PAPER *Special Section on Quality of Communication Services Improving Quality of Life*

Bitwise Operation-Based In-Network Processing for Loss Tomography

Takahiro MATSUDA[†a)] ***and*** **Tetsuya TAKINE**[†], ***Members***

SUMMARY Network tomography is an inference technique for internal network characteristics such as link loss rate and link delay from end-to-end measurements. In this paper, we consider network tomography for link loss rates, which is referred to as loss tomography. We propose a loss tomography scheme with bitwise operation-based in-network processing. Intermediate nodes generate coded packets by performing bitwise-operations on received packets so as to embed information about paths along which those packets have been transmitted. The coded packets are then forwarded to downstream nodes. In this way, receiver nodes obtain information about paths along which packets are transmitted successfully. Moreover, we show a recursion to compute the likelihood function of path loss rates, which can be utilized in estimating link loss rates from path loss information.

key words: *network tomography, link loss rates, in-network processing, likelihood function*

1. Introduction

Network monitoring is an important technique for network management in various network environments such as IP networks, optical networks, and wireless sensor networks [5], [6], [16], [17]. In this paper, we consider *network tomography*, which is an inference technique for internal network characteristics such as link loss rates and link delays through end-to-end measurements [6]. In particular, we focus on active *loss tomography*, i.e., network tomography for link loss rates with active measurements, where source nodes transmit probe packets in order to collect information about end-to-end characteristics of the network.

In general, the difficulty in network tomography lies in the fact that link states cannot be estimated uniquely from the result of end-to-end measurements. In fact, if we assume temporal and spatial independency of network characteristics, the relationship between end-to-end measurements and link states will be represented to be an ill-posed linear system [6]. In order to solve such an identifiability problem, several schemes have been proposed, taking account of correlation in measured network characteristics [2], [3], [20]. In [3], a single-source multicast transmission is applied to networks with tree topologies. As for networks with general topologies, multiple single-source multicast trees are applied in [2] and the performance correlation among unicast flows is used in [20].

Network coding [1] is a promising technique for network tomography in networks with general topologies [7]–[9], [11], [16]. Network coding-based loss tomography schemes correlate the performance of different paths by sharing probe packets among the paths, while the multicast-based schemes utilize a tree topology. In network coding, intermediate nodes are allowed to encode several received packets into coded packets before forwarding. Because coding information in received packets implicitly contain information about paths along which they are transmitted, path loss rates may be obtained from received packets, from which link loss rates will be estimated. Note also that network coding has an additional advantage of improving bandwidth usage because it can reduce the number of forwarded packets in the network.

The study of loss tomography with network coding was started in [11] and since then, several loss tomography schemes have been studied. In [8], the effect of the number of source/receiver nodes and their positions are discussed and several algorithms to estimate link loss rates are proposed. In these schemes, a factor graph [14] is formulated from a network topology and established paths, and link loss rates are estimated with inference algorithms such as Belief Propagation. Related work can be found in [9], [16].

In tree topologies, linear network coding defined on GF(2) (i.e., XOR operations) is enough to estimate link loss rates. If a transmitted packet is encoded more than two times on its paths, however, coding information of the packet is canceled out. Therefore, in general directed acyclic networks, we have to use linear network coding defined on GF(q) ($q > 2$) [7]. Note here that the increase of the field size q improves the capability to identify paths which packets passed through. Also, the number of paths should increase with the number of links in the network. Therefore, q should be increased with the size of the network. In [11], in order to determine an appropriate value of q, the *monitoring ambiguity* is evaluated for random linear network coding [12], where the monitoring ambiguity is said to exist if different path loss patterns yield the same coding information.

As discussed above, the problem of loss tomography can be divided into two subproblems. One is the method for collecting path loss information and the other is the inference technique for link loss rates from path loss information. This paper is mainly concerned with the former problem. More specifically, this paper proposes a method for collecting path loss information by means of a bitwise operation-based in-network processing. As we will see, our scheme does not suffer from the monitoring ambiguity in terms of

Manuscript received May 31, 2012.
Manuscript revised September 10, 2012.

[†]The authors are with the Graduate School of Engineering, Osaka University, Suita-shi, 565-0871 Japan.
a) E-mail: matsuda@comm.eng.osaka-u.ac.jp

DOI: 10.1587/transcom.E96.B.470

path loss patterns and it is applicable to any directed acyclic networks. The scheme is no longer network coding because coded packets received at receiver nodes are not decodable and this is the reason why we call it *in-network processing*, instead of network coding. Note here that in active network tomography, the undecodability does not cause any problems because probe packets do not need to be decoded unless path loss information can be extracted from them. Furthermore, we provide a recursion to compute the likelihood function, which can be utilized in conducting the statistical inference of link loss rates.

It is worth mentioning that in a situation that intermediate nodes are allowed to implement additional functionalities, we can achieve link loss monitoring without loss tomography: intermediate nodes themselves monitor link loss rates by counting packet losses in their output buffers. In fact, in TCP/IP-based networks, network internal characteristics can be retrieved by end hosts via SNMP (Simple Network Management Protocol) [4]. Although this approach is a possible solution to monitor congestion related link loss rates, it cannot monitor non-congestion related packet loss such as packet erasure on wireless links. Further, in this approach, we cannot monitor path loss rates and cannot analyze the relationship between link loss rates and path loss rates. One of advantages of loss tomography is that we can monitor both pass loss rates and link loss rates due to congestion and non-congestion related packet loss. In addition to the advantage of loss tomography, the in-network processing approach such as network coding-based schemes and the proposed scheme can reduce the number of probe packets injected into the network.

The remainder of this paper is organized as follows. In Sect. 2, we explain the network model in this paper. In Sect. 3, we explain our proposed loss tomography scheme. In Sect. 4, we provide a recursion to compute the likelihood function of path loss rates. In Sect. 5, we demonstrate how to use the likelihood function in estimating link loss rates. Finally, we conclude the paper in Sect. 6.

2. Network Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a directed acyclic network, where $\mathcal{V}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denote the sets of nodes and links, respectively. We define $N = |\mathcal{V}|$ and $L = |\mathcal{E}|$ as the numbers of nodes and links, respectively, in the network. Let $\mathcal{V}_\mathrm{S}$ and $\mathcal{V}_\mathrm{R}$ denote the sets of source and receiver nodes, respectively, where source nodes generate probe packets and transmit them into the network, and receiver nodes serve as the destinations of those probe packets. We define $N_\mathrm{S} = |\mathcal{V}_\mathrm{S}|$ and $N_\mathrm{R} = |\mathcal{V}_\mathrm{R}|$ as the numbers of source and receiver nodes, respectively. We refer to other nodes in $\mathcal{V} \setminus \{\mathcal{V}_\mathrm{S} \cup \mathcal{V}_\mathrm{R}\}$ as intermediate nodes, which may encode received packets and forward coded packets to their output links. Let $\mathcal{P}$ denote the set of all paths from nodes in $\mathcal{V}_\mathrm{S}$ to nodes in $\mathcal{V}_\mathrm{R}$. We define indegree(v) and outdegree(v) as the numbers of input and output links, respectively, of node $v \in \mathcal{V}$. Note that indegree(s) $= 0$ for $s \in \mathcal{V}_\mathrm{S}$ and outdegree(r) $= 0$ for $r \in \mathcal{V}_\mathrm{R}$.

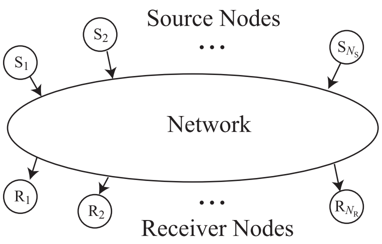
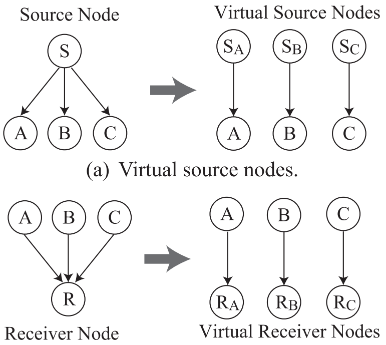


Fig. 1 Network structure for loss tomography.



