Measuring Lost Packets with Minimum Counters in Traffic Matrix Estimation*

Kohei WATABE††, 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 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 \( t_{ij} \) is a traffic volume of \( i \)-th path (\( i = 1, 2, \ldots, m \)) 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 (\( j = 1, 2, \ldots, n \)), let \( x \) be an \( n \)-dimensional column vector whose \( j \)-th element is the packet loss ratio on \( j \)-th link, and let \( y \) 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 \( y \) are measured using counters placed at ingress and egress switches, while \( A \) is determined from forwarding configurations. Given \( T, A, \) and \( y \), equation \( TAx = y \) can be uniquely solved for \( x \) 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

---

*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
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 counters, but it only locates failed links without volume change. Instead of counters, we utilize the Boolean measurement matrix \( A \) with another Boolean vectors \( x \) and \( y \), where \( j \)-th element \( x_j \) of \( x \) indicates whether \( j \)-th link is failed while \( i \)-th element \( y_i \) of \( y \) indicates whether \( i \)-th path is failed. Note that the norm of \( x \), \( ||x|| \), 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, \( x \mapsto Ax \), if \( ||x|| \leq K \) (i.e., for sparse \( x \)) \[13\], \[14\]. This property allows us to uniquely determine \( x \) from \( y \) subject to \( Ax = y \) 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. As a result, while the error bounds can be controlled by threshold parameters, 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 10x more traffic than \( P_2 \). Packets are counted at nodes \( d_1 \), \( d_2 \), and \( d_3 \), 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 \( e_2 \). 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 unaccept-able 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\(^1\). 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 \( e_j \in E \) (\( j = 1, 2, \ldots, n \)) 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 \( \kappa_j \). 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

---

\(^1\)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.
The packet loss ratio is defined by the 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 $e_1$ was 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 vertices; 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 $\ell_j$ is expressed by $1 - \tau_j / \ell_j$ for path $P$ shown in Fig. 2. Let $\tilde{\tau}_j$ be an estimated value of $\tau_j$ (Note that counters are not necessarily placed at all arcs, and $\tau_j$ cannot be obtained directly from counter values if a counter is not placed at $(e_j, P)$). The counter set $C$ is $\alpha$-measurable if,

$$\forall P \in \mathcal{P}, \forall e_j \in P, \forall \tau_j : |\alpha \cdot \tau_j - \tilde{\tau}_j| \leq \frac{1}{\alpha},$$

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

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

where $|C|$ is the number of counters and $\alpha$ 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 $\alpha$-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 $\alpha$-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 $C$ $\alpha$-measurable. Hence, we reformulate our problem as follows,

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

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 $Q = \{e_j, e_{j+1}, \ldots, e_{k-1}\}$ is a part of a

