac{d-1}{\log_{1-\epsilon}(1-\delta)-1} |Q'| \quad (17)$$

$$\leq \left(1 + \frac{d-1}{\log_{1-\epsilon}(1-\delta)-1}\right) |X_0|. \quad (18)$$

Proof of Lemma 8. From the proof of Lemma 6, we have

$$|X_0| \leq H(\max_{x \in \tilde{C}} g_0(\{x\})) |X_0^*| \leq H\left(\frac{n(n+1)}{2}\right) |X_0^*| \quad (19)$$

where X_0^* is an optimal solution of (10) for $\epsilon = 0$. Also, we have $|X_0^*| \leq |X^*|$ since X^* is also a feasible counter set for $\epsilon = 0$. This is because every irreducible separable subpath Q of X^* for given ϵ and δ is also separable for $\epsilon = 0$. \square

5. Path-Set Measurement

This subsection briefly discusses how to extend Sect. 4 for “path-set” measurement, in order to reduce the number of counters. Figure 3 shows a tree that is the union of l paths; the ingress arcs, e_1, e_2, \dots, e_l , are their own but the egress arc e_0 is common. If the l paths were observed individually, we would require $2l$ counters, but the tree has only $l+1$ counters because the egress counter covers the l paths. This implies that a counter is defined as a pair of an arc and a set of feasible paths, i.e., $C \subseteq E \times 2^{\mathcal{P}}$.

Due to lack of space, we only present the overview of the theorems in an informal manner. The measurement matrix $A(C)$ is also defined for path-set counter C in a similar manner. This matrix induces also sufficient condition of measurability:

Theorem 3. When τ_j is estimated by a similar algorithm with Algorithm 1, a counter set C is weakly α -measurable, if the measurement matrix $A(C)$ is l -independent, where the constant α is defined as[†]

$$\alpha = \sqrt{\zeta \left\{ \frac{1+\zeta}{2} - \sqrt{\left(\frac{1-\zeta}{2}\right)^2 + \delta - \epsilon} \right\}}. \quad (20)$$

where ζ equals to $(1-\epsilon)^{d-2}$.

Here weakly α -measurable is a relaxed condition of α -measurable. The counter set C is weakly α -measurable if,

$$\forall P \in \mathcal{P}, \forall e_j \in P, \forall \tau_j : \quad \alpha \cdot \tau_j \leq \hat{\tau}_j \leq \frac{1}{\alpha} \cdot \tau_j \quad \text{or,}$$

$$\alpha \cdot \lambda_j \leq \hat{\lambda}_j \leq \frac{1}{\alpha} \cdot \lambda_j,$$

where λ_j is the number of packets dropped between e_1 and e_{j-1} , i.e., $\lambda_j = \tau_1 - \tau_j$. Again, we reformulate our problem

[†]The last term $\delta - \epsilon$ can be considered as the overhead of path-set measurement since if we remove the last term then we obtain, $\alpha = \zeta = (1-\epsilon)^{d-2}$; the same results of Theorem 1.

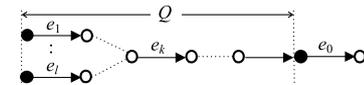


Fig. 3 A subtree Q . The root arc is e_0 while the leaf arcs are e_j ($j = 1, 2, \dots, l$); the root is not included in Q . Counters, (e_j, \mathcal{P}_j) 's, are placed at the black vertices. \square

Algorithm 5: Counter Placement (Path-Set)

Input: A set of feasible paths \mathcal{P} and loss thresholds ϵ, δ , counter values κ , and a real number ρ .
Output: A counter set X_{ALG} that satisfies $g(X_{\text{ALG}}) = g(\tilde{C})$.