(a) Virtual source nodes.

(b) Virtual receiver nodes.

Fig. 2 Virtual source and receiver nodes.

Figure 1 shows the network structure for the proposed loss tomography scheme, where there are N_S source nodes and N_R receiver nodes. Without loss of generality, we assume that each source node has only one output link and each receiver node has only one input link (i.e., outdegree(s) $= 1$ for $s \in \mathcal{V}_\mathrm{S}$ and indegree(r) $= 1$ for $r \in \mathcal{V}_\mathrm{R}$). If source (resp. receiver) nodes have multiple output (resp. input) links, they are separated equivalently into virtual source (resp. receiver) nodes with one output (resp. input) link, as shown in Fig. 2.

We assume that packets injected into the network are lost on each link independently and randomly. We define $\boldsymbol{\alpha}$ as a $1 \times L$ vector whose ith ($i = 1, 2 \ldots, L$) component α_i represents the probability that a packet transmitted on the ith link e_i is lost.

3. Simple Bitwise Operation for Loss Tomography

3.1 Monitoring Ambiguity [11]

Before the proposed scheme, we explain monitoring ambiguity with Fig. 3, where packets are transmitted with network coding. In the figure, nodes S_1 and S_2 transmit packets x_1 and x_2, respectively, and nodes R_1 and R_2 receive coded packets y_1 and y_2, respectively. In the network, packets are encoded at nodes C, D, and E. When using linear network coding defined on GF(q) ($q \geq 2$), if no packet loss occurs in the network, the relationship between $(y_1\ y_2)^\top$ and $(x_1\ x_2)^\top$ is given by

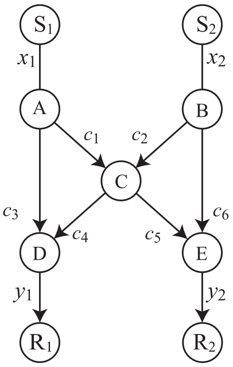


Fig. 3 An example of network coding.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} c_1c_4 + c_3 & c_2c_4 \\ c_1c_5 & c_2c_5 + c_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Interested readers may refer to the literature [13], [15], [18] for details of linear network coding.

In the equation, the matrix includes path information which packets passed through. For example, the $(1, 1)$ component $c_1c_4 + c_3$ of the matrix includes information of two paths $p_1 = \mathrm{S}_1 \to \mathrm{A} \to \mathrm{D} \to \mathrm{R}_1$ and $p_2 = \mathrm{S}_1 \to \mathrm{A} \to \mathrm{C} \to \mathrm{D} \to \mathrm{R}_1$. When a packet loss occurs on path p_1 and no packet loss occurs on path p_2, the $(1, 1)$ component is given by c_1c_4. When a packet loss occurs on path p_2 and no packet loss occurs on path p_1, the $(1, 1)$ component is given by c_3. When packet losses occur on both paths p_1 and p_2, the $(1, 1)$ component is given by 0. Therefore, in order to identify paths which packets passed through, four values $\{c_1c_4 + c_3, c_1c_4, c_3, 0\}$ should be distinguishable. When smaller q is used (e.g., $q = 2$), however, these values are not distinguishable. This is called monitoring ambiguity. The monitoring ambiguity increases with the number of paths used for network tomography. The proposed scheme, which is explained in the following sections, does not suffer from the monitoring ambiguity. In linear network coding, the monitoring ambiguity can be avoided with sufficiently large q.

In this paper, we do not quantitatively evaluate the performance of loss tomography because a contribution of the paper is not to show advantages of the proposed scheme against network coding, but to show a mechanism to avoid the monitoring ambiguity problem. In fact, the proposed scheme has the same performance with linear network coding with sufficiently large q.

3.2 Probe Packet Format

Let $\mathcal{P}_{i,j}$ $(i = 1, 2, \ldots, N_\mathrm{S},\ j = 1, 2, \ldots, N_\mathrm{R},\ \mathcal{P}_{i,j} \subset \mathcal{P})$ denote sets of paths from the ith source node s_i to the jth receiver node r_j. Therefore, there are $N_\mathrm{path} = \sum_{i=1}^{N_\mathrm{S}} \sum_{j=1}^{N_\mathrm{R}} |\mathcal{P}_{i,j}|$ paths in total, on which probe packets are transmitted.

In the proposed scheme, probe packet $\boldsymbol{X}$ is represented with N_path bits. Let $\boldsymbol{x}_{i,j}$ $(i = 1, 2, \ldots, N_\mathrm{S},\ j = 1, 2, \ldots, N_\mathrm{R})$ denote a $1 \times |\mathcal{P}_{i,j}|$ vector whose kth $(k = 1, 2, \ldots, |\mathcal{P}_{i,j}|)$ component $x_{i,j}^{(k)}$ is defined as

$$x_{i,j}^{(k)} = \begin{cases} 1, & \text{a packet is delivered successfully} \\ & \quad \text{on the } k\text{th path between } s_i \text{ and } r_j, \\ 0, & \text{otherwise.} \end{cases}$$

With those, probe packet $\boldsymbol{X}$ is given by

$$\boldsymbol{X} = (\boldsymbol{x}_{1,1}\ \boldsymbol{x}_{1,2}\ \cdots\ \boldsymbol{x}_{1,N_\mathrm{R}}\ \boldsymbol{x}_{2,1}\ \cdots\ \boldsymbol{x}_{N_\mathrm{S},N_\mathrm{R}}).$$

If a packet is lost on an input link to node $v \in \mathcal{V}$, we consider that v receives packet $\boldsymbol{X} = (0\ 0\ \cdots\ 0)$ for simplicity in description.

3.3 Description of Bitwise Operations

Let $\boldsymbol{X}_m^{(s)}$ $(m = 1, 2, \ldots,\ s \in \mathcal{V}_\mathrm{S})$ denote the mth probe packet generated at source node s. Let $\boldsymbol{Y}_m^{(v,k)}$ $(m = 1, 2, \ldots,\ v \in \mathcal{V} \setminus \{\mathcal{V}_\mathrm{S} \cup \mathcal{V}_\mathrm{R}\},\ k = 1, 2, \ldots, \mathrm{indegree}(v))$ denote the mth probe packet received at the kth input link of intermediate node v. Intermediate node v first merges received packets $\boldsymbol{Y}_m^{(v,k)}$ $(k = 1, 2, \ldots, \mathrm{indegree}(v))$ into the coded packet and then forwards a bit-masked coded packet to each output link. Note that intermediate nodes never encode probe packets $\boldsymbol{Y}_m^{(v,k)}$ and $\boldsymbol{Y}_{m'}^{(v,k')}$ $(m \neq m')$ in different generations. Let $\boldsymbol{Y}_m^{(r)}$ $(m = 1, 2, \ldots,\ r \in \mathcal{V}_\mathrm{R})$ denote the mth packet received at receiver node r. Finally, $\boldsymbol{Y}_m^{(r)}$ $(r = 1, 2, \ldots, N_\mathrm{R})$ are encoded into an *aggregated packet* $\boldsymbol{Z}_m$.