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>A directed graph $(V, E)$ that represent a network. $V$ and $E$ represent a set of vertices and arcs.</td>
</tr>
<tr>
<td>$P$</td>
<td>A feasible path: a path along which there exist some packets forwarded.</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>The set of feasible paths in the network.</td>
</tr>
<tr>
<td>$e_j$</td>
<td>An arc.</td>
</tr>
<tr>
<td>$e$</td>
<td>An abnormal arc.</td>
</tr>
<tr>
<td>$(e_j, P)$</td>
<td>A counter placed at arc $e_j$ on path $P$.</td>
</tr>
<tr>
<td>$C$</td>
<td>A set of counters placed in the network.</td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>The set of all potential counters $E \times \mathcal{P}$.</td>
</tr>
<tr>
<td>$G_0$</td>
<td>The set of ingress counters in the network.</td>
</tr>
<tr>
<td>$X$</td>
<td>A set of additional counters $C \setminus G_0$.</td>
</tr>
<tr>
<td>$X_0$</td>
<td>An intermediate counter set assuming $\epsilon = 0$.</td>
</tr>
<tr>
<td>$X_{\alpha, \delta}$</td>
<td>A counter set that outputted by Algorithm 3.</td>
</tr>
<tr>
<td>$\tau_j$</td>
<td>An estimated value of $\tau_j$.</td>
</tr>
<tr>
<td>$\ell_j$</td>
<td>The packet loss ratio through arc $e_j$.</td>
</tr>
<tr>
<td>$\delta$</td>
<td>An estimator of the pass ratio on the abnormal arc.</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>The threshold of loss ratio for an abnormal state. Loss ratio of a normal state arc is smaller than $\epsilon$.</td>
</tr>
<tr>
<td>$\delta$</td>
<td>The threshold of loss ratio for an abnormal state. Loss ratio of a abnormal state arc is larger than $\delta$. It also used as a loss threshold for a separable path.</td>
</tr>
<tr>
<td>$d$</td>
<td>The maximum path length $\max</td>
</tr>
<tr>
<td>$Q$</td>
<td>A separable subpath: a part of a feasible path between a pair of counters that satisfies $(1 - \delta) &lt; (1 - \epsilon)^{d-2}$ (see Fig. 2).</td>
</tr>
<tr>
<td>$A(C)$</td>
<td>A measurement matrix whose $(i, j)$ element $a_{ij}$ indicates that a separable path $Q_i$ includes an arc $e_j$ (see (4)).</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>The $j$-th column vector of a measurement matrix $A(C)$.</td>
</tr>
<tr>
<td>$x$</td>
<td>A $n$-dimensional column vector whose $j$-th element indicates that $j$-th arc is an abnormal state.</td>
</tr>
<tr>
<td>$y$</td>
<td>A $m$-dimensional column vector whose $i$-th element indicates that subpath $Q_i$ includes an abnormal arc.</td>
</tr>
<tr>
<td>$x^*$</td>
<td>The unique solution of $Ax = y$.</td>
</tr>
<tr>
<td>$g$</td>
<td>A coverage function that represents how close the measurement matrix is to 1-independent.</td>
</tr>
<tr>
<td>$g_0$</td>
<td>A coverage function for $\epsilon = 0$.</td>
</tr>
<tr>
<td>$H(x)$</td>
<td>$x$-th harmonic number $\sum_{i=1}^1 1/i$.</td>
</tr>
</tbody>
</table>
feasible path \( P = (e_1, e_2, \ldots, e_l) \) between a pair of counters \( \{(e_j, P), (e_k, P)\} \) (Fig. 2) and satisfies \( (1 - \delta) < (1 - e)^{k-j} \). Inspecting the counter values \( \{k_j, \kappa_k\} \), we can tell the subpath contains an abnormal arc or not; The subpath contains an abnormal arc if and only if \( k_k/k_j \leq 1 - \delta \). The condition \( (1 - \delta) < (1 - e)^{k-j} \) is needed in order to make subpaths containing an abnormal arc separable. The minimum loss ratio of a subpath containing an abnormal arc is \( \delta \). Conversely, the maximum loss ratio of a subpath in which all arc is normal state is \( 1 - (1 - e)^{k-j} \). If \( (1 - e)^{k-j} \leq (1 - \delta) \), 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 \( Q = \{Q_1, Q_2, \ldots, Q_m\} \) 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

\[
a_{ij} = \begin{cases} 
1 & e_j \in Q_i \\
0 & \text{otherwise} 
\end{cases}
\]

\[
(4)
\]

The \( j \)-th column vector is denoted by \( a_j \) \((j = 1, 2, \ldots, n)\). 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., \( \forall j \neq j' : a_j \neq a_{j'} \) and \( \forall j : a_j \neq 0 \).

Vectors \( x \) and \( y \) in the introduction are rigorously re-defined as follows. \( x \) is an \( n \)-dimensional column vector whose \( j \)-th element indicates that \( j \)-th arc is an abnormal state. \( y \) 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 \( \alpha = (1 - e)^{d-2} \), we can relate the sufficient conditions of \( \alpha \)-measurable to the measurement matrix. Given the measurement matrix \( A(C) \), sufficient conditions of measurability are described by the following theorem.

**Theorem 1.** When \( \tau_j \) is estimated by Algorithm 1, a counter set \( C \) is \( (1 - e)^{d-2} \)-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 counters as \( X = C \setminus C_0 \).