- 1 $X_0 \leftarrow \emptyset$;
- 2 **while** $\exists x \in \tilde{C} : g_0(x|X_0) > 0$ **do**
- 3 $\tilde{x} \leftarrow$ Find x such that
 $((1/3) - \rho) \max_{x^* \in \tilde{C}} g_0(x^*|X_0) \leq g_0(x|X_0)$;
- 4 $X_0 \leftarrow X_0 \cup \{\tilde{x}\}$;
- 5 $Q' \leftarrow$ Irreducible separable subtrees of X_0 at $\epsilon = 0$;
- 6 **forall** $Q \in Q'$ **do** $X_{\text{ALG}} \leftarrow \text{SEPARATE}(Q, X_0, \epsilon, \delta, \kappa)$;
- 7 **return** X_{ALG}

Algorithm 6: SEPARATE (Path-Set)

Input: A subtree $Q = \{e_j, \dots, e_{k-1}\} \subseteq \bigcup \mathcal{P}_0$, a counter set X_0 , loss thresholds ϵ, δ , and counter values κ .
Output: A counter set X_{ALG} that divides the subtree Q into separable subtrees.

- 1 $X_{\text{ALG}} \leftarrow X_0$;
- 2 **if** Q is separable **then return** X_{ALG} ; // Do nothing
- 3 $e_* \leftarrow \arg \min_{e_k \in Q} \tilde{h}(e_k)$;
- 4 $\mathcal{P}_* \leftarrow \arg \min_{\mathcal{P}_j \in \mathcal{P}_0} \tilde{h}(\mathcal{P}_j)$;
- 5 **if** $\tilde{h}(e_*) \leq \tilde{h}(\mathcal{P}_*)$ **then** // Divide at e_*
- 6 $X_{\text{ALG}} \leftarrow X_{\text{ALG}} \cup (e_*, \bigcup_{e_s \in \mathcal{P}_j} \mathcal{P}_j)$;
- 7 $(Q_1, Q_2) \leftarrow (U_{e_*}, Q \setminus U_{e_*})$;
- 8 **else** // Peel off \mathcal{P}_*
- 9 $X_{\text{ALG}} \leftarrow X_{\text{ALG}} \setminus \{(e_0, \mathcal{P}_0)\} \cup \{(e_0, \mathcal{P}_*), (e_0, \mathcal{P}_*^C)\}$;
- 10 $(Q_1, Q_2) \leftarrow (Q \cap \mathcal{P}_*, Q \cap \mathcal{P}_*^C)$;
- 11 $X_{\text{ALG}} \leftarrow \text{SEPARATE}(Q_1, X_{\text{ALG}}, \epsilon, \delta, \kappa)$;
- 12 $X_{\text{ALG}} \leftarrow \text{SEPARATE}(Q_2, X_{\text{ALG}}, \epsilon, \delta, \kappa)$;
- 13 **return** X_{ALG}

as follows,

$$\min_{C \subseteq E \times \mathcal{P}} \{|C| : A(C) \text{ is 1-independent}\}. \quad (21)$$

Due to the space limit, we do not describe the path-set version of Algorithms 1 and 2, which are used to estimate traffic volumes based on packet counts.

The remaining of this section, we explain the differences between path-set algorithm shown in Algorithms 5 and 6 and per-path one.

The algorithm approximately finds the maximum gain $\arg \max g_0(x|X_0)$ since calculating all the gain $g_0(x|X_0)$ for all $x \in \tilde{C} \setminus X_0$ takes exponential time (Compare Line 3 in Algorithms 3 and 5). This is because the number of candidate counters roughly equals to $|E|2^{|\mathcal{P}|}$. The following Theorem provides $((1/3) - \rho)$ -approximation algorithm by utilizing a submodular maximization algorithm [23] (We omit the proof due to lack of space).

Theorem 4. *There exists a polynomial-time $((1/3) - \rho)$ -approximation algorithm for the problem $\max_{x \in \tilde{C}} g_0(x|X_0)$.*

We explain differences between subroutine for path-set (Algorithm 6) and per-path (Algorithm 4). Algorithm 6 divides a subtree Q and/or peels off a path from it to make