3.3.1 Node Operations

We first describe the format of a bitmask assigned to each output link. We define $\boldsymbol{b}_{i,j}$ $(i = 1, 2, \ldots, N_\mathrm{S},\ j = 1, 2, \ldots, N_\mathrm{R})$ as a $1 \times |\mathcal{P}_{i,j}|$ vector:

$$\boldsymbol{b}_{i,j} = (b_{i,j}^{(1)}\ b_{i,j}^{(2)}\ \cdots\ b_{i,j}^{|\mathcal{P}_{i,j}|}),$$

whose kth $(k = 1, 2, \ldots, |\mathcal{P}_{i,j}|)$ component $b_{i,j}^{(k)}$ is given by

$$b_{i,j}^{(k)} = \begin{cases} 1, & \text{the output link is on the } k\text{th path in } \mathcal{P}_{i,j}, \\ 0, & \text{otherwise.} \end{cases}$$

The bitmask $\boldsymbol{B}$ is a 0-1 vector of N_path bits, which is defined as

$$\boldsymbol{B} = (\boldsymbol{b}_{1,1}\ \boldsymbol{b}_{1,2}\ \cdots\ \boldsymbol{b}_{1,N_\mathrm{R}}\ \boldsymbol{b}_{2,1}\ \cdots\ \boldsymbol{b}_{N_\mathrm{S},N_\mathrm{R}}).$$

In what follows, $\boldsymbol{B}^{(s)}$ $(s \in \mathcal{V}_\mathrm{S})$ represents the bitmask assigned to the output link of source node s and $\boldsymbol{B}^{(v,l)}$ $(v \in \mathcal{V} \setminus \{\mathcal{V}_\mathrm{S} \cup \mathcal{V}_\mathrm{R}\},\ l = 1, 2, \ldots, \mathrm{outdegree}(v))$ represents the lth output link of intermediate node v.

Source node s $(s \in \mathcal{V}_\mathrm{S})$ transmits probe packets $\boldsymbol{X}_m$ $(m = 1, 2, \ldots)$, where $\boldsymbol{X}_m = \boldsymbol{B}^{(s)}$ for all m. As shown in Fig. 4, this might be considered that source node s first generates $\boldsymbol{X}_m^{(s)} = (1\ 1\ \cdots\ 1)$, and then forwards packet $\boldsymbol{B}^{(s)} \otimes \boldsymbol{X}_m^{(s)}$ to its output link, where ⊗ stands for AND operation.

Intermediate node v $(v \in \mathcal{V} \setminus \{\mathcal{V}_\mathrm{S} \cup \mathcal{V}_\mathrm{R}\})$ generates packets $\boldsymbol{W}_m^{(v,l)}$ $(m = 1, 2, \ldots,\ l = 1, 2, \ldots, \mathrm{outdegree}(v))$ by bitwise operations of $\boldsymbol{Y}_m^{(v,k)}$ $(k = 1, 2, \ldots, \mathrm{indegree}(v))$ and $\boldsymbol{B}^{(v,l)}$:

$$\boldsymbol{W}_m^{(v,l)} = \boldsymbol{B}^{(v,l)} \otimes \left(\boldsymbol{Y}_m^{(v,1)} \oplus \boldsymbol{Y}_m^{(v,2)} \oplus \cdots \oplus \boldsymbol{Y}_m^{(v,\mathrm{indegree}(v))}\right),$$

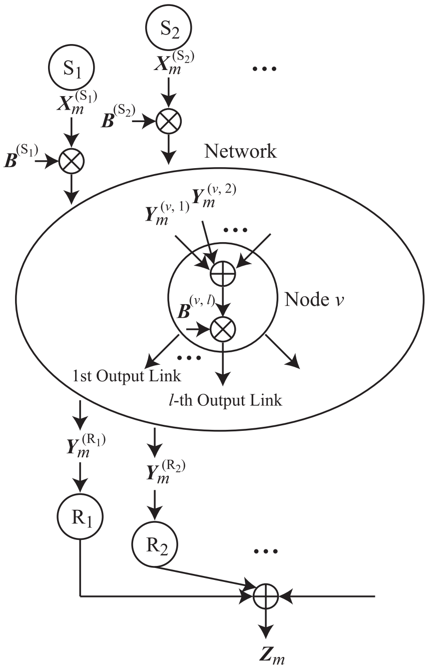


Fig. 4 Node operations.

where ⊕ stands for OR operation. Node v forward $\boldsymbol{W}_m^{(v,l)}$ ($m = 1, 2, \ldots, l = 1, 2, \ldots, \text{outdegree}(v)$) to its lth output link. We gather packets $\boldsymbol{Y}_m^{(r)}$ ($r \in \mathcal{V}_{\text{R}}$) reached at receiver nodes and obtain the aggregated packet $\boldsymbol{Z}_m = \boldsymbol{Y}_m^{(r_1)} \oplus \boldsymbol{Y}_m^{(r_2)} \oplus \cdots \oplus \boldsymbol{Y}_m^{(r_{N_{\text{R}}})}$.

Consider practical network environments that probe packets asynchronously arrive at each intermediate node. In such a situation, in in-network processing approaches such as network coding-based schemes and the proposed scheme, each intermediate node cannot definitely distinguish lost packets and delayed packets. The forwarding timer [19] is a simple scheme to solve this problem. Namely, when the node receives a probe packet, it sets a forwarding timer and waits for other probe packets to arrive. The node encodes probe packets before the timer expires, and forward the coded packets to their output links. In this paper, we assume that the forwarding timer is perfectly operated at each intermediate node, and do not consider the forwarding timer problem because it is beyond the scope of the paper.

3.3.2 Example

We consider an example of the proposed scheme with Fig. 3. Let $\mathcal{P} = \{p_i \mid i = 1, 2, \ldots, 6\}$ denote the set of paths from the source nodes to the receiver nodes, where

$$\begin{aligned} p_1 &= \text{S}_1 \to \text{A} \to \text{D} \to \text{R}_1, \\ p_2 &= \text{S}_1 \to \text{A} \to \text{C} \to \text{D} \to \text{R}_1, \\ p_3 &= \text{S}_1 \to \text{A} \to \text{C} \to \text{E} \to \text{R}_2, \\ p_4 &= \text{S}_2 \to \text{B} \to \text{C} \to \text{D} \to \text{R}_1, \\ p_5 &= \text{S}_2 \to \text{B} \to \text{C} \to \text{E} \to \text{R}_2, \\ p_6 &= \text{S}_2 \to \text{B} \to \text{E} \to \text{R}_2. \end{aligned} \tag{1}$$

Table 1 shows bitmasks assigned to the output links.

Table 1 Bitmasks corresponding to the network topology in Fig. 3.

Link	Bitmask	Link	Bitmask
(S$_1$, A)	$\boldsymbol{B}^{(\text{S}_1)}$ =(1 1 1 0 0 0)	(B, E)	$\boldsymbol{B}^{(\text{B},2)}$=(0 0 0 0 0 1)
(S$_2$, B)	$\boldsymbol{B}^{(\text{S}_2)}$ =(0 0 0 1 1 1)	(C, D)	$\boldsymbol{B}^{(\text{C},1)}$=(0 1 0 1 0 0)
(A, D)	$\boldsymbol{B}^{(\text{A},1)}$=(1 0 0 0 0 0)	(C, E)	$\boldsymbol{B}^{(\text{C},2)}$=(0 0 1 0 1 0)
(A, C)	$\boldsymbol{B}^{(\text{A},2)}$=(0 1 1 0 0 0)	(D, R$_1$)	$\boldsymbol{B}^{(\text{D},1)}$=(1 1 0 1 0 0)
(B, C)	$\boldsymbol{B}^{(\text{B},1)}$=(0 0 0 1 1 0)	(E, R$_2$)	$\boldsymbol{B}^{(\text{E},1)}$=(0 0 1 0 1 1)

Suppose that a probe packet is lost on link (S$_1$, A). In this case, nodes R$_1$ and R$_2$ receive packets (0 0 0 1 0 0) and (0 0 0 0 1 1), respectively. Therefore, we obtain the aggregated packet (0 0 0 1 1 1), which indicates that packets losses occur on paths p_1, p_2, and p_3.