### Algorithm 1: Transmitted Packets Estimation (Per-Path)

**Input:** Paths \( \mathcal{P} \), Separable subpaths \( \{Q_1, Q_2, \ldots, Q_m\} \), and counter values \( \kappa \).

**Output:** Estimated number of packets \( \hat{\tau} \) at every arc for each feasible path.

1. for all \( i \in \{1, \ldots, m\} \) do
2. \( \text{if } Q_i \text{ contains an abnormal arc then } y_i \leftarrow 1 \text{ else } y_i \leftarrow 0; \)
3. \( e_{\delta} \leftarrow \text{NULL}; \)
4. \( \phi \leftarrow 0; \)
5. if \( y \neq 0 \text{ then } / / \text{ Find an abnormal arc} \)
6. \( e_{\delta} \leftarrow \text{Find an arc } e \text{ such that } y = a_e; \)
7. Find a separable subpath \( Q = \{e_l, \ldots, e_{l-1}\} \) that contains \( e_{\delta}; \)
8. \( \phi \leftarrow k_l/(\beta_{-\delta-1}); \)
9. return [Estimate \( \hat{\tau} (P, e, \kappa) : P \in \mathcal{P} \) ];

### 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 \( \kappa \), where \( \beta_j \) equals to \( \sqrt{(1 - e)^j} \) 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 \( P = [e_1, \ldots, e_l] \), the number of transmitted packets \( \tau_j \) falls between \( (1 - e)^{j-1} \tau_1 \) and \( (1 - e)^{j-1} \tau_1 \leq \tau_j \leq (1 - \epsilon) \tau_1 \), since a loss ratio of normal arc is less than or equal to \( \epsilon \) and there are \( j - 1 \) arcs between \( e_1 \) and \( e_{j-1} \). Therefore, we define the estimator \( \hat{\tau}_j = \beta_{j-1}\kappa_l \) as the midpoint between \( (1 - e)^{j-1} \kappa_l \) and \( \kappa_l \) in a logarithmic scale since \( \kappa_l = \tau_1 \) from the definition. Consequently, the estimated value \( \hat{\tau}_j \) satisfies the condition \( \beta_{j-1}\tau_l \leq \hat{\tau}_j \leq \tau_j/\beta_{j-1} \) for measurability. If there is an abnormal arc between \( e_1 \) and \( e_{j-1} \), \( j \)-th satisfies \( (1 - \epsilon_3)(1 - e)^{j-1} \tau_1 \leq \tau_j \leq (1 - \epsilon_4) \tau_1 \), since a loss ratio of normal arc is less than or equal to \( \epsilon \) and there are \( j - 2 \) normal arcs between \( e_1 \) and \( e_{j-1} \). By replacing \( (1 - \epsilon_4) \) with the estimator \( \phi \) of the pass ratio \( (1 - \epsilon_4) \), we define the estimator \( \hat{\tau}_j = \phi \beta_{j-2}\kappa_l \). Rigorous arguments, including the definition of \( \phi \) 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 \( Ax = y \) from measurement results \( \kappa \) (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 \( e_j = \beta_{j-1}\kappa_l \) if \( j \leq a \), and \( \phi \beta_{j-2}\kappa_l \) 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 $P = \{e_1, \ldots, e_l\}$, 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: $T \leftarrow 0$
2: for all $j \in \{1, \ldots, l\}$ do
3: if $e_a \notin P \cup j \leq a$ then
4: $\hat{\tau}_j \leftarrow \beta_j \cdot \tau_j$
5: else
6: $\hat{\tau}_j \leftarrow \phi \beta_j \cdot \tau_j$
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 $\lor$ 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 $(j = 1, \ldots, n)$. Then the system of Boolean equations $Ax = y$,

$$a_{k_1}x_1 \lor a_{k_2}x_2 \lor \cdots \lor a_{k_n}x_n = y_i \quad (i = 1, 2, \ldots, m),$$

have unique solution $x^* \in [0,1]^n$ such that $\|x^*\| \leq 1$ and $x_i^* = 1$ if and only if the arc $e_j$ is abnormal.

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

Note that if the $j$-th arc is abnormal, $x_j^* = 1$, then $Ax^* = a_j$. 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 $k_i/k_j \leq 1 - \delta$. 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 $P = \{e_1, e_2, \ldots, e_l\}$ be a feasible path that does not include abnormal arcs. Then

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

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 $j = 1, 2, \ldots, l$.

