Network Monitoring: It Depends on your Points of View

Christina Fragouli
EPFL, Lausanne
christina.fragouli@epfl.ch

Athina Markopoulou
University of California, Irvine
athina@uci.edu

Ramya Srinivasan
EPFL, Lausanne
r.srinivasan@student.tue.nl

Suhas Diggavi
EPFL, Lausanne
suhas.diggavi@epfl.ch

***Abstract*— End-to-end active network monitoring infers network characteristics by sending and collecting probe packets from the network edge, while probes traverse the network through multicast trees or a mesh of unicast paths. Most reported methods consider given source and receiver locations and study the path selection and the associated estimation algorithms. In this paper, we show that appropriately choosing the number of sources and receivers, as well as their location, may have a significant effect on the accuracy of the estimation; we also give guidelines on how to choose the best "points of view" of a network for link loss monitoring purposes. Though this observation applies across all monitoring methods, we consider, in particular, networks where nodes are equipped with network coding capabilities; our framework includes as special cases the scenarios of pure multicast and network coding. We show that, in network-coding enabled networks, multiple source active monitoring can exploit these capabilities to estimate link loss rates more efficiently than purely tomographic methods. To address the complexity of the estimation problem for large networks, we also propose efficient algorithms, including the decomposition into smaller multicast inference problems, belief-propagation, and a MINC-like algorithm.**

I. Introduction

Network monitoring is an important component of network engineering. For small-scale networks, local monitoring of link characteristics, such as loss rates, delay and bandwidth, is feasible. However, for large-scale networks, as well as for interconnections of diverse networks with distributed control over them, local monitoring becomes difficult. Therefore, it is desirable to be able to infer network characteristics through end-to-end measurements. Over the past decade, significant progress has been made in inferring network characteristics using end-to-end measurements, also known as tomographic techniques. Most of the tomography work has focused on sending active probes from a single source node through a multicast network and using the probes observed at the receivers to estimate the metric of interest [1]; this work has also been extended to unicast [2] measurements and to multiple sources [3], [4].

In this paper, we are also interested in estimating link loss rates using end-to-end measurements. One aspect we explore is the effect of the placement of sources and receivers on the link-loss estimation. The placement of sources and receivers gives us different "views" of the network: we show that the "points of view" matters in terms of estimation error. This observation is of course applicable to the tomographic methods as well. However, we explore this idea specifically in the context of networks that already have *network coding* functionalities deployed. We show that in such networks, multiple source active monitoring can exploit the network coding capabilities to better estimate the metric of interest, which in our case is the link-loss rates.

Our interest in network-coding enabled networks is motivated by the fact that network coding seems likely to be included in tomorrow's networks. The pioneering work in [5], [6] showed that for multicast networks, if intermediate nodes can do simple local XOR-operations on incoming packets, then one can achieve the min-cut throughput of the network to each receiver. These linearly combined packets can then be utilized at the end-receivers to recover the original information symbols by solving a set of linear equations over a finite field [7]. This breakthrough idea has spawned a significant effort in applying network coding to other network topologies, developing practical algorithms that achieve this performance, as well as quantifying the throughput benefits of network coding [8]. In terms of applications, the network coding idea is well-matched to content distribution over peer-to-peer networks as seen by several ongoing projects for this application [9], [10]. It has also been shown that network coding can bring benefits in multihop wireless networks [11].

Motivated by the fact that, in the future, network coding can be deployed in large scale networks, we explore how we can utilize it for efficient network monitoring. For example, this idea is suited for overlay networks [12] or for multihop wireless networks [11] since (i) performance monitoring is particular important for the control of such networks [13], [14] and (ii) network coding could be deployed incrementally on their nodes (unlike legacy routers). In general, our approach is applicable to any network where network coding is deployed.

This paper builds on our previous work [15], where we first introduced the idea of exploiting network coding to estimate link-loss rates, using a toy example topology. In this paper, we further demonstrate the benefits of this approach: when intermediate nodes linearly combine incoming probes from multiple sources, we can have the benefits of using multiple sources *without* increasing the load on the links. In addition, we investigate the best point of view, i.e., the best choice of sources and receivers. Combining the two, we show that we can obtain faster convergence rates using multiple sources and appropriately chosen views of

the network than when using multicast probes.

Note that the loss rate can be different on the two directions of the same link. We show that, both in the cases of symmetric and asymmetric link-loss rates, we can estimate the loss rates with much fewer "views" of the network, by using multiple sources than having a single source.

The main contributions of this paper are the following. We observe that the placement of sources and receivers are an important aspect of the design of a network monitoring system. We explore this in the context of networks with network coding functionalities and we develop properties of estimators. We propose several computationally efficient algorithms that are suitable for large networks. One method is based on the belief propagation algorithm [16] which has had a lot of success in error correcting codes on graphs. An alternative method is based on decomposing the network into sub-networks and solving multiple multicast inference problems. Overall, we show that we can significantly improve link-loss estimation over previous purely tomographic methods by (i) using multiple sources (ii) appropriately choosing their location and (iii) exploiting the network coding capabilities of the underlying network.