it separable for given ϵ and δ . Algorithm 6 adds a counter $(e_k, \bigcup_{e_k \in \mathcal{P}_j} \mathcal{P}_j)$ inside the subtree and the subtree is divided into an upper part U_{e_k} and a lower part $Q \setminus U_{e_k}$ (Lines 6 and 7). Here U_{e_k} is the upper side of the subtree that obtained by dividing the subtree Q at the arc e_k (e_k is excluded from U_{e_k}). Or, Algorithm 6 replaces the counter on the root arc (e_0, \mathcal{P}_0) with two counters $\{(e_0, \mathcal{P}_j), (e_0, \mathcal{P}_j^C)\}$, where \mathcal{P}_j^C is the path set of Q other than \mathcal{P}_j : $\mathcal{P}_j^C = \bigcup_{j'=1: j' \neq j}^l \mathcal{P}_{j'}$ (Line 9). This peels the subpath $Q \cap \mathcal{P}_j$ from the subtree Q and we have new subpath $Q \cap \mathcal{P}_j$ and subtree $Q \cap \mathcal{P}_j^C$ (Line 10). Algorithm 6 selects counter addition or replacement based on a greedy strategy, which minimizes the deviation of the separable condition for path-set (Line 3 to 5). This deviation is measured by the following function h :

$$h(Q) = \max \left(0, \sum_{j=1}^l (1 - (1 - \epsilon)^{d_j}) \kappa_j - \delta \min_j \kappa_j \right). \quad (22)$$

Note that $h(Q) = 0$ if and only if Q is separable. Hence, we define the deviation of dividing at e_k as $\tilde{h}(e_k) = \max(h(U_{e_k}), h(Q \setminus U_{e_k}))$ and that of peeling off \mathcal{P}_j as $\tilde{h}(\mathcal{P}_j) = \max(h(Q \cap \mathcal{P}_j), h(Q \cap \mathcal{P}_j^C))$.

6. Practical Modeling Issues

This section describes how to model a network that consists of actual switches. We begin with a switch model. We assume that each switch consists of a forwarding element and several ports. A forwarding element includes a forwarding information base (FIB) and a switching bus; the FIB determines an output port through which a packet transmitted, while the switching bus transfers the packet to the output port. Each port has a packet queue, an access control list (ACL), and counters. Ports are categorized to input ports and output ports. A pair of input and output ports at different switches is connected by a point-to-point link.

A FIB and ACL is a set of rules. Each rule associates a packet filter with an action; a FIB action is one of output ports in the switch, while an ACL action is “permit” or “discard”. A path along which a packet is transferred is determined by rules matched to the packet, and so we can calculate all the feasible paths in a network using the FIBs and ACLs [24], [25]. Counters are associated with a set of feasible paths in our method, so packets have to be classified based on the paths. Packet classification was traditionally designed only for actions at a single switch, but recently new techniques has been developed to classify packets based on a combination of switch actions, i.e., a path, at the rate of 10 Gbps links [15], [16]; this advancement has made our method feasible. We only consider paths that are not terminated by ACLs, since we are interested in transit traffic for the TM estimation.

In our graph model introduced in Sect. 3, a FIB and ACL is represented by a tail of arc, because a path is switched by FIBs and packets are counted around ACLs. Figure 4 shows an example. For an input port, we assume

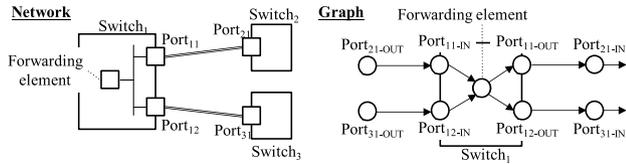


Fig. 4 A part of network and the corresponding graph model.

that the tail includes the ACL and counters while the arc is the bus to the FIB. For a forwarding element, the vertex is the FIB while each arc represents a bus to the corresponding output port. For an output port, the tail is the ACL and counters, while the arc includes the output queue, the link, and the opposed input queue.

Since no counter is usually placed at forwarding elements in ordinary switches, we assume that tails of forwarding elements cannot have counters. This implies that we might not be able to make a network measurable due to the lack of counters. The packet loss ratios of indistinguishable arcs should be estimated conservatively in Algorithm 1; take the worst case as if either arc were determined as abnormal. Note that this cannot be an issue in OpenFlow networks, because OpenFlow switches are required to return packet counts for every flow entry.