4. Computational Procedure for Likelihood Function

In standard statistical inference of link loss rates from path loss information, likelihood function $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$ plays a central role, where $\boldsymbol{\omega} = (\omega_1\ \omega_2\ \cdots\ \omega_{N_{\text{path}}})$ denotes a realization of an aggregated packet $\boldsymbol{Z}_m$ and $\boldsymbol{\alpha}$ denotes the vector of link loss probabilities. Let $\boldsymbol{T}$ denote an $N_{\text{path}} \times L$ matrix whose (i, j)th component $T_{i,j}$ is given by

$$T_{i,j} = \begin{cases} 1, & \text{link } e_j \text{ is included in path } p_i, \\ 0, & \text{otherwise.} \end{cases}$$

Note that the jth ($j = 1, 2, \ldots, L$) column of $\boldsymbol{T}$ is identical to the transposed bitmask for the jth link e_j.

If there exists a subset $\mathcal{E}_{\text{logical}}$ of $\mathcal{E}$ such that $|\mathcal{E}_{\text{logical}}| \geq 2$ and links in $\mathcal{E}_{\text{logical}}$ have the same bitmask, losses on links in $\mathcal{E}_{\text{logical}}$ are not distinguishable from path loss information. In fact, losses on any links in $\mathcal{E}_{\text{logical}}$ lead to simultaneous losses on paths in $\mathcal{P}(\mathcal{E}_{\text{logical}})$, where $\mathcal{P}(\mathcal{E}_{\text{logical}})$ denotes a set of paths including links in $\mathcal{E}_{\text{logical}}$. We thus treat links in $\mathcal{E}_{\text{logical}}$ as a single logical link and reconstruct the set $\boldsymbol{E}$ of links and matrix $\boldsymbol{T}$. In what follows, we assume that all columns of $\boldsymbol{T}$ are different from each other.

4.1 Recursive formula for $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$

In this subsection, we provide a recursive procedure for computing $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$. For a given path loss information $\boldsymbol{\omega} \in \{0, 1\}^{N_{\text{path}}}$, we divide the set $\mathcal{P}$ of paths into two disjoint subsets $\mathcal{P}_0$ and $\mathcal{P} \setminus \mathcal{P}_0$ based on $\boldsymbol{\omega}$, where

$$\mathcal{P}_0 = \{p_i \in \mathcal{P};\ \omega_i = 0\}. \tag{2}$$

Similarly, for a given $\boldsymbol{\omega} \in \{0, 1\}^{N_{\text{path}}}$, we divide the set $\mathcal{E}$ of links into two disjoint subsets $\mathcal{E}_0$ and $\overline{\mathcal{E}}_0 = \mathcal{E} \setminus \mathcal{E}_0$, where link e_j is included in $\overline{\mathcal{E}}_0$ iff a probe packet is transmitted successfully on link e_j. $\mathcal{E}_0$ can be obtained in the following way.

We define f_j ($j = 1, 2, \ldots, L$) as

$$\begin{aligned} f_j \ = \ & (\omega_1 \otimes T_{1,j}) \oplus (\omega_2 \otimes T_{2,j}) \oplus \cdots \\ & \oplus (\omega_{N_{\text{path}}} \otimes T_{N_{\text{path}},j}). \end{aligned} \tag{3}$$

By definition, $\omega_i \otimes T_{i,j} = 1$ iff a probe packet on path p_i including link e_j is transmitted successfully. Therefore $f_j =$

1 indicates that a probe packet is transmitted successfully on link e_j and $f_j = 0$ indicates that all probe packets transmitted on paths including e_j are lost. We thus have

$$\mathcal{E}_0 = \{e_j \in \mathcal{E};\ f_j = 0\}. \tag{4}$$

With $\mathcal{P}_0$ and $\mathcal{E}_0$, we can rewrite $\boldsymbol{T}$ to be

$$\boldsymbol{T} = \begin{array}{c} \\ \mathcal{P}_0 \\ \mathcal{P}\setminus\mathcal{P}_0 \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{E}_0 & \mathcal{E}\setminus\mathcal{E}_0 \end{array} \\ \begin{pmatrix} \boldsymbol{T}(\mathcal{P}_0,\mathcal{E}_0) & \boldsymbol{T}_0 \\ \boldsymbol{O} & \boldsymbol{T}_1 \end{pmatrix} \end{array}.$$

Because a probe packet is lost on link $e_j \in \mathcal{E}$ with probability α_j, likelihood function $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$ is given by

$$\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha}) = \left(\prod_{\substack{j=1 \\ e_j \in \overline{\mathcal{E}}_0}}^{L} (1-\alpha_j) \right) \cdot r(\mathcal{P}_0, \mathcal{E}_0), \tag{5}$$

where $r(\mathcal{P}_0, \mathcal{E}_0)$ denotes the conditional probability that probe packets on all paths in $\mathcal{P}_0$ are lost given that those packets are not lost on any links in $\overline{\mathcal{E}}_0[\mathcal{P}_0] = \mathcal{E}[\mathcal{P}_0] \setminus \mathcal{E}_0$, where

$$\mathcal{E}[\mathcal{P}_0] = \{e_j;\ \exists p_i \in \mathcal{P}_0, p_i \text{ includes link } e_j\}.$$

Therefore, the computation of likelihood function $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$ is reduced to that of the conditional probability $r(\mathcal{P}_0, \mathcal{E}_0)$, whose computational procedure is discussed below.

For a given nonempty subset $\mathcal{P}_{\text{sub}}$ of $\mathcal{P}$, we choose link e_j in $\mathcal{E}_{\text{sub}}$ arbitrarily and evaluate $r(\mathcal{P}_{\text{sub}}, \mathcal{E}_{\text{sub}})$ by conditioning two exclusive events that a probe packet is transmitted successfully on link e_j (with probability $1-\alpha_i$) and that it is lost in link e_j (with probability α_i). We then have

$$\begin{aligned} r(\mathcal{P}_{\text{sub}}, \mathcal{E}_{\text{sub}}) &= (1-\alpha_j) r(\mathcal{P}_{\text{sub}}, \mathcal{E}_{\text{sub}}(e_j)) \\ &\quad + \alpha_j r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}_{\text{sub}}(e_j)), \end{aligned} \tag{6}$$

where

$$\mathcal{P}_{\text{sub}}(e_j) = \mathcal{P}_{\text{sub}} \setminus \{p_i;\ p_i \in \mathcal{P}_{\text{sub}}, e_j \in p_i\}, \tag{7}$$

$$\mathcal{E}_{\text{sub}}(e_j) = \mathcal{E}_{\text{sub}} \setminus \{e_j\}. \tag{8}$$

The first term on the right hand side of (6) represents the case that the probe packet is transmitted successfully in link e_j and in this case, probe packets on all paths in $\mathcal{P}_{\text{sub}}$ should be lost due to packet losses on some links in $\mathcal{E}_{\text{sub}}(e_j)$. As in Sect. 3.3.1, with coding operations at each intermediate node, probe packets transmitted on link e_j are shared by paths including link e_j. Therefore, the second term represents the case that the probe packet is lost in link e_j, so that the probe packets on all paths in $\mathcal{P}_{\text{sub}} \setminus \mathcal{P}_{\text{sub}}(e_j)$ are lost.