The paper is organized as follows. In Section II we define the problem. In Section III, we study a basic topology, estimate the loss rate of a single link, and illustrate the importance of multiple sources and the network view. In Section IV, we extend these ideas to larger networks and to estimating loss rates for all links; we also give guidelines on how to place the sources and receivers. In Section V, we develop efficient heuristics suitable for large networks and evaluate their performance via simulation. Sections VI and VII provide the discussion and conclusion respectively.

II. Problem Statement

A. Motivating Example

The following example illustrates the use of network coding to infer the loss rate of a single link [15].

Consider the basic topology shown in Fig. 1. In particular consider Case 1, where nodes A and B send probes and nodes E and F receive them. Our goal is to measure the loss rate of the link CD, using probes from A, B and observations at E, F. Node A sends to node C a probe packet with payload that contains the binary string $x_1 = [1\ 0]$. Similarly, the node B sends probe packet $x_2 = [0\ 1]$ to node C. If node C receives only x_1 or only x_2 then it just forwards the received packet to node D; if C receives both packets x_1 and x_2, then it creates a new packet, with payload their linear combination $x_3 = [1\ 1]$, and forwards it to node D; more generally $x_3 = x_1 \otimes x_2$, where $\otimes$ is bitwise XOR operation. Node D sends the incoming packet x_3 to both outgoing links DE and DF. All above operations happen in one " time slot", which is to be defined.

In every time slot, probes (x_1, x_2) are sent from A, B and may reach E, F, depending on a random experiment: on every link in $\{AC, BC, CD, DE, DF\}$, the transmitted packet is lost with probability α_{link}. The possible outcomes

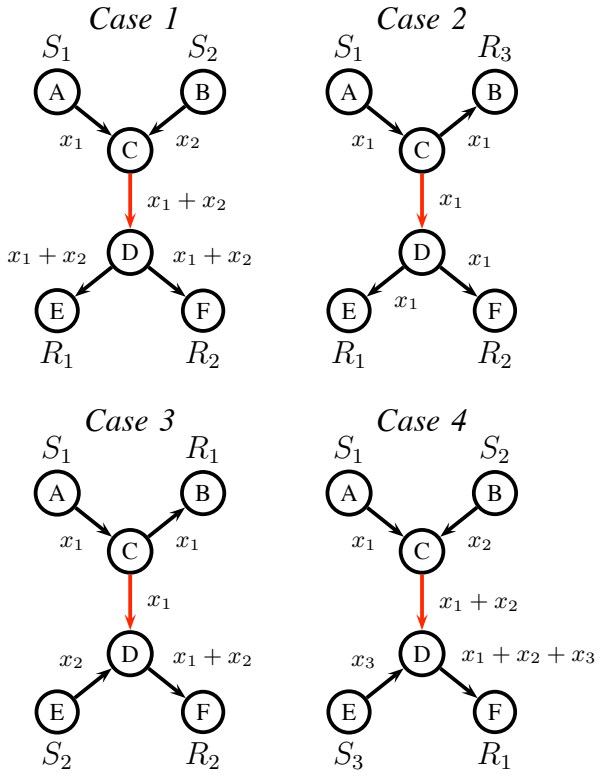


Fig. 1. Basic 5-links topology. Four possible cases (choices of sources and receivers) that allow to calculate the loss rate of link CD, when neither C or D are edge nodes, by sending and receiving probe packets from the edge nodes A, B, E and F.

observed at nodes E and F are summarized in the left two columns of Table I. The five right columns at the same table show the combination of loss and success events on the links that lead to the observed outcome. For example, the outcome (x_1, x_1) is due to the event $(AC = 1, BC = 0, CD = 1, DE = 1, DF = 1)$ and happens with probability $(1 - \alpha_{AC})(\alpha_{BC})(1 - \alpha_{CD})(1 - \alpha_{DE})(1 - \alpha_{DF})$. Similarly, we can write the probability of each of the 10 observed events as a function of the link loss probabilities. The problem then becomes how to estimate α_{CD} from the observations at E, F.

Received at		Is link ok?				
E	F	AC	BC	CD	DE	DF
0	0	Multiple possible events				
x_1	–	1	0	1	1	0
x_2	–	0	1	1	1	0
x_3	–	1	1	1	1	0
–	x_1	1	0	1	0	1
x_1	x_1	1	0	1	1	1
–	x_2	0	1	1	0	1
x_2	x_2	0	1	1	1	1
–	x_3	1	1	1	0	1
x_3	x_3	1	1	1	1	1

TABLE I

Possible observed probes at nodes E and F, together with the combination of loss (0) and success (1) in all five links that led to the observed outcome.

B. Estimation Problem for a General Tree

Consider a network $G = (V, E)$, with nodes V and logical links E. Although we demonstrate our ideas using trees for most of the paper, they are also applicable to general graphs, as discussed in section VI.A. Let each link $e \in E$ have an associated loss rate α_e, which we are interested in estimating.

We assume that probes can be sent in either direction of a link. Each direction is in general associated with a different link loss rate. We will call a network "undirectional" if the loss rate of a link is the same in both directions, and "directional" if the link loss rates are different in opposite directions. Undirectional networks may include for example wireless networks, while directional networks can be duefor example to non-overlapping overlay paths with different congestion levels. For most of the paper, we focus on undirectional networks, and address the case of directional networks in Sec. IV-B.