7. Experiments

We perform experiments to evaluate our approach using three configuration datasets and one traffic dataset. The configuration datasets are obtained from Internet2[†], Stanford backbone network [26], and Purdue campus network [27], while the traffic data comes from Internet2^{††}. The configuration datasets include a network topology and FIBs at every switch. Actual ACLs are used only for Stanford. All transit paths are extracted from the datasets and are used as feasible paths. The paths for Stanford including ACLs are extracted according to the model described in Sect. 6. The statistics are given in Table 2 (arcs on no transit path are ignored). In the traffic dataset, traffic volumes in byte for 144 paths were measured every 5 min for 7 days. The average, the variance, and the maximum/minimum ratio are 9.47×10^8 , 2.32×10^{18} , and 1.89×10^5 , respectively. Traffic volume in the traffic dataset is randomly mapped to a path in a configuration dataset. Since the traffic data is used to determine packet counts κ , the traffic volume in byte is translated into the traffic volume in packets, by assuming the average packet length equals for all paths.

7.1 Per-Path Measurement

7.1.1 Size of Counter Set

We first calculate the optimal counter set, $C_0 \cup X$, for per-path measurement and examine indistinguishable arc-pairs

[†]<http://vn.gnoc.iu.edu/Internet2/fib>

^{††}<http://www.cs.utexas.edu/~yzhang/research/AbileneTM/>

Table 2 Number of elements in the configuration datasets.

	Internet2	Stanford	Purdue
Switches	9	16	1,503
Ports used	140	58	526
Feasible paths $ \mathcal{P} $	11,159	331	2,985
Maximum path length d	5	9	5
Arcs with FIB/ACL $ E $	140	132	526

Table 3 Number of counters and arc-pairs for $\epsilon = 0$.

	Internet2	Stanford	Purdue
Initial counters $ C_0 $	11,159	331	2,985
Additional counters $ X $	116	65	380
Indistinguishable arc-pairs / all arc-pairs	0 / 9,730	49 / 8,646	0 / 138,075

Table 4 Number of matrix rows for the Boolean algebra approach.

	Internet2	Stanford	Purdue
Number of matrix rows	115	66	380

for $\epsilon = 0$; an arc pair $(e_i, e_j) \in E^2$ is indistinguishable if the corresponding column vectors $\mathbf{a}_i, \mathbf{a}_j$ are equal. As shown in Table 3, some arc-pairs are indistinguishable only for Stanford due to the limitation noted in Sect. 6, but they are only successive FIB and ACL in a same switch, and the arc-pairs are mutually exclusive. The number of additional counters, which is shown in Table 3, corresponds to the number of rows in the measurement matrix. The number of rows is also calculated for the Boolean algebra approach [14], and the results are shown in Table 4. The number of additional counters $|X|$ in Table 3 and the number of matrix rows in Table 4 are almost the same value. The results imply that our approach almost agrees with the lower bound provided by the Boolean algebra approach; our approach is better than the bound for Stanford due to the approximation of optimization algorithms.

Next, we verify the number of matrix rows when $\epsilon \neq 0$. In this case, the number of matrix rows depends on threshold δ ; Fig. 5 shows the number of matrix rows for $\epsilon = 10^{-5}$ and $10^{-5} \leq \delta \leq 10^{-1}$. The green horizontal line shows the lower bound, i.e., the number of rows for the Boolean algebra approach [14]. Our approach almost converges to the lower bound for $\delta > 5.0 \times 10^{-5}$, and the number at $\delta = 10^{-1}$ is coincident with that for $\epsilon = 0$ and $\delta = 1$. The Boolean algebra approach [14] assumes $\epsilon \ll \delta$ to clearly distinguish abnormal links, but the results show the assumption is unnecessary even for the traffic volume estimation, which is far beyond the Boolean algebra approach. Note that, for $\delta \approx \epsilon$ in Stanford, measurement matrices cannot be obtained since separable subpaths are not able to be composed due to the restrictions mentioned in Sect. 6.

7.1.2 Error Bounds

Theorem 1 shows that our approach is $(1 - \epsilon)^{d-2}$ -measurable for per-path measurement. For $\epsilon = 10^{-5}$ and d shown in Table 2, $(1 - \epsilon)^{d-2} \geq 0.99993$. The error bounds on a estimation of traffic volumes are only $\pm 0.01\%$. Our approach can accu-