Associated with $r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}_{\text{sub}}(e_j))$, we define $\mathcal{E}^*_{\text{sub}}(e_j)$ as a maximum subset of $\mathcal{E}_{\text{sub}}(e_j)$ such that

$$\begin{aligned} &\exists e_k \in \mathcal{E}^*_{\text{sub}}(e_j) \subset \mathcal{E}_{\text{sub}}(e_j) \\ &\qquad \Rightarrow \exists p_i \in \mathcal{P}_{\text{sub}}(e_j),\ p_i \text{ includes link } e_k. \end{aligned}$$

In other words, if there exists $e_k \in \mathcal{E}_{\text{sub}}(e_j) \setminus \mathcal{E}^*_{\text{sub}}(e_j)$, any $p_i \in \mathcal{P}_{\text{sub}}(e_j)$ do not include e_k. Therefore the status of links in $\mathcal{E}_{\text{sub}}(e_j) \setminus \mathcal{E}^*_{\text{sub}}(e_j)$ does not affect $r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}_{\text{sub}}(e_j))$, so that

$$r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}_{\text{sub}}(e_j)) = r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}^*_{\text{sub}}(e_j)).$$

As a result, we have for $\mathcal{P}_{\text{sub}} \neq \emptyset$ and $\mathcal{E}_{\text{sub}} \neq \emptyset$,

$$\begin{aligned} r(\mathcal{P}_{\text{sub}}, \mathcal{E}_{\text{sub}}) &= (1-\alpha_j) r(\mathcal{P}_{\text{sub}}, \mathcal{E}_{\text{sub}}(e_j)) \\ &\quad + \alpha_j r(\mathcal{P}_{\text{sub}}(e_j), \mathcal{E}^*_{\text{sub}}(e_j)), \end{aligned} \tag{9}$$

where the boundary conditions are given by

$$\begin{aligned} r(\emptyset, \mathcal{E}_{\text{sub}}) &= 1, \\ r(\mathcal{P}_{\text{sub}}, \emptyset) &= 0, \qquad \mathcal{P}_{\text{sub}} \neq \emptyset. \end{aligned}$$

Therefore, applying (9) recursively to $r(\mathcal{P}_0, \mathcal{E}_0)$ and using (5), we can evaluate likelihood function $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$.

Even though the selection of e_j in (9) does not affect the final result, the computational cost depends on it. Generally speaking, it would be efficient to make $|\mathcal{P}_{\text{sub}}(e_j)|$ small as much as possible, which will be demonstrated below.

4.2 Example

Consider the example in Fig. 5, where $\boldsymbol{T}$ is a 6×10 matrix:

$$\boldsymbol{T} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array},$$

where paths are labeled as in (1). We assume $\boldsymbol{\omega} = (0\,0\,1\,0\,1\,1)$, i.e., R_1 does not receive the probe packet, while R_2 receives the probe packet indicating that probe packets are transmitted successfully on all three paths p_3, p_5, and p_6 to R_2. We then have from (2)

$$\mathcal{P}_0 = \{p_1, p_2, p_4\}. \tag{10}$$

Applying (3) to the above $(\boldsymbol{T}, \boldsymbol{\omega})$, we obtain

$$f_j = \begin{cases} 1, & j = 1, 2, 3, 4, 6, 8, 10, \\ 0, & j = 5, 7, 9. \end{cases}$$

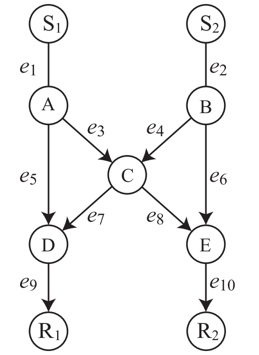


Fig. 5 An example of network tomography.

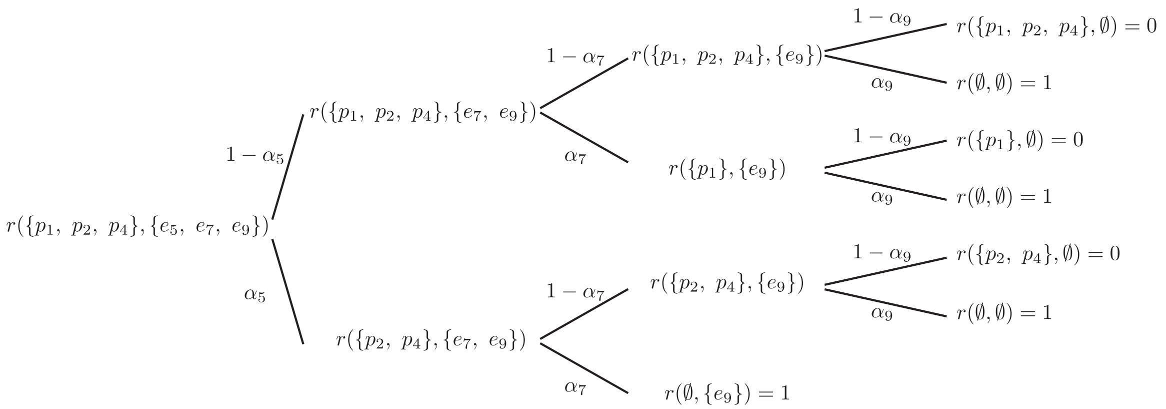

$r(\{p_1,\ p_2,\ p_4\},\{e_5,\ e_7,\ e_9\})$
$1-\alpha_5$
α_5
$r(\{p_1,\ p_2,\ p_4\},\{e_7,\ e_9\})$
$r(\{p_2,\ p_4\},\{e_7,\ e_9\})$
$1-\alpha_7$
α_7
$1-\alpha_7$
α_7
$r(\{p_1,\ p_2,\ p_4\},\{e_9\})$
$r(\{p_1\},\{e_9\})$
$r(\{p_2,\ p_4\},\{e_9\})$
$r(\emptyset,\{e_9\})=1$
$1-\alpha_9$
α_9
$1-\alpha_9$
α_9
$1-\alpha_9$
α_9
$r(\{p_1,\ p_2,\ p_4\},\emptyset)=0$
$r(\emptyset,\emptyset)=1$
$r(\{p_1\},\emptyset)=0$
$r(\emptyset,\emptyset)=1$
$r(\{p_2,\ p_4\},\emptyset)=0$
$r(\emptyset,\emptyset)=1$

Fig. 6 Weighted binary tree (ascending case).

We thus have from (4)

$$\mathcal{E}_0 = \{e_5, e_7, e_9\}. \tag{11}$$

It then follows from (5) that

$$\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha}) = \left(\prod_{i \in \{1,2,3,4,6,8,10\}} (1-\alpha_i) \right) \cdot r(\mathcal{P}_0, \mathcal{E}_0),$$

where $\mathcal{P}_0$ and $\mathcal{E}_0$ are given by (10) and (11), respectively.

In applying (9) to $r(\mathcal{P}_0, \mathcal{E}_0)$ recursively, we have freedom of the order of links chosen from $\mathcal{E}_0 = \{e_5, e_7, e_9\}$. In what follows, we consider two cases: (i) links are chosen in an ascending order of their indices, and (ii) links are chosen in an descending order of their indices.

(i) Ascending order case: $e_5 \to e_7 \to e_9$.

To understand the operation in (9) well, we introduce a submatrix of $\boldsymbol{T}$. Specifically, for a given nonempty subset $\mathcal{P}_{\rm sub}$ of $\mathcal{P}$ and nonempty subset $\mathcal{E}_{\rm sub}$ of $\mathcal{E}$, we define $\boldsymbol{T}(\mathcal{P}_{\rm sub}, \mathcal{E}_{\rm sub})$ as a $|\mathcal{P}_{\rm sub}| \times |\mathcal{E}_{\rm sub}|$ matrix whose (i, j)th component $T_{i,j}(\mathcal{P}_{\rm sub}, \mathcal{E}_{\rm sub})$ is given by

$$T_{i,j}(\mathcal{P}_{\rm sub}, \mathcal{E}_{\rm sub}) = \begin{cases} 1, & \text{link } e_j \text{ in } \mathcal{E}_{\rm sub} \text{ is included} \\ & \qquad \text{in path } p_i \in \mathcal{P}_{\rm sub}, \\ 0, & \text{otherwise.} \end{cases}$$

We then have

$$\boldsymbol{T}(\mathcal{P}_0, \mathcal{E}_0) = \begin{array}{c} \\ p_1 \\ p_2 \\ p_4 \end{array} \begin{array}{c} \begin{array}{ccc} e_5 & e_7 & e_9 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}. \tag{12}$$