We inject n probes from each source and make measurements only at the network edge (e.g. at the leaf nodes of the tree). We denote this set of potential sources or receivers as $\mathcal{L} \subset V$, the sources by $\mathcal{S} \subset \mathcal{L}$ and the receivers by $\mathcal{R} \subset \mathcal{L}$. Each set of probes injected, one from every source, corresponds to one experiment, or one timeslot.

We assume that the internal nodes of the network which are neither sources or receivers ($V \backslash (\mathcal{S} \cup \mathcal{R})$) are capable of doing the simplest network coding operation: `XOR` incoming probes before forwarding them to all outgoing links. If a packet is lost, only those packets that arrive at the internal node within a given time-window are linearly combined. Nodes that perform such `XOR` operations are called *coding points*. Note that the number of coding points might be quite small. For example, in a tree topology with two sources, there exists exactly one coding point, irrespective of the number of intermediate nodes since the two probe packet flows from the two sources can meet only once. Generally in a tree with $|\mathcal{S}|$ sources we have at most $|\mathcal{S}| - 1$ coding points.

With network coding, unlike the multicast trees approach, using multiple sources does not increase the required bandwidth: even though $|\mathcal{S}|$ probes are injected into the network, each link carries only the `XOR` of the probes it receives, therefore at most one packet.

Given a certain topology, and choice of sources and receivers, the goal is to infer the link-loss rates $\{\alpha_e\}$. The sets $\mathcal{S}, \mathcal{R}$ should be chosen to give the "view" of the network which yields the best estimation of these parameters.

At the receiver set $\mathcal{R}$, we observe the outcome of sending the probe packets from the source set $\mathcal{S}$. At timeslot (experiment) i, each source sends one probe packet. Each receiver $r \in R$ observes $X_r^{(i)}$, which can either be $\emptyset$, i.e., it receives nothing, or some linear combination of the source packets that is determined by the topology and the loss pattern. Let $\{X_r^{(i)}\}$ denote the set of outcomes at time i for all receivers $r \in R$. If we assume that successive probes experience independent losses after sending n probes we can write the probability of observing a set of events as

$$p(\{X_r^{(1)}\}, \ldots, \{X_r^{(n)}\}; \{\alpha_e\}) = \prod_{i=1}^{n} p(\{X_r^{(i)}\}; \{\alpha_e\}).$$

This probability can also be related to the counts of the various linear combinations received. The maximum likelihood estimate (MLE) of the link-loss probabilities after n observations is

$$\begin{aligned} \{\hat{\alpha}_e\} &= \arg\max \mathtt{L}(\{\alpha_e\}) = \\ &= \arg\max \log p(\{X_r^{(1)}\}, \ldots, \{X_r^{(n)}\}; \{\alpha_e\}). \end{aligned} \tag{1}$$

In later sections, we examine the performance of the MLE as well as of some computationally efficient sub-optimal estimators. For the rest of this paper we will assume that losses occur independently across links. This assumption allows to simplify the expression of our estimator in (1); if it doesnt hold, the expressions become more complicated but the principle approach remains the same.

The quality of the estimation for a single link e is captured by the *mean-squared error* metric, *i.e.,*

$$MSE = \mathbb{E}[|\hat{\alpha}_e - \alpha_e|^2],$$

where $\hat{\alpha}_e$ is the estimator based on the observations on R of sources S, and α_e is the true value of the loss rate on e.

In order to get a measure of performance for the set of estimators across all links $e \in E$, we need a metric that summarizes all links. We use an entropy measure ENT that captures the residual uncertainty. Since we expect the scaled estimation errors to be asymptotically Gaussian (similar to the case in [1]), we define the quality of the estimation across all links as

$$\mathrm{ENT} = \sum_{e \in E} \log\left(\mathbb{E}[\hat{\alpha}_e - \alpha_e]^2\right), \tag{2}$$

which is a shifted version of the entropy of independent Gaussian random variables with the given variances [17]. If the entire error covariance matrix R is available, then we can compute the metric as $ENT = \log\det R$, which captures also the correlations among the errors on different links. The metric ENT as defined above captures only the diagonal elements of R, i.e., the MSE for each link independently of the others.

Under mild regularity conditions (see for example Chapter 7 in [18]), the scaled (by sample size n) asymptotic covariance matrix of the optimal estimator is lower-bounded[1] by the Cramer-Rao bound I^{-1}. The Fisher information matrix I is a square matrix with element $I_{p,q}$ defined as

$$\begin{aligned} &I_{p,q}(\{\alpha_e\}) = \\ &\mathbb{E}\left[\frac{\partial}{\partial \alpha_p} \log p(\{X_r\}; \{\alpha_e\}) \frac{\partial}{\partial \alpha_q} \log p(\{X_r\}; \{\alpha_e\})\right], \end{aligned} \tag{3}$$

where α_p, α_q the loss probabilities of two links. In particular, under the regularity conditions, the MLE is asymptotically efficient, *i.e.,* it asymptotically (in sample size) achieves this lower bound.[2] Hence the asymptotic error covariance matrix

[1]For symmetric matrices $A \geq B$ means that $A - B$ is positive semi-definite.