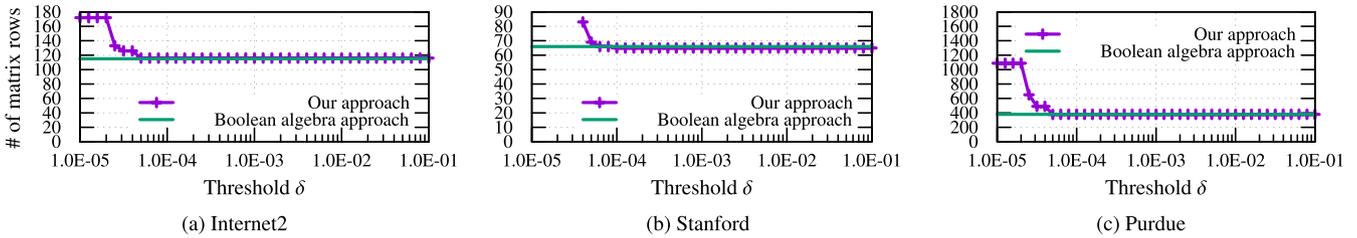


Fig. 5 The number of rows in a measurement matrix.

Table 5 Additional counters $|X|$ for per-path and path-set measurements.

	Internet2	Stanford	Purdue
Additional counters $ X $ (per-path)	116	65	380
Additional counters $ X $ (path-set)	116	67	359

rately estimate traffic volumes, since the accuracy is highly guaranteed with these error bounds.

7.2 Path-Set Measurement

The size of counter set is also evaluated for path-set measurement; Table 5 gives the size for $\epsilon = 0$ and $\delta = 1$. Purdue gets the largest gain, because it has many large trees like Fig. 3. The results for Internet2 and Stanford are similar to the results in Table 3 though $|X|$ for Stanford slightly increases due to the approximation algorithm. Internet2 has the largest number of feasible paths but they cannot be merged into trees due to the loss of measurability, while Stanford does not have feasible paths enough to reduce the counters.

The time to spend for the path-set version of Algorithm 4 is shorter than 10 sec using a PC with 3.4 GHz Intel Core i5 processor. The algorithm can be done in an online manner following a change of traffic volume in 10 sec time scale or longer.

8. Related Work

To estimate the traffic volumes, conventional TM estimation methods [5]–[11], [28] rely on link traffic measured at edge switches without breaking the traffic volume to paths or flows. The estimation is performed with EM algorithm [5], entropy maximization [6], neural networks [7], and compressive sensing [8], [9]. Some methods are compared in [10], [11]. Computational costs [28] and monitoring errors [9] are studied. To the best of our knowledge, no literature addresses packet drops in the context of TM estimation.

A rich collection of fault localization techniques in networks has been developed. Network tomography localizes failed links usually using active probes [29]–[33]. Since active probes disrupt production traffic, their use is limited to off-peak hours. Passive approach to the network tomography has been also studied [14], [34]–[36]. References [34]–[36] depend on TCP signaling packets (e.g., TCP-ACK), they cannot be applied to UDP traffic. In addition to the network tomography, machine learning approaches have been

also proposed in order to diagnose fault cause [37]. There has been no method designed to estimate the amount of lost packets, while the conventional tomography techniques report loss ratio of packets.

Optimization algorithms regarding a submodular function are well-studied in combinatorial optimization [20]. Approximation algorithms for the submodular cover problem has been proposed [21], and it is recently generalized to a monotone submodular cost function [22]. A constant-factor approximation for maximizing nonnegative submodular functions has been achieved in reference [23].

9. Conclusion

This paper proposed a TM estimation method that handles packet dropping in a network. With the solid theory about the measurability based on Boolean matrices, we developed an optimization algorithm for the minimum counter set. Experiments in real datasets showed its strong performance. We believe that this paper gives a new viewpoint to network measurement and opens a novel research field linking the TM estimation and the fault localization. This research field embraces mathematical interests as well as practical importance.

We will extend our method to consider the constraint of SRAM size. We also will investigate the impact of traffic fluctuation and packet sampling on the accuracy of our method. This paper assumed path volumes are directly counted at ingress switches, but they could be estimated by TM estimation techniques; this approach gives us opportunities to reduce counters further, though estimation errors might not be bounded.

Acknowledgments

This work was partly supported by the Japan Prize Foundation, and JSPS KAKENHI Grant Number JP17K00008 and JP18K18035.

References

- [1] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol.20, no.4, pp.756–767, 2002.
- [2] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," *ACM IMC*, pp.248–258, 2003.
- [3] R.S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, Morgan Kaufmann, 1998.