From (7), we have $\mathcal{P}_0(e_5) = \{p_2, p_3\}$, which can be identified readily from (12), i.e., only p_1 includes e_5, so that we obtain $\mathcal{P}_0(e_5)$ by deleting p_1 from $\mathcal{P}_0$. On the other hand, we have $\mathcal{E}_0(e_5) = \{e_7, e_9\}$, and therefore

$$\boldsymbol{T}(\mathcal{P}_0, \mathcal{E}_0(e_5)) = \begin{array}{c} \\ p_1 \\ p_2 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cc} e_7 & e_9 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array},$$

$$\boldsymbol{T}(\mathcal{P}_0(e_5), \mathcal{E}_0(e_5)) = \begin{array}{c} \\ p_2 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cc} e_7 & e_9 \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array}. \tag{13}$$

It is easy to see from (13) that $\mathcal{E}_0^*(e_5) = \mathcal{E}_0(e_5)$ because $\boldsymbol{T}(\mathcal{P}_0(e_5), \mathcal{E}_0(e_5))$ contains no columns whose components are all equal to zero. As a result, we have

$$\begin{aligned} r(\mathcal{P}_0, \mathcal{E}_0) &= (1-\alpha_5) r(\{p_1, p_2, p_4\}, \{e_7, e_9\}) \\ &\qquad + \alpha_5 r(\{p_2, p_4\}, \{e_7, e_9\}). \end{aligned}$$

Next we apply (9) with e_7 to $r(\{p_1, p_2, p_4\}, \{e_7, e_9\})$ and $r(\{p_2, p_4\}, \{e_7, e_9\})$, and obtain

$$\begin{aligned} & r(\{p_1, p_2, p_4\}, \{e_7, e_9\}) = (1-\alpha_7) r(\{p_1, p_2, p_4\}, \{e_9\}) \\ & \qquad + \alpha_7 r(\{p_1\}, \{e_9\}), \\ & r(\{p_2, p_4\}, \{e_7, e_9\}) = (1-\alpha_7) r(\{p_2, p_4\}, \{e_9\}) \\ & \qquad + \alpha_7 r(\emptyset, \{e_9\}) \\ & \quad = (1-\alpha_7) r(\{p_2, p_4\}, \{e_9\}) + \alpha_7. \end{aligned}$$

We repeat such a procedure recursively until the values of all $r(.)$ on the right hand side are determined. See Fig. 6, where the recursive operation with (9) is represented in a form of a weighted binary tree. By traversing the tree from the root to leaves, we obtain $r(\mathcal{P}_0, \mathcal{E}_0) = r(\{p_1, p_2, p_4\}, \{e_5, e_7, e_9\})$ to be

$$\begin{aligned} r(\mathcal{P}_0, \mathcal{E}_0) &= (1-\alpha_5)(1-\alpha_7)\alpha_9 + (1-\alpha_5)\alpha_7\alpha_9 \\ &\qquad + \alpha_5(1-\alpha_7)\alpha_9 + \alpha_5\alpha_7 \\ &= \alpha_9 + \alpha_5\alpha_7(1-\alpha_9) \end{aligned} \tag{14}$$

It is now clear that the depth of the weighted binary tree representing the recursive operation in (9) is equal to $|\mathcal{E}_0|$ at most. We also notice that the recursive operation will be terminated if $r(\emptyset, \mathcal{E}_{\rm sub})$ appears. In other others, the computational cost of (9) is proportional to the number of leaves in the weighted binary tree, and the reduction of the number of leaves is beneficial. This suggests that we may select e_j in (9) in such a way that $|\mathcal{P}_{\rm sub}(e_j)|$ is minimized. In the current example, this strategy corresponds to the selection of links in a descending order of their indices.

(ii) Descending order case: $e_9 \to e_7 \to e_5$.

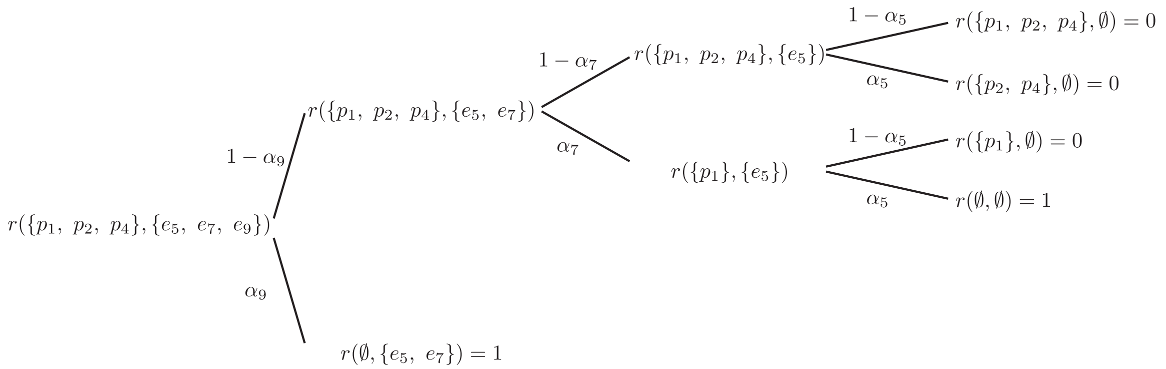

$r(\{p_1,\ p_2,\ p_4\},\{e_5,\ e_7,\ e_9\})$
$1-\alpha_9$
α_9
$r(\{p_1,\ p_2,\ p_4\},\{e_5,\ e_7\})$
$r(\emptyset,\{e_5,\ e_7\})=1$
$1-\alpha_7$
α_7
$r(\{p_1,\ p_2,\ p_4\},\{e_5\})$
$r(\{p_1\},\{e_5\})$
$1-\alpha_5$
α_5
$r(\{p_1,\ p_2,\ p_4\},\emptyset)=0$
$r(\{p_2,\ p_4\},\emptyset)=0$
$r(\{p_1\},\emptyset)=0$
$r(\emptyset,\emptyset)=1$

Fig. 7 Weighted binary tree (descending case).

From (12), we observe that the selection of e_9 seems to be most effective. In fact, applying (9) with e_9 to $r(\mathcal{P}_0,\mathcal{E}_0)$ yields

$$\begin{aligned} r(\mathcal{P}_0,\mathcal{E}_0) &= (1-\alpha_9)r(\{p_1,p_2,p_4\},\{e_5,e_7\}) \\ &\qquad + \alpha_9 r(\emptyset,\{e_5,e_7\}) \\ &= (1-\alpha_9)r(\{p_1,p_2,p_4\},\{e_5,e_7\})+\alpha_9. \end{aligned}$$

Next, we apply (9) to $r(\{p_1,p_2,p_4\},\{e_5,e_7\})$. Note here that

$$\boldsymbol{T}(\{p_1,p_2,p_4\},\{e_5,e_7\}) = \begin{array}{c} \\ p_1 \\ p_2 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cc} e_5 & e_7 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \end{array},$$

and therefore we choose e_7, We then perform the procedure in (9) recursively, and finally obtain the weighted binary tree shown in Fig. 7. By traversing the tree from the root to leaves, we recover $r(\mathcal{P}_0,\mathcal{E}_0)$ given in (14). Comparing Fig. 7 with Fig. 6, we confirm that the computational cost in the descending case is smaller that that in the ascending case.

5. Loss Tomography Examples

In this section, we show how the result in this paper is utilized in the loss tomography. For this purpose, we consider two loss tomography examples: Maximum likelihood estimation (MLE) [7], [8] and sequential loss tomography based on MAP (Maximum a Posteriori) estimation [10], both of which require the evaluation of likelihood function. MLE is suitable for networks with stationary link loss rates, while sequential loss tomography is suitable for networks with dynamic link loss rates. In this paper, we do not discuss numerical results obtained from these examples, because the proposed scheme can achieve the same performance with network coding-based schemes defined on a sufficiently large finite field, and network coding-based schemes have been evaluated so far. Interested reader may refer to the literature such as [8], where network coding-based loss tomography schemes are evaluated quantitatively.