[2]In [1], it has been shown that the asymptotic mean-squared error converges to this Fisher information bound for the multicast case. We believe that this should also be true for the multiple source case as well; so far, we have only numerically verified it so in our simulations.

of the MLE is approximately $\frac{1}{n}I^{-1}$. Therefore, we study the behavior of the Fisher information matrix for different topologies and network views as a basis of comparison; we can then lower bound the asymptotic mean-squared errors by examining the Fisher information matrix.

III. Optimal Estimation of a Single Link

In this section, we are interested in estimating the loss rate on a *single link*, typically in the middle of the network, by sending and observing probes from the edge. Let us revisit Fig. 1 and estimate the loss rate on link CD. Apart from illustrating our approach this basic 5-links topology is important in two ways: (i) it is the basic structure required for link CD to be identifiable as we discuss in the following subsection and (ii) any arbitrary topology can be reduced to this basic topology, if we view all links (except the link of interest CD) as directed paths from/to edge nodes A, B, E and F, with the same loss rates as their equivalent links. For example, a path from A to C, denoted as (A, C), can be reduced to link AC with loss rate α_{AC} the overall path loss rate.

A. Four Cases of Identifiability

A link $e \in E$ is said to be *identifiable* if it is possible to estimate the associated loss-rate α_e by sending probing packets from nodes in $\mathcal{S}$ to nodes in $\mathcal{R}$.

Fig. 1 depicts the four cases, i.e. choices of sources and receivers, that form the basic structures for the identifiability of the loss rate of link CD, when neither C or D are edge nodes. Notice that *Cases 1* and *3* use network coding with 2 sources and 2 receivers, *Case 2* uses a multicast tree with source A, and *Case 4* uses a reverse multicast tree with sink F. The necessary and sufficient conditions for identifiability, first observed in [15], are summarized in the following:

Theorem 1: Given $G = (V, E)$ and sets $\mathcal{S}$ and $\mathcal{R}$, a link CD is identifiable if and only if both conditions (1) and (2) hold:

Condition 1: At least one of the following holds:
(a) $C \in \mathcal{S}$.
(b) There exist two edge disjoint paths (A, C) and (B, C) that do not employ edge CD with $A, B \in \mathcal{S}$.
(c) There exists two edge disjoint paths (A, C) and (C, B) that do not employ CD with $A \in \mathcal{S}$, $B \in \mathcal{R}$.

Condition 2: At least one of the following holds:
(a) $D \in \mathcal{R}$.
(b) There exist two edge disjoint paths (D, E) and (D, F) that do not employ edge CD with $E, F \in \mathcal{R}$.
(c) There exists two edge disjoint paths (E, D) and (D, F) that do not employ CD with $E \in \mathcal{S}$, $F \in \mathcal{R}$.

Sketch of Proof. The proof of Theorem 1 is based on the observation that C and D need to be branching points or edge nodes, otherwise the link loss rate of edge CD will be indistinguishable from the loss rate of an ascendent or a descendant edge. ■

In [15] we considered the case where nodes A, B, E and F were constrained to belong in either $\mathcal{S}$ or in $\mathcal{R}$, and showed that use of network coding operations increases the number of identifiable links. Here, we assume that A, B, E and F are allowed to act as either sources or receivers. Our observation is that our choice of sources and receivers impacts the accuracy of our estimator; i.e., for a fixed number of probes, each topology leads to a different estimation accuracy. This implies that to achieve the same MSE, we may need to use a different number of probes for each topology.

B. Performance Comparison of the Four Cases

In Fig. 2 we assume that all 5 links have $\alpha = 0.3$ and we look at the convergence of the MLE vs. number of probes for *Case 1* (using network coding) and for *Case 2* (multicast probes with source A). Fig. 2(a) shows the estimated value (for one loss realization). Both estimators converge to the true value, with the network coding being only slightly faster in this scenario.

In Fig.2(b) we plot the mean-squared error of the MLE for *Case 1* (using network coding) and for *Case 2* (multicast) across number of probes. For comparison, we have also plotted the Cramer-Rao bound for link CD, which is consistent with the simulation results. For this scenario, *Case 1* does slightly better than *Case 2* but not by a significant amount. This motivated us to exhaustively compare all four cases in Fig. 1, for all combinations of loss rates on the 5 links.

Fig. 3 plots the Cramer-Rao bound for the four cases as a function of the link-loss probability at the middle link. The left plot assumes that α is the same for five links, while the right plot looks at the case where the edge links have fixed loss rate equal to 0.5. We observe that *Case 1* shows to achieve a lower MSE bound. Interestingly, the curves for *Case 2* (multicast) and *Case 4* (reverse multicast) coincide. The difference between the performance of different cases is more evident in the right plot (Fig. 3(b)).

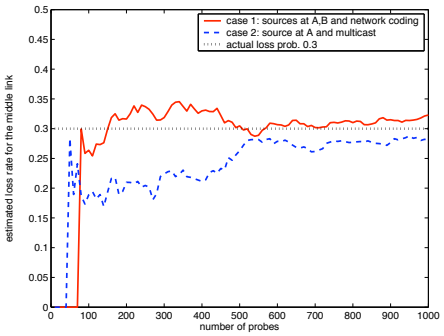
In Fig. 4, we systematically consider possible combinations of loss rates on the 5 links and we show which case estimates better the middle link. In the left figure, we assume that all edge links have the same loss rate and observe that for most combinations of $(\alpha_{middle}, \alpha_{edge})$, *Case 1* (shown in "+") performs better. In the right plot, we assume that the middle link is fixed at $\alpha_{CD} = 0.8$ and that $\alpha_{AC} = \alpha_{BC} = \alpha_s$,$\alpha_{DE} = \alpha_{DF} = \alpha_r$. Considering all combinations (α_s,α_r), each one of the four cases dominates for some scenarios. An interesting observation is, again, the symmetry between *Case 2* (multicast) and *Case 4* (reverse multicast). We prove in the next section that this symmetry holds over general trees.