- [4] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, “Network anomography,” ACM IMC, pp.317–330, 2005.
- [5] Y. Vardi, “Network tomography: Estimating source-destination traffic intensities from link data,” J. Am. Stat. Assoc., vol.91, no.433, pp.365–377, 1996.
- [6] Y. Zhang, M. Roughan, C. Lund, and D.L. Donoho, “Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach,” IEEE/ACM Trans. Netw., vol.13, no.5, pp.947–960, 2005.
- [7] D. Jiang, X. Wang, L. Guo, H. Ni, and Z. Chen, “Accurate estimation of large-scale IP traffic matrix,” Int. J. Electron. Commun., vol.65, no.1, pp.75–86, 2011.
- [8] D. Jiang, Z. Xu, Z. Chen, Y. Han, and H. Xu, “Joint time-frequency sparse estimation of large-scale network traffic,” Comput. Netw., vol.55, no.15, pp.3533–3547, 2011.
- [9] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, “Spatio-temporal compressive sensing and internet traffic matrices (extended version),” IEEE/ACM Trans. Netw., vol.20, no.3, pp.662–676, 2012.
- [10] A. Gunnar, M. Johansson, and T. Telkamp, “Traffic matrix estimation on a large IP backbone: A comparison on real data,” ACM IMC, pp.149–160, 2004.
- [11] M.M. Rahman, S. Saha, U. Chengan, and A.S. Alfa, “IP traffic matrix estimation methods: Comparisons and improvements,” IEEE ICC, pp.90–96, 2006.
- [12] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z.W. Lin, and V. Kurien, “Pingmesh: A large-scale system for data center network latency measurement and analysis,” ACM SIGCOMM Comput. Commun. Rev., vol.45, no.4, pp.139–152, 2015.
- [13] H.X. Nguyen and P. Thiran, “Active measurement for multiple link failures diagnosis in IP networks,” 5th Passive and Active Measurement Conference (PAM 2004) Workshop, pp.185–194, 2004.
- [14] S. Agrawal, K.V.M. Naidu, and R. Rastogi, “Diagnosing link-level anomalies using passive probes,” IEEE INFOCOM, pp.1757–1765, 2007.
- [15] T. Inoue, T. Mano, K. Mizutani, S.i. Minato, and O. Akashi, “Re-thinking packet classification for global network view of software-defined networking,” IEEE ICNP, pp.296–307, 2014.
- [16] H. Wang, C. Qian, and S.S. Lam, “Practical network-wide packet behavior identification by AP classifier,” CoNEXT, 2015.
- [17] S.B. Moon, P. Skelly, and D. Towsley, “Estimation and removal of clock skew from network delay measurements,” IEEE INFOCOM, vol.1, pp.227–234, 1999.
- [18] Y. Gu, L. Breslau, N. Duffield, and S. Sen, “On passive one-way loss measurements using sampled flow statistics,” IEEE INFOCOM, pp.2946–2950, 2009.
- [19] N.G. Duffield and M. Grossglauser, “Trajectory sampling for direct traffic observation,” IEEE/ACM Trans. Netw., vol.9, no.3, pp.280–292, 2001.
- [20] S. Fujishige, Submodular Functions and Optimization, Elsevier, 2005.
- [21] L.A. Wolsey, “An analysis of the greedy algorithm for the submodular set covering problem,” Combinatorica, vol.2, no.4, pp.385–393, 1982.
- [22] R.K. Iyer and J.A. Bilmes, “Submodular optimization with submodular cover and submodular knapsack constraints,” Advances in Neural Information Processing Systems 26, pp.2436–2444, Curran Associates, 2013.
- [23] U. Feige, V.S. Mirrokni, and J. Vondrák, “Maximizing non-monotone submodular functions,” SIAM J. Comput., vol.40, no.4, pp.1133–1153, 2011.
- [24] G. Xie, J. Zhan, D. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford, “On static reachability analysis of IP networks,” IEEE INFOCOM, pp.2170–2183, 2005.
- [25] H. Yang and S.S. Lam, “Real-time verification of network properties using atomic predicates,” IEEE/ACM Trans. Netw., vol.24, no.2, pp.887–900, 2016.
- [26] P. Kazemian, G. Varghese, and N. McKeown, “Header space analysis: Static checking for networks,” 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2012), 2012.
- [27] Y.W.E. Sung, X. Sun, S.G. Rao, G.G. Xie, and D.A. Maltz, “Towards systematic design of enterprise networks,” IEEE/ACM Trans. Netw., vol.19, no.3, pp.695–708, 2011.
- [28] G. Liang, N. Taft, and B. Yu, “A fast lightweight approach to origin-destination IP traffic estimation using partial measurements,” IEEE/ACM Trans. Netw., vol.52, no.6, pp.2634–2648, 2006.
- [29] M. Coates and R. Nowak, “Network loss inference using unicast end-to-end measurement,” 13th ITC Specialist Seminar on IP Traffic, Measurement, and Modeling, 2000.
- [30] N. Duffield, “Network tomography of binary network performance characteristics,” IEEE Trans. Inf. Theory, vol.52, no.12, pp.5373–5388, 2006.
- [31] H.X. Nguyen and P. Thiran, “Network loss inference with second order statistics of end-to-end flows,” ACM IMC, p.227, 2007.
- [32] M.H. Firooz and S. Roy, “Network tomography via compressed sensing,” IEEE GLOBECOM, 2010.
- [33] T. Matsuda, M. Nagahara, and K. Hayashi, “Link quality classifier with compressed sensing based on ℓ_1 - ℓ_2 optimization,” IEEE Commun. Lett., vol.15, no.10, pp.1117–1119, 2011.
- [34] V. Padmanabhan, L. Qiu, and H. Wang, “Server-based inference of Internet link lossiness,” IEEE INFOCOM, pp.145–155, 2003.
- [35] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, “Network radar: Tomography from round trip time measurements,” ACM IMC, pp.175–180, 2004.
- [36] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, “On the performance of round trip time network tomography,” IEEE ICC, pp.483–488, 2006.
- [37] M. Chen, A.X. Zheng, J. Lloyd, M.I. Jordan, and E. Brewer, “Failure diagnosis using decision trees,” 1st International Conference on Autonomic Computing (ICAC 2004), pp.36–43, 2004.