Lemma 3. Let $\ell_a$ and $Q = \{e_f, \ldots, e_{k-1}\}$ be the loss ratio of the abnormal arc $e_a$ and a separable subpath that contains $e_a$, respectively. Then

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

where $\phi = k_e / (\beta_{k-1-1} \kappa_1)$ is an estimator of the pass ratio $1 - \ell_a$ on the abnormal arc.

Proof. This is because the separable subpath $Q = \{e_f, \ldots, e_{k-1}\}$ contains $e_a$ and $(1 - \epsilon)^{k-1} \cdot (1 - \ell_a) \cdot \kappa_1 \leq \kappa_e \leq (1 - \ell_a) \cdot \kappa_1$ since $\kappa_e = \kappa_e$.

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

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

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

4.3 Counter Set Minimization Algorithm

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

$$\min_{X \subseteq \mathcal{C}} \left[ |X : g(X) = g(\hat{\mathcal{C}})| \right],$$

where $\hat{\mathcal{C}}$ is the set of per-path counters, $E \times P$, 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

$$g(X) = \left[ |(j,k) : 0 \leq j < k \leq n, a_j \neq a_k| \right],$$

where $a_j$ is the $j$-th column vector of the measurement matrix $A(C_0 \cup X)$ and $a_0$ is the zero vector 0. We call this function $g$ coverage function. Note that the coverage function $g(X)$ 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

$$\left(1 + \frac{d-1}{\log_2(1-\delta)-1}\right) H\left(\frac{n(n+1)}{2}\right),$$

where $\delta$ is the degree of the graph $G$ and $H(x)$ is the entropic function.
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 $x \in \hat{C}$ 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 $Q = \{e_j, \ldots, e_{k-1}\} \subseteq P$ 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: $\forall e_{j'} \in Q \setminus \{e_j, e_{k-1}\} = \{e_{j+1}, \ldots, e_{k-2}\} : (e_{j'}, P) \notin X$. 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 $a_{i_f,j} = a_{i_f,j'} = a_{i_f,j} \vee a_{i_f,j'}$ for all $j$ and $j'$, where $Q_{i_f}$ is a separable but not irreducible subpath that includes two irreducible subpaths $Q_1$ and $Q_2$.

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. $X_{\text{ALG}}$ be the output of Algorithm 3, $X^*$ be the optimal counter set of (10) for given thresholds $\epsilon$ and $\delta$. We define a feasible counter set as a counter set that makes a measurement matrix 1-independent. Both $X_0$ and $X_{\text{ALG}}$ 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 $X_{\text{ALG}}$ for given $\epsilon$ and $\delta$. This is because, when the lower threshold $\epsilon$ 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 $X_{\text{ALG}}$ for given $\epsilon$ and $\delta$ can differ from the solution $X_0$ for $\epsilon = 0$ since the measurement matrix $A(X)$ can be changed depending on $\epsilon$ and $\delta$. The measurement matrix is defined by separable subpaths under the given counters, and whether or not a subpath is separable depends on $\epsilon$ and $\delta$. 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 $x = (e, P) \in E \times P \setminus X$. After the while loop, the counter set $X_0$ is a feasible counter set for $\epsilon = 0$: $g_0(X_0) = g_0(\hat{C})$ (Lemma 6).

Then, Algorithm 3 converts the feasible counter set $X_0$ into a feasible one for given thresholds $\epsilon$ and $\delta$ 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 $Q = \{e_j, \ldots, e_{k-1}\}$ is divided into two subpaths $Q_1 = \{e_j, \ldots, e_{k'-1}\}$ and $Q_2 = \{e_{k'}, \ldots, e_{k-1}\}$. Line 1 identifies the subscript $k'$ of the arc at which a counter should be added. $-1 + \lfloor \log_{1-\delta} (1-\epsilon) \rfloor$ 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 $\epsilon$ and $\delta$. As a result, every irreducible subpath $Q^*$ for $\epsilon = 0$ is divided into irreducible subpaths $Q_1, Q_2, \ldots, Q_l$ for given thresholds $\epsilon$ and $\delta$. After Line 6, the counter set $X_{\text{ALG}}$ is a feasible counter set: $g(X_{\text{ALG}}) = g(\hat{C})$ (Lemma 5).