IV. Estimation of All Links in a Tree

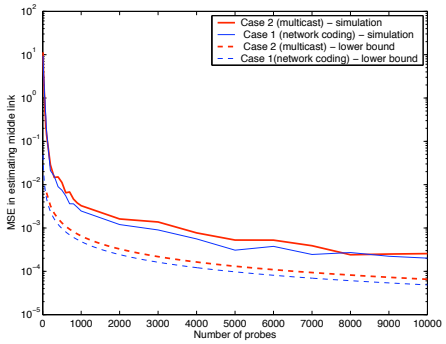
So far we studied the basic 5-links topology and focused on estimating a single link. In this section, we study larger trees and the estimation of all links. We show that several observations from the basic 5-links topology extend to general trees.

A. Dual Configurations and Reversibility

Consider a tree with $\mathcal{L}$ leave nodes, where $\mathcal{S}$ leaves act as sources and the remaining $\mathcal{R} = \mathcal{L} \setminus \mathcal{S}$ act as receivers of probes, and a given compatible orientation of the links. We

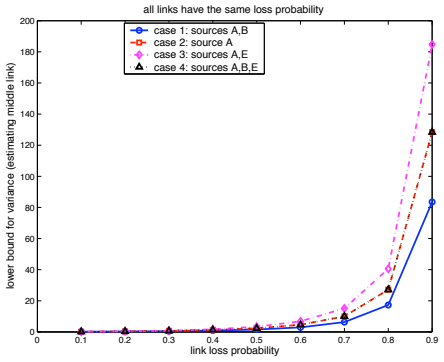


(a) Estimator vs. number of probes

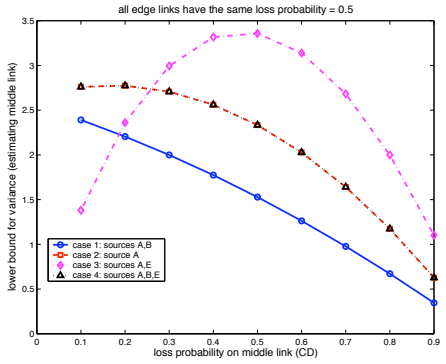


(b) Estimation variance vs. number of probes

Fig. 2. Convergence of the ML estimator for cases 1,2

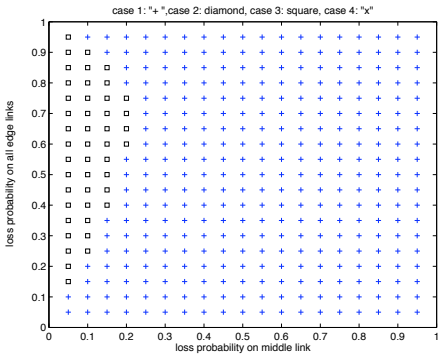


(a) All links have the same α

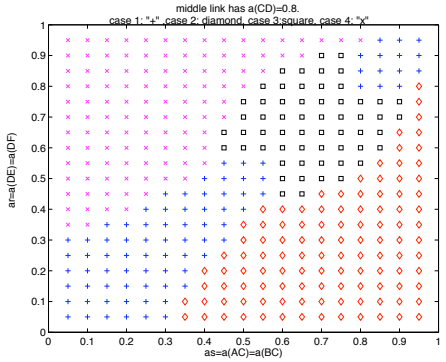


(b) All edge links have the same $\alpha_{egde} = 0.5$.

Fig. 3. Comparing the 4 cases in Fig. 1 in terms of the lower-bound of variance.



(a) All edge links have the same α_{edge}. Consider all possible combinations of (α_{edge},α_{middle}).



(b) $\alpha_{AC} = \alpha_{BC} = \alpha_s$,$\alpha_{DE} = \alpha_{DF} = \alpha_r$, $\alpha_{CD} = 0.8$. Consider all combinations (α_s,α_r)

Fig. 4. We indicate which Case (among the four) performs better (has the lowest Cramer-Rao bound), for a given combination of loss rates on all 5 links.

refer as "dual configuration" the configuration that results from reversing the orientation of all links in the network, and from having the $\mathcal{S}$ sources become receivers, while the $\mathcal{R}$ receivers act as sources. We show that, for the purposes of parameter estimation, the associated ML estimator function for a network and its dual is the same.

For example, a multicast tree is the dual configuration of an inverse multicast tree (Case 2 and 4 in Fig. 1). In Section III , we saw in Fig. 3(a) and Fig. 4(b) that the dual configurations resulted in the same mean square error bound. In fact, we observed that their associated ML estimator functions coincide. The following theorem generalizes this notion to arbitrary tree-like networks.

Theorem 2: The ML estimator for a tree configuration and its dual coincide.