Appendix: Proofs

A.1 Submodularity of Coverage Functions

This appendix proves the following Theorem 5.

Theorem 5. *Coverage function g_0 for $\epsilon = 0$ is submodular.*

We only show the proof for the path measurability, but the proof for the pathset is almost same.

The function g_0 for $\epsilon = 0$ is submodular if and only if $g_0(x|X) \leq g_0(x|X')$ for all $X \supseteq X'$ and $x \notin X$. If $\mathbf{a}_i = \mathbf{a}_j$ in X but $\mathbf{a}'_i \neq \mathbf{a}'_j$ in $X \cup \{x\}$, this is mentioned as *the arc pair, e_j, e_k , is distinguished by adding the new counter x to X where (a_{ij}) and (a'_{ij}) are the measurement matrices of X and X' , respectively. A part of feasible path P , which starts from e_i to e_j , is denoted by $P_{i \rightarrow j}$ ($e_i \in P_{i \rightarrow j}$ but $e_j \notin P_{i \rightarrow j}$).*

Lemma 9. *Let a new counter $x = (e_i, P)$ be added to counter set X and be placed after an arc e_k , where e_k was the last counter arc on feasible path P (Fig. A·1 (Top)). If an arc pair, $e_i, e_j \in E$, is distinguished by the new counter x , then they satisfy $e_i \in P_{k \rightarrow i} \wedge e_j \notin P_{1 \rightarrow i}$ or vice versa.*

Proof. Proof by contradiction. Suppose the other the other arc e_j is in $P_{1 \rightarrow i}$. If e_j is in the same part $P_{k \rightarrow i}$, then the pair e_i, e_j is clearly undistinguished, which is a contradiction. If e_j is in $P_{1 \rightarrow k}$, then the pair e_i, e_j is already distinguished in

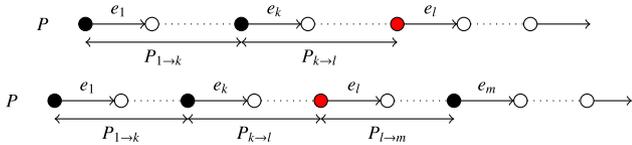


Fig. A-1 (Top) Counter (e_l, P) is put at the red vertex, which follows the preceding counters. (Bottom) Counter (e_l, P) is put at the red vertex, which is between other counters.