5.1 Maximum Likelihood Estimation

We define $\mathcal{Z} = \{\boldsymbol{Z}_m \mid m = 1,2,\dots,N_{\mathrm{pkt}}\}$ as the set of N_{pkt} aggregated packets and $n(\mathcal{Z},\boldsymbol{\omega})$ ($\boldsymbol{\omega} \in \boldsymbol{\Omega}$) as the number of components in $\mathcal{Z}$ equal to $\boldsymbol{\omega}$, where $\boldsymbol{\Omega} = \{0,1\}^{N_{\mathrm{path}}}$. Given $\mathcal{Z}$, log-likelihood function $\mathcal{L}(\boldsymbol{\alpha})$ is obtained by

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}) &= \log \Pr(\mathcal{Z} \mid \boldsymbol{\alpha}) \\ &= \log \prod_{\boldsymbol{\omega}\in\boldsymbol{\Omega}} \Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})^{n(\mathcal{Z},\boldsymbol{\omega})} \\ &= \sum_{\boldsymbol{\omega}\in\boldsymbol{\Omega}} n(\mathcal{Z},\boldsymbol{\omega}) \log \Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha}). \end{aligned}$$

In the maximum likelihood estimation, link loss rates are estimated by

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}). \tag{15}$$

In solving (15), we have to evaluate $\Pr(\boldsymbol{\omega} \mid \boldsymbol{\alpha})$. At this stage, we can utilize the procedure in Sect. 4.

5.2 Sequential Loss Tomography

Figure 8 shows the structure of the sequential loss tomography scheme. Each source node successively generates probe packets and injects them into the network. Link loss rates are estimated with every N_{pkt} aggregated packets. We refer to a period that each set of N_{pkt} aggregated packets are collected as a *cycle*. We define $\mathcal{Z}_m = \{\boldsymbol{Z}_{(m-1)N_{\mathrm{pkt}}+1}, \boldsymbol{Z}_{(m-1)N_{\mathrm{pkt}}+2}, \dots, \boldsymbol{Z}_{mN_{\mathrm{pkt}}}\}$ as the set of aggregated packets collected within the m-th cycle.

Let $\hat{\boldsymbol{\alpha}}^{(m)} = (\hat{\alpha}_1^{(m)}\ \hat{\alpha}_2^{(m)}\ \cdots\ \hat{\alpha}_L^{(m)})$ denote the vector of link loss rates estimated after the m-th cycle. The sequential loss tomography is based on a MAP estimation, where $\hat{\boldsymbol{\alpha}}^{(m)}$ is obtained by $\mathcal{Z}_m$ and the prior distribution of $\boldsymbol{\alpha}$. Given $\{\mathcal{Z}_k;\ k = 1,2,\dots,m\}$, the posterior distribution $\Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k = 1,2,\dots,m\})$ is given by

$$\begin{aligned} &\Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k = 1,2,\dots,m\}) \\ &\quad = \Pr(\boldsymbol{\alpha} \mid \mathcal{Z}_m, \{\mathcal{Z}_k;\ k = 1,2,\dots,m-1\}) \\ &\quad \propto \Pr(\mathcal{Z}_m \mid \boldsymbol{\alpha}, \{\mathcal{Z}_k;\ k = 1,2,\dots,m-1\}) \\ &\qquad\quad \cdot \Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k = 1,2,\dots,m-1\}) \\ &\quad = \Pr(\mathcal{Z}_m \mid \boldsymbol{\alpha}) \Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k = 1,2,\dots,m-1\}). \end{aligned}$$

In MAP estimation, $\hat{\boldsymbol{\alpha}}^{(m)}$ is obtained by solving the following optimization problem:

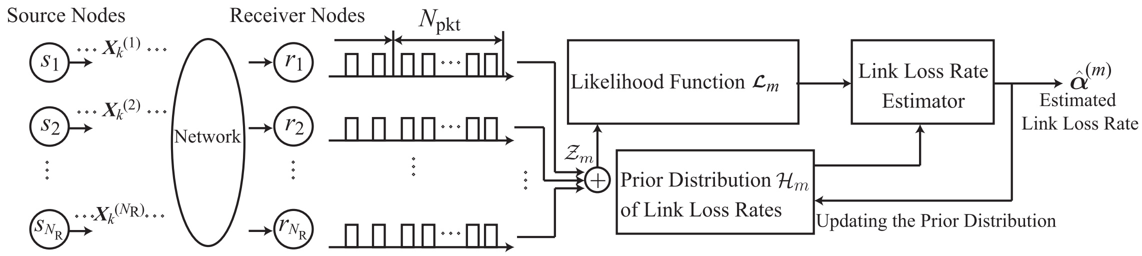

Source Nodes
Receiver Nodes
N_{pkt}
s_1
$\cdots X_k^{(1)} \cdots$
s_2
$\cdots X_k^{(2)} \cdots$
s_{N_R}
$\cdots X_k^{(N_\mathrm{R})} \cdots$
Network
r_1
r_2
r_{N_R}
Likelihood Function $\mathcal{L}_m$
Link Loss Rate Estimator
$\hat{\boldsymbol{\alpha}}^{(m)}$
Estimated Link Loss Rate
$\mathcal{Z}_m$
Prior Distribution $\mathcal{H}_m$ of Link Loss Rates
Updating the Prior Distribution

Fig. 8 Sequential loss tomography.

$$\begin{aligned}\hat{\boldsymbol{\alpha}}^{(m)} &= \arg\max_{\boldsymbol{\alpha}}\Big[\log \Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k=1,2,\dots,m\})\Big] \\ &= \arg\max_{\boldsymbol{\alpha}}\Big[\mathcal{L}_m(\boldsymbol{\alpha}) \\ &\qquad + \mathcal{H}_m(\boldsymbol{\alpha};\ \{\mathcal{Z}_k;\ k=1,2,\dots,m-1\})\Big], \quad (16)\end{aligned}$$

where $\mathcal{L}_m(\boldsymbol{\alpha})$ denotes a log-likelihood function of $\boldsymbol{\alpha}$,

$$\mathcal{L}_m(\boldsymbol{\alpha}) = \log \Pr(\mathcal{Z}_m \mid \boldsymbol{\alpha}),$$

and $\mathcal{H}_m(\boldsymbol{\alpha};\ \{\mathcal{Z}_k;\ k=1,2,\dots,m-1\})$ denotes a log-prior function of $\boldsymbol{\alpha}$ given by

$$\begin{aligned}&\mathcal{H}_m(\boldsymbol{\alpha};\ \{\mathcal{Z}_k;\ k=1,2,\dots,m-1\}) \\ &\qquad = \log \Pr(\boldsymbol{\alpha} \mid \{\mathcal{Z}_k;\ k=1,2,\dots,m-1\}).\end{aligned}$$

Therefore, in solving (16), we have to evaluate likelihood function $\Pr(\mathcal{Z}_m \mid \boldsymbol{\alpha})$, which can be done with the procedure in Sect. 4.

On the other hand, to evaluate $\mathcal{H}_m(\boldsymbol{\alpha}; \{\mathcal{Z}_k; k = 1,2,\dots,m-1\})$ $(m = 1,2,\dots)$, we may follow an EWMA (Exponentially Weighted Moving Average) approach for binomial distributions, which is given in [10]. For completeness, we outline it in Appendix.

6. Conclusion

In this paper, we proposed the bitwise operation-based in-network processing for loss tomography. The proposed scheme is very simple and does not suffer from the monitoring ambiguity, which is inherent in network tomography with network coding. We also provided a recursion to compute the likelihood function, which is required in conducting standard statistical inference of link loss rates from path loss information.

Future work includes a study on the way of establishing measurement paths in a given network. In general, the accuracy of statistical inference depends on the formation of measurement paths, because it directly affects the amount of information about link loss events, which we can extract from path loss information.

Acknowledgment

This research was supported in part by Grant-in-Aid for Young Scientists (B) of Japan Society for the Promotion of Science under Grant No. 22700069.

References

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network information flow," IEEE Trans. Inf. Theory, vol.46, no.4, pp.1204–1206, July 2000.