Proof: Let $G = (V, E)$ be the original tree, with $|E| = n$, and G^d its dual. For every probe trial, there exist 2^n possible error events, depending on which links fail. Observing the outcomes at the receiver nodes corresponds to observing unions of such events, that occur with the corresponding probability. For a given configuration, the ML estimator depends on the observable outcomes at the receiver nodes. Therefore, it is sufficient to show that a network and its dual have effectively the same set of observable outcomes. In particular, we will show that for every observable outcome, that occurs with probability p in G, there exists exactly one observable outcome that occurs with the same probability in G^d and vice-versa, and thus

establish a bijection.

With every edge e of G, we can associate a set of sources $S(e) \subset V$ that flow through this edge, and a set of receivers $R(e) \subset V$ that observe the flow through e. Our main observation is that the pair $\{S(e),\ R(e)\}$ uniquely identifies e, i.e., no other edge has the same pair. In the dual network G^d, edge e is uniquely identified by the pair $\{R(e),\ S(e)\}$. If in G edge e fails while all other edges do not, the receivers $R(e)$ will not receive the contribution in the probe packets of the sources $S(e)$. If in G^d edge e fails while all other edges do not, the receivers $S(e)$ will not receive the contribution in the probe packets of the sources $R(e)$. Thus there is a one-to-one mapping between these events. Using this equivalence, an observable outcome consisting of a union of events can be mapped to an observable outcome at the reverse tree. ■

Note that this theorem establishes reversibility only for the maximum likelihood estimation. The performance of suboptimal algorithms might differ when applied to a configuration and its dual.

B. Measuring Directional Networks

We now examine the case where the loss rates are different in the two directions of a link. Our basic observation is that it is sufficient to send probes over only two configurations: the original and its dual.

Theorem 3: Consider a tree configuration with $|\mathcal{L}|$ leaves. We are interested in measuring the loss rates in both directions for all links of the tree. Using network coding saves a factor of $|\mathcal{L}|$ in bandwidth usage by probes, compared to the multicast tree approach.

Proof: Consider a tree configuration with $|\mathcal{L}|$ leaves. To measure the link loss rates in both directions for all edges of the tree, using the multicast approach, we need to use $\mathcal{L}$ multicast trees. Indeed, let $e = AC$ be the link adjacent to leaf $A \in L$, we can measure α_{AC} only if A is the root of the multicast tree. Using the network coding approach, for any choice of sources and receivers, we only need to perform two rounds of measurements: one on the network G and one on its dual G^d. ■

The previous theorem can also be interpreted as a tradeoff in directional measurement. We can either $|\mathcal{L}|$-fold increase the measurement bandwidth (using multicast probes), or allow intermediate nodes to do linear combinations (network coding). The former option keeps intermediate nodes simple at the expense of using extra bandwidth. The latter option sends exactly one probe per link for each measurement, but requires some operations from intermediate nodes.

A consequence of the reversibility established in the previous section is that, if we do not have a-priori knowledge of the link loss rates, the optimal choice of sources and receivers apply both to a configuration and its dual. That is, reversing the configuration, from Theorem 2, does not affect the estimation function, and thus will not lead to any performance degradation.

C. Number and Position of Sources

In Section III we observed that the number of sources and receivers affects the estimation accuracy for the basic topology in Fig. 1. This idea extends to larger topologies. In fact, not only the number of sources and receivers, but also their relative position on the tree (the "viewing point") affect the estimation accuracy.

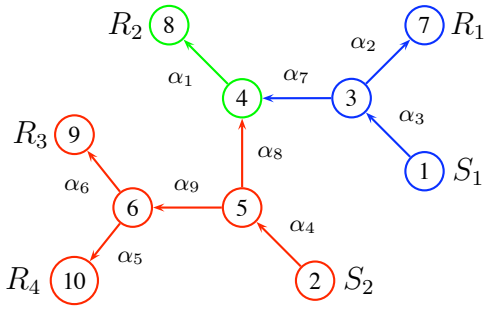


Fig. 5. A network topology with 9 links. The link orientations depicted correspond to nodes 1 and 2 acting as sources of probes.

To illustrate these concepts we use the tree shown in Fig. 5. We run simulations for three cases: (1) a multicast tree with source at node 1 (2) a multicast tree with source at node 2 (3) two sources at nodes 1 and 2 and a coding point at 4.[3] The same observations hold in this 9-link topology, as for the basic topology of Fig. 1. Simulations results are reported for this 9-link topology (and more extensive for a larger 45-link topology) in section V.D. Here we just report our observations and guidelines.

First, adding more than one source improves estimation; intuitively, this is because coding points partition the tree into smaller multicast components. Second, the number and placement of sources matter. Third, between two multicast trees with the same number of receivers, better performance is achieved by the tree that is more "balanced" and has the smallest height.

Elaborating on the first observation, note that in trees, each intermediate node is a vertex cut set. For the example of Fig. 5 node 4 decomposes the tree into three components. If node 4 could collect and produce probes, our estimation problem would be reduced in estimating the link-loss rates in three smaller multicast trees: the first tree consisting of source S_1 and receivers R_1 and node 4, the second tree with source S_2 and receiver nodes 4, R_3 and R_4 and the third tree with source node 4 and receiver R_2. Allowing node 4 to XOR incoming packets approximates this functionality: observing whether R_2 receives a packet that depends on x_1 or x_2, we can conclude on whether node 4 received a packet from S_1 or S_2 respectively.