X . This, however, contradicts to the assumption that adding x distinguishes the pair e_i, e_j . \square

Lemma 10. *Let a new counter $x = (e_l, P)$ be added to counter set X and be placed between two counter arcs e_k and e_m on feasible path P (Fig. A-1 (Bottom)). If an arc pair, $e_i, e_j \in E$, is distinguished by x , then they satisfy $e_i \in P_{k \rightarrow l} \wedge e_j \in P_{l \rightarrow m}$ or vice versa.*

Proof. It can be proven by contradiction like Lemma 9. \square

The submodularity of g is finally shown in the following lemma.

Lemma 11. *The submodularity condition, $g_0(x|X) \leq g_0(x|X')$, holds for $X \supseteq X'$ and $x \notin X$.*

Proof. It is sufficient to show that any arc pair distinguished by adding x to X is also distinguished by adding x to X' .

- Consider the case that the counter $x = (e_l, P)$ is placed after the last counter arc on P (Fig. A-1 (Top)). If an arc pair e_i, e_j is distinguished by adding the new counter x to X , then $e_i \in P_{k \rightarrow l}$ and $e_j \notin P_{1 \rightarrow l}$ from Lemma 9. This arc pair is also distinguished by adding x to X' , because every feasible path has the ingress counter (e_1, P) as C_0 and e_i is in the subpath of $P_{1 \rightarrow l}$, which is created by x .
- Consider the other case, the counter $x = (e_l, P)$ is placed before the last counter arc on P (Fig. A-1 (Bottom)). In a similar way with the previous case, we can show that the arc pair is also distinguished by adding x to X' , using Lemma 10.

\square



Kohei Watabe received his B.E. and M.E. degrees in Engineering from Tokyo Metropolitan University, Tokyo, Japan, in 2009 and 2011, respectively. He also received the Ph.D. degree from Osaka University, Japan, in 2014. He was a JSPS research fellow (DC2) from April 2012 to March 2014. He has been an Assistant Professor of Graduate School of Engineering, Nagaoka University of Technology since April 2014. He is a member of the IEEE and the IEICE.



Toru Mano is a researcher in NTT Network Innovation Labs. He received the B.E. and M.E. degrees in Information Science and Technology from the University of Tokyo in 2009 and 2011, respectively. His research interests are network architectures and network optimization.



Takeru Inoue is a senior researcher in NTT Laboratories. He was an ERATO researcher at the Japan science and technology agency from 2011 through 2013. His research interests widely cover the design and control of network systems. He received the best paper award from the Asia-Pacific conference on communications in 2005, and also the research awards of the IEICE Information Network Group in 2002, 2005, and 2012. He received the B.E., M.E., and Ph.D. degrees from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2006, respectively. He is a member of IEEE.



Society.

Kimihiro Mizutani is a researcher in NTT Network Innovation Labs. He received the M.S. degree in information system from Nara Institute Science and Technology in 2010. His research interests include future internet architectures. He received the best student paper from International Conference on Communication Systems and Application (ICCSA) in 2010. He also received research awards from IPSJ and IEICE in 2010 and 2013, respectively. He is a member of IEICE and IEEE Communication



Osamu Akashi is a senior research scientist in NTT Network Innovation Laboratories. He received the M.S. degree in information science and the Ph.D. degree in mathematical and computing sciences from Tokyo Institute of Technology, in 1989 and 2001, respectively. His research interests include distributed systems, multi-agent systems, and network architectures.



Kenji Nakagawa received the B.S., M.S. and D.S. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 1980, 1982 and 1986, respectively. In 1985, he joined NTT (Nippon Telegraph and Telephone Corp.). Since 1992, he has been an associate professor of Graduate School of Engineering, Nagaoka University of Technology. He is an associate editor of the IEICE Transactions on Communications. His research interests include queueing theory, performance evaluation of networks, and geometric

theory of statics. Dr. Nakagawa is a member of the IEEE, SITA, and Mathematical Society of Japan.