4.3.3 Proof

First, we prove the correctness of the Algorithm 3, i.e., the output counter set $X_{\text{ALG}}$ satisfies $g(X_{\text{ALG}}) = g(\hat{C})$. 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 $|P|$.

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

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

**Lemma 6.** The intermediate counter set $X_0$ satisfies...
Proof of Lemma 5. We will prove by contradiction. Let $A = (a_{ij})$ be the measurement matrix of $X_{\text{ALG}}$ for given $\epsilon$ and $\delta$. $A' = (a'_{ij})$ be the irreducible measurement matrix of $X_0$ for $\epsilon = 0$. As shown in Algorithm 3, the subroutine Separ1e shown in Algorithm 4 computes the difference between $X_{\text{ALG}}$ and $X_0$. The irreducible separable subpath of $X_0$ for $\epsilon = 0$ is divided into separable subpaths for given $\epsilon$ and $\delta$ by $X_{\text{ALG}}$. Suppose that $g(X_{\text{ALG}}) < g(X_0)$. Then, there exist two arc $e_j$ and $e_k$ such that $a_j = a_k$ and $a'_j \neq a'_k$. Hence, there exists an irreducible separable subpath $Q'$ for $\epsilon = 0$ at $X_0$ such that $a'_{ij} \neq a'_{ik}$. This subpath $Q'$ must be divided into some subpaths $Q_1, \ldots, Q_l$ through the subroutine Separ1e. Thus, we have

$$a'_{ij} = a_{ij} \lor a'_{ij} \lor \cdots \lor a_{ij},$$

$$a'_{ik} = a_{ik} \lor a'_{ik} \lor \cdots \lor a_{ik},$$

(13) (14)

since all arcs in subpath $Q_l$ are included in either of $Q_1, \ldots, Q_l$. The right-hand sides of (13) and (14) must not be equal since $a'_j \neq a'_k$. However, this contradicts $a_j = a_k$ because the right-hand sides of (13) and (14) are equal if $a_j = a_k$.

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 $H(\max_{x \in C} g_0((x)))$.

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

Lemma 7. For given $\epsilon$, $\delta$, and $d$, the following inequality holds.

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

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

$$|X_0| \leq H\left(\frac{n(n + 1)}{2}\right) |X'|.$$  

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 $Q \in Q'$, Separ1e adds at most $\frac{d - 1}{\log_{1-\epsilon}(1 - \delta) - 1}$ counters to $X_0$. Hence, we have

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

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

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

$$|X_0| \leq H(\max_{x \in C} g_0((x))) |X_0'| \leq H\left(\frac{n(n + 1)}{2}\right) |X_0'|$$  

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 $\epsilon$ and $\delta$ is also separable for $\epsilon = 0$.

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, $e_1, e_2, \ldots, e_l$, 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., $C \subseteq E \times 2^P$.

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 $\tau_j$ is estimated by a similar algorithm with Algorithm 1, a counter set $C$ is weakly $\alpha$-measurable, if the measurement matrix $A(C)$ is $l$-independent, where the constant $\alpha$ is defined as\(^\dagger\)

$$\alpha = \sqrt{\frac{\zeta (1 + \frac{1}{\alpha})}{\frac{1}{\alpha} + \zeta}}.$$  

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

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

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

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

where $\lambda_j$ is the number of packets dropped between $e_1$ and $e_j$, i.e., $\lambda_j = \tau_1 - \tau_j$. Again, we reformulate our problem

\(^\dagger\)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.

\begin{figure}[h]
\centering
\includegraphics[width=5cm]{fig3.png}
\caption{A subtree $Q$. The root arc is $e_0$ while the leaf arcs are $e_j$ ($j = 1, 2, \ldots, l$); the root is not included in $Q$. Counters, ($e_j, P$)'s, are placed at the black vertices.}
\end{figure}


Algorithm 5: Counter Placement (Path-Set)

Input: A set of feasible paths $\mathcal{P}$ and loss thresholds $\epsilon, \delta$, counter values $k$, and a real number $\rho$.

Output: A counter set $\mathcal{X}_{\text{ALG}}$ that satisfies $g(\mathcal{X}_{\text{ALG}}) = g(\hat{C})$.