The optimal selection of the number of sources and of the best points of view is quite involved. In general, it is a function of the network topology, the values of link-loss rates, and possibly the number of employed probes. From our simulation experiments, we found that the following guidelines apply to a tree with $\mathcal{L}$ leaf nodes.

[3]For the configuration in Fig. 5, the probes could also get combined in node 5. That is, although the choice of sources and receivers automatically determines the orientation of their adjacent links, there may still exist a choice of coding points and orientation for the intermediate links.

1) Select a fraction of sources to receivers that allows to partition the tree into roughly equal-size[4] subcomponents, where each subcomponent should have at least $2-3$ receivers.
2) Distribute the sources in roughly equal distances along the periphery of the network.

V. Algorithms and Simulation Results

The general ML estimator is computationally challenging for large tree topologies. In this section we propose three low complexity, yet efficient heuristics and evaluate them through simulation.

A. Subtree Estimation Algorithm

Based on the intuition discussed in Sec. IV, this algorithm partitions the tree into multicast subtrees separated by coding points. Each coding point virtually acts as a receiver for incoming flows and as a source for outgoing flows. As a result, each subtree will either have a coding point as its source, or will have at least one coding point as a receiver. In each subtree, we can then use the tomographic method proposed in [1].

Note that we can only observe packets received at the edge of the network but not at the coding points. However, we can still infer that information from the observations at the receivers downstream from the coding point. The fact that we infer the coding-points' observations from the leaves' observations is what makes this algorithm suboptimal.

Subtree Estimation Algorithm

– Consider a graph G. Given a choice of sources $\mathcal{S}$ and receivers $\mathcal{R}$, the coding points are determined and G is partitioned into $|\mathcal{T}| \leq 2\mathcal{S}-1$ subtrees.
– Each source sends one probe packet.Each receiver receives at most one probe packet.
– For each of the $|\mathcal{T}|$ subtrees:
If the multicast tree is rooted at a coding point:

- if any of the descendant receivers receives a probe, use this experiment as a measurement on the subtree,
- otherwise, w.p. p assume a measurement happened where no node in $\mathcal{R}$ received a probe packet, and w.p. $1-p$ ignore the experiment.

If the multicast tree is rooted at a source S_i:
Sequentially consider the descendant coding points that act as receiver. For coding point $\mathcal{C}$:

- if no descendant receivers $C(\mathcal{R})$ observed a probe, assume, w.p. p, that $\mathcal{C}$ did receive a packet, and w.p. $1-p$, that $\mathcal{C}$ did not receive a packet.
- otherwise
  - if at least one of $C(\mathcal{R})$ observed a linear combination of x_i, deduce that $\mathcal{C}$ received x_i.

---

[4]When links have similar loss rates, then "size" refers the number of nodes/links. In general "size" also should capture how lossy the links in the subcomponent are.

The probability p attempts to account for the fact that, if none of the receivers in $C(\mathcal{R})$ receives a packet, this might be attributed to two distinct events: either the coding point C itself did not receive a packet, or C did receive a packet, which then got subsequently lost in the descendent edges.

For example, in Fig. 5, consider the tree rooted at S_1, if R_2 receives x_1 or x_1+x_2 we deduce that x_1 was received at node 4. If R_2 received x_2, we deduce that x_1 was not received at node 4. If R_2 does not receive a probe packet, then, with probability $1-p$, we assume that 4 did not receive a probe packet. In general, the parameter p depend on the graph structure and possibly prior information we may have about the link-loss rates.

B. MINC-like Heuristic Algorithm

For every multicast node, we can use the MINC algorithm described in [1]. For every coding point, we can use reverse MINC, exploiting the reversibility property that we established in previous section. In order to infer which probes have been received in interior nodes from observations at the edge, we use the same procedures as in the subtree estimation. This heuristic is optimal for multicast and reverse multicast configurations, and for configurations that are concatenations of the two, but suboptimal for any other configuration.

C. Belief Propagation

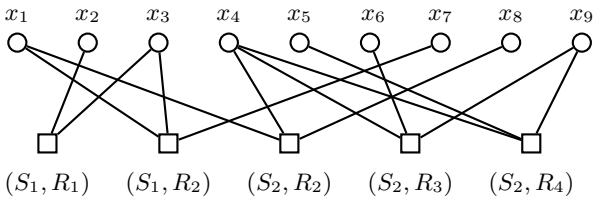


Fig. 6. Bipartite graph $\mathcal{B}$ corresponding to the 9-links tree in Fig. 5. $\mathcal{B}$ indicates which edges belong to which observable paths.

In [16], it has been observed that linear complexity belief propagation algorithms can naturally be used to estimate the link-loss rates in an inverse multicast tree. Our observation here is that the exact same approach can also be applied for tree configurations with multiple sources and network coding operations. Moreover, it can also be applied over arbitrary acyclic graph configurations, with appropriate network coding operations at intermediate links. We refer the reader to [16] for the message passing equations and rigorous description, and we only outline the basic idea in our context.