[2] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley, "Network tomography on general topologies," Proc. ACM SIGMETRICS'02, pp.21–30, Marina Del Rey, CA, June 2002.

[3] R. Cáceres, N.G. Duffield, J. Horowitz, and D.F. Towsley, "Multicast-based inference of network-internal loss characteristics," IEEE Trans. Inf. Theory, vol.45, no.7, pp.2462–2480, Nov. 1999.

[4] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," IETF RFC 1157, May 1990.

[5] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed network monitoring for IP and all-optical networks," Proc. 7th ACM SIGCOMM Conference on Internet Measurement (IMC'07), pp.241–252, San Diego, CA, Oct. 2007.

[6] M. Coates, A.O. Hero, III, R. Nowak, and B. Yu, "Internet tomography," IEEE Signal Process. Mag., vol.19, no.3, pp.47–65, May 2002.

[7] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," Proc. 43rd Allerton Conf. Commun., Control, and Comput., pp.774–783, Monticello, IL, Sept. 2005.

[8] C. Fragouli, A. Markopoulou, R. Srinivasan, and S. Diggavi, "Network monitoring: It depends on your points of view," Proc. 2007 Inf. Theory and Applicat. Workshop, San Diego, CA, Jan. 2007.

[9] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," Proc. IEEE GLOBECOM 2007, pp.381–386, Washington, DC, Nov. 2007.

[10] A.C. Harvey, Forecasting, Structural Time Series Models and the Kalman Filter, Cambridge University Press, 1989.

[11] T. Ho, B. Leong, Y.-H. Chang, Y. Wen, and R. Koetter, "Network monitoring in multicast networks using network coding," Proc. 2005 IEEE Int'l Symp. Inf. Theory (ISIT 2005), pp.1977–1981, Adelaide, Australia, Sept. 2005.

[12] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, S. Jun, and B. Leong, "A random linear network coding approach to multicast," IEEE Trans. Inf. Theory, vol.52, no.10, pp.4413–4430, Oct. 2006.

[13] R. Koetter and M. Médard, "An algebraic approach to network coding," IEEE/ACM Trans. Netw., vol.11, no.5, pp.782–795, Oct. 2003.

[14] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," IEEE Trans. Inf. Theory, vol.47, no.2, pp.498–519, Feb. 2001.

[15] S.-Y.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," IEEE Trans. Inf. Theory, vol.29, no.2, pp.371–381, Feb. 2003.

[16] Y. Lin, B. Liang, and B. Li, "Passive loss inference in wireless sensor networks based on network coding," Proc. IEEE INFOCOM 2009, pp.1809–1817, Rio de Janeiro, Brazil, April 2009.

[17] Y. Mao, F.R. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," IEEE J. Sel. Areas Commun., vol.23, no.4, pp.820–829, April 2005.

[18] T. Matsuda, T. Noguchi, and T. Takine, "Survey of Network Coding and Its Applications," IEICE Trans. Commun., vol.E94-B, no.3, pp.698–717, March 2011.

[19] K. Minami, T. Matsuda, T. Takine, and T. Noguchi, "Asynchronous multiple source network coding for wireless broadcasting," Numerical Algebra, Control, and Optimization (NACO), vol.1, no.4, pp.577–592, Dec. 2011.

[20] M. Tsuru, T. Takine, and Y. Oie, "Inferring link loss rates from unicast-based end-to-end measurement," IEICE Trans. Commun., vol.E85-B, no.1, pp.70–78, Jan. 2002.

Appendix: Evaluation of Log-Prior Function $\mathcal{H}_m$

Let $B(\alpha;\ a, b)$ denote a beta function with parameter a and b:

$$B(\alpha;\ a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}(1-\alpha)^a \alpha^b,$$

where $\Gamma(\cdot)$ denotes the gamma function. According to the approach in [10, Sect. 6.6.2], $\mathcal{H}_m(\boldsymbol{\alpha};\ \{\mathcal{Z}_k; k = 1, 2, \ldots, m-1\})$ $(m = 1, 2, \ldots)$ is given by

$$\mathcal{H}_m(\boldsymbol{\alpha};\ \{\mathcal{Z}_k; k = 1, 2, \ldots, m-1\}) = \sum_{i=1}^{L} \log B(\alpha_i;\ a_i^{(m)}, b_i^{(m)}).$$

Note that parameters $a_i^{(m)}$ and $b_i^{(m)}$ $(m = 1, 2, \ldots)$ are given by

$$\begin{aligned} a_i^{(1)} &= q_i^{(1)}, \\ a_i^{(m)} &= \gamma a_i^{(m-1)} + q_i^{(m)}, \quad m = 2, 3, \ldots, & \text{(A· 1)} \\ b_i^{(1)} &= N_{\text{pkt}} - q_i^{(1)}, \\ b_i^{(m)} &= \gamma b_i^{(m-1)} + N_{\text{pkt}} - q_i^{(m)}, \quad m = 2, 3, \ldots, & \text{(A· 2)} \end{aligned}$$

respectively, where $q_i^{(m)}$ $(i = 1, 2, \ldots, L,\ m = 1, 2, \ldots)$ denotes the number of packets transmitted successfully on link e_i in the mth cycle and γ $(0 < \gamma \leq 1)$ denotes the weight in EWMA. The detailed derivation of Eqs. (A· 1) and (A· 2) is explained in [10, Sect. 6.6.2].

Because $q_i^{(m)}$ cannot be observed, Eqs. (A· 1) and (A· 2) are not applicable directly. We may approximate $q_i^{(m)}$ by $N_{\text{pkt}}(1 - \hat{\alpha}_i{}^{(m-1)})$. As a result, $a_i^{(m)}$ and $b_i^{(m)}$ are updated as follows:

$$\begin{aligned} a_i^{(m)} &= \gamma a_i^{(m-1)} + N_{\text{pkt}}(1 - \hat{\alpha}_i^{(m-1)}), \\ b_i^{(m)} &= \gamma b_i^{(m-1)} + N_{\text{pkt}} \hat{\alpha}_i^{(m-1)}. \end{aligned}$$

Takahiro Matsuda received his B.E. with honors, M.E., and Ph.D. in communications engineering from Osaka University in 1996, 1997, 1999, respectively. He joined the Department of Communications Engineering at the Graduate School of Engineering, Osaka University in 1999. He is currently an Associate Professor in the Department of Information and Communications Technology, Graduate School of Engineering, Osaka University. His research interests include performance analysis and the design of communication networks and wireless communications. He received Best Tutorial Paper Award and Best Magazine Paper Award from IEICE ComSoc in 2012. He is a member of IPSJ and IEEE.

Tetsuya Takine is currently a Professor in the Department of Information and Communications Technology, Graduate School of Engineering, Osaka University. He was born in Kyoto, Japan, on November 28, 1961. He received B.Eng., M.Eng., and Dr.Eng. degrees in applied mathematics and physics from Kyoto University, Kyoto, Japan, in 1984, 1986, and 1989, respectively. In April 1989, he joined the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, as an Assistant Professor. Beginning in November 1991, he spent one year at the Department of Information and Computer Science, University of California, Irvine, on leave of absence from Kyoto University. In April 1994, he joined the Department of Information Systems Engineering, Faculty of Engineering, Osaka University as a Lecturer, and from December 1994 to March 1998, he was an Associate Professor in the same department. From April 1998 to May 2004, he was an Associate Professor in the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University. His research interests include queueing theory, emphasizing numerical computation, and its application to performance analysis of computer and communication networks. He is now serving as an associate editor of Queueing Systems, Stochastic Models, and International Transactions in Operational Research. He received Telecom System Technology Award from the Telecommunications Advancement Foundation in 2003 and 2010, Best Paper Awards from ORSJ in 1997, from IEICE in 2004 and 2009, and from ISCIE in 2006, Best Tutorial Paper Award and Best Magazine Paper Award from IEICE ComSoc in 2012. Dr. Takine is a fellow of ORSJ and a member of IPSJ, ISCIE, and IEEE.