1. $X_0 \leftarrow \emptyset$;
2. while $\exists x \in \hat{C} : g_0(x; X_0) > 0$ do
3. \[ \bar{x} \leftarrow \text{Find } x \text{ such that } \left( ( (1/3) - \rho \right) \max_{x \in \mathcal{C}} g_0(x; X_0) \leq g_0(x; X_0); \]
4. $X_0 \leftarrow X_0 \cup \{x\};$
5. $Q' \leftarrow \text{Irreducible separable subtrees of } X_0 \text{ at } \epsilon = 0;\]
6. for all $Q \in Q'$ do $X_{\text{ALG}} \leftarrow \text{Separate}(Q, X_0, \epsilon, \delta, k);$
7. return $\mathcal{X}_{\text{ALG}}$

Algorithm 6: Separate (Path-Set)

Input: A subtree $Q = \{e_1, \ldots, e_k\} \subseteq \mathcal{P}_0$, a counter set $X_0$, loss thresholds $\epsilon, \delta$, and counter values $k$.

Output: A counter set $\mathcal{X}_{\text{ALG}}$ that divides the subtree $Q$ into separable subtrees.

1. $\mathcal{X}_{\text{ALG}} \leftarrow X_0$;
2. if $Q$ is separable then return $\mathcal{X}_{\text{ALG}}$; // Do nothing
3. $e_\epsilon \leftarrow \text{arg min}_{e \in Q} h(e);$
4. $\mathcal{P}_e \leftarrow \text{arg min}_{e \in e, \cup e, \setminus e, \mathcal{P}} h(P);$
5. if $h(e_\epsilon) \leq h(P_e)$ then // Divide at $e_\epsilon$
6. $\mathcal{X}_{\text{ALG}} \leftarrow \mathcal{X}_{\text{ALG}} \cup \{e_\epsilon, \cup e, \setminus e, \mathcal{P}_e\};$
7. $(Q_1, Q_2) \leftarrow (U_{e_\epsilon}, Q, \cup e, \setminus e, \mathcal{P}_e);$
8. else // Peel off $P_e$
9. $\mathcal{X}_{\text{ALG}} \leftarrow \mathcal{X}_{\text{ALG}} \setminus \{e_\epsilon, \mathcal{P}_e\} \cup \{(e_\epsilon, \mathcal{P}_e), (e_\epsilon, \mathcal{P}_e^C)\};$
10. $(Q_1, Q_2) \leftarrow (Q, \cup P_e, \setminus P_e^C);$
11. $\mathcal{X}_{\text{ALG}} \leftarrow \text{Separate}(Q_1, \mathcal{X}_{\text{ALG}}, \epsilon, \delta, k);$
12. $\mathcal{X}_{\text{ALG}} \leftarrow \text{Separate}(Q_2, \mathcal{X}_{\text{ALG}}, \epsilon, \delta, k);$
13. return $\mathcal{X}_{\text{ALG}}$

as follows,

\[ \min_{C \subseteq E \times \mathbb{R}} |A(C)| : A(C) \text{ is 1-independent}. \] (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 $\max \{g_0(x; X_0) : x \in \hat{C} \setminus X_0\}$ by calculating all the gain $g_0(x; X_0)$ for $x \in \hat{C} \setminus X_0$ which takes exponential time (Compare Line 3 in Algorithms 3 and 5). This is because the number of candidate counters roughly equals to $|E|^2|\mathbb{R}|$. 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 $\max_{x \in \mathcal{C}} g_0(x; X_0)$.

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 $\epsilon$ and $\delta$. Algorithm 6 adds a counter $(e_k, \cup e_k, \setminus e_k, \mathcal{P}_j)$ inside the subtree and the subtree is divided up 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 $\mathcal{P}_j$: $\mathcal{P}_j^C = \bigcup_{P \in \mathcal{P}_j^C} P_j$ (Line 9). This peels the subpath $Q \cap P_j$ from the subtree $Q$ and we have new subpath $Q \cap P_j$ 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 $h$: 

\[ h(Q) = \max_0 \left( \sum_{j=1}^{l} (1 - (1 - \epsilon)^{\delta j} - \delta \min k_j) \right). \] (22)

Note that $h(Q) = 0$ if and only if $Q$ is separable. Hence, we define the deviation of dividing at $e_k$ as $h(e_k) = \max_0 (h(U_{e_k}), h(Q \setminus U_{e_k}))$ and that of peeling off $P_j$ as $h(P_j) = \max_0 (h(Q \cap P_j), h(Q \cap \mathcal{P}_j^C))$.

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 tradition-
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 $9.47 \times 10^{8}$, $2.32 \times 10^{18}$, and $1.89 \times 10^{5}$, 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 $\kappa$, 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

---

**Table 2** Number of elements in the configuration datasets.

<table>
<thead>
<tr>
<th></th>
<th>Internet2</th>
<th>Stanford</th>
<th>Purdue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switches</td>
<td>9</td>
<td>16</td>
<td>1,503</td>
</tr>
<tr>
<td>Ports used</td>
<td>140</td>
<td>58</td>
<td>526</td>
</tr>
<tr>
<td>Feasible paths $</td>
<td>P</td>
<td>$</td>
<td>11,159</td>
</tr>
<tr>
<td>Maximum path length $d$</td>
<td>5</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Arcs with FIB/ACL $</td>
<td>E</td>
<td>$</td>
<td>140</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th></th>
<th>Internet2</th>
<th>Stanford</th>
<th>Purdue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial counters $</td>
<td>C_0</td>
<td>$</td>
<td>11,159</td>
</tr>
<tr>
<td>Additional counters $</td>
<td>X</td>
<td>$</td>
<td>116</td>
</tr>
<tr>
<td>Indistinguishable arc-pairs / all arc-pairs</td>
<td>0 / 9,730</td>
<td>49 / 8,646</td>
<td>0 / 138,075</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th></th>
<th>Internet2</th>
<th>Stanford</th>
<th>Purdue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of matrix rows</td>
<td>115</td>
<td>66</td>
<td>380</td>
</tr>
</tbody>
</table>

---

for $\epsilon = 0$; an arc pair $(e_i, e_j) \in E^2$ is indistinguishable if the corresponding column vectors $a_i, a_j$ 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 $\delta$; Fig. 5 shows the number of matrix rows for $\epsilon = 10^{-5}$ and $10^{-2} \leq \delta \leq 10^{-1}$. 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 $\delta > 5.0 \times 10^{-3}$, 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.

<table>
<thead>
<tr>
<th></th>
<th>Internet2</th>
<th>Stanford</th>
<th>Purdue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional counters</td>
<td>116</td>
<td>65</td>
<td>380</td>
</tr>
<tr>
<td>Additional counters</td>
<td>116</td>
<td>67</td>
<td>359</td>
</tr>
</tbody>
</table>

rrately 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 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

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 \( g_0(\{x\}) \leq g_0(\{x'\}) \) for all \( x \supseteq x' \) and \( x \notin X \). If \( a_i = a_j \) in \( X \) but \( a'_i \neq a'_j \) in \( X \cup \{x\} \), this is mentioned as the arc pair, \( e_i, e_j \), is distinguished by adding the new counter \( x \) to \( X \) where \( (a_i) \) and \( (a'_j) \) 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 \( x = (e_j, P) \) be added to set counter \( X \) 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 \( e_i \in P_{k\rightarrow i} \land e_j \notin P_{i\rightarrow k} \) or vice versa.

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...
Lemma 10. Let a new counter \( x = (e_i, P) \) 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 \( e_i \in P_{k \rightarrow l} \cap e_j \in P_{l \rightarrow m} \), or vice versa.

Proof. It can be proven by contradiction like Lemma 9. □

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 \( x = (e_i, P) \) 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 \( e_i \in P_{l \rightarrow j} \) and \( e_j \notin P_{l \rightarrow i} \) from Lemma 9. This arc pair is also distinguished by adding \( x \) to \( X' \), because every feasible path has the ingress counter \( (e_i, P) \) as \( C_0 \) and \( e_i \) is in the subpath of \( P_{l \rightarrow i} \), which is created by \( x \).

- Consider the other case, the counter \( x = (e_i, P) \) 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. □

\[ P \]

\[ P \]

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