Consider a graph $G=(V,E)$, an experiment where one probe packet is sent from each source and let $x_e \in \{0,1\}$ denote whether edge e has failed or not. Construct a bipartite graph $\mathcal{B}$ that has $|E|$ variable nodes $\{x_e\}$, one for each edge of the graph, and $|W|$ check nodes, where each check node corresponds to a path w between a source and a receiver. Each path is connected to the edges it uses. For example, Fig. 6 shows the bipartite graph corresponding to the nine-links configuration of Fig. 5. In the real network, path

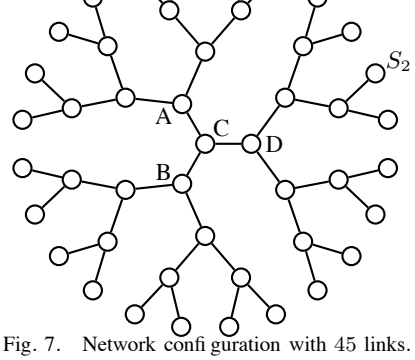


Fig. 7. Network configuration with 45 links.

(S_1, R_1) contains edges 2 and 3; therefore, in the bipartite graph $\mathcal{B}$, node (S_1, R_1) is connected to nodes x_2 and x_3.

Using the observations at the receivers, we allocate 1 or 0 to each check node, depending on whether the corresponding path operated or not. For example, in Fig. 5, if R_2 received the probe packet $x_1 + x_2$, we know that both paths (S_1, R_2) and (S_2, R_2) operate. A path operates if none of its edges fails, therefore the value for each check node equals the logical `AND` of its adjacent edges. Running the belief propagation algorithm on the bipartite graph $\mathcal{B}$, we can calculate with what probability each variable x_e takes value 0 or 1 or 0. Repeating the same procedure n times, we collect a set of n probability values for each edge e, which can be used to estimate α_e.

D. Simulation Results

In this section, we are interested (i) in evaluating the heuristics themselves and (ii) in showing that two sources (even with suboptimal estimation) outperform a single source (even with ML estimation).

Consider the 45-links topology shown in Fig. 7, where all links have the same loss rate α. We are interested in estimating α for all 45 links, and in comparing different methods in terms of their estimation variance.

First, we did simulations for $\alpha = 0.3$, a large number of probes, and repeated for many experiments. We then looked at the mean square error (MSE) in the estimation of each link. The results are shown in Fig. 8 for three algorithms:

1) a single multicast source S_1 and maximum likelihood estimation (top plot)
2) two sources S_1, S_2, network coding at the middle node C, and the MINC-like heuristic (middle plot)
3) the same two sources and coding point, with the subtree decomposition heuristic (bottom plot).

Notice that in the case of two sources, the 45-links topology is partitioned into 3 subtrees: one rooted at A (where probe x_1 "flows"), another rooted at D (where probe x_2 flows) and a third one rooted at B (where probe $x_1 + x_2$ flows).

There are several observations one can make from this graph. First, using two sources and network coding, even with suboptimal estimators, performs better than using a single multicast source and an ML estimator. Indeed the residual entropy (which is the metric that summarizes the MSE across all 45 links) is lower for two sources with

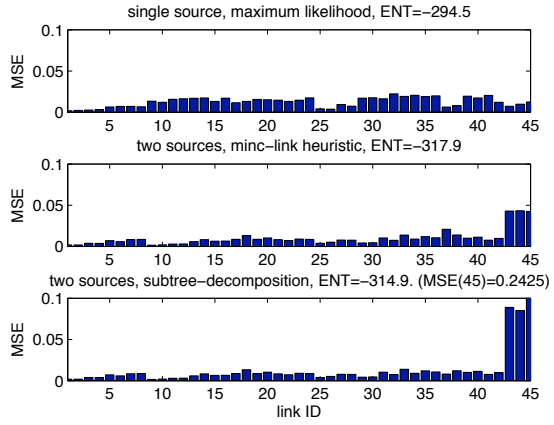


Fig. 8. MSE for each link in the 45-links topology.

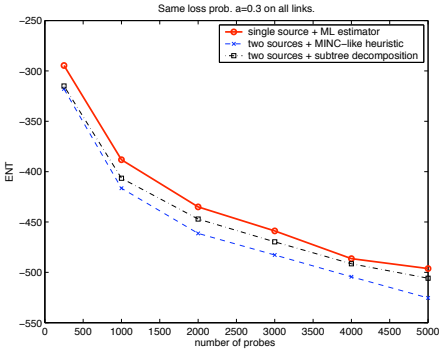
the MINC-like ($ENT = -317.9$) and for the subtree-decomposition ($ENT = -314.9$) heuristics, than it is for the single source MLE ($ENT = -294.5$). This illustrates the benefit of using multiple sources. Second, notice that the MSE for individual links is smaller in the lower two graphs than in the top graph, for all links except for links $43, 44, 45$, for which it is significantly higher. This is no coincidence: links $43, 44, 45$ are the middle ones (CA, CB, CD in Fig. 7. This is due to the fact that we cannot directly observe the packets received at the coding point C and we have to infer them from observations at the leaves of subtree rooted at B. The performance of the heuristics could further improve by using the following tweak: we could estimate what probes are received at C, using observations from leaves not only in the subtree rooted at B, but also from the subtrees rooted of A and D.

The above simulations were for a single value of $\alpha = 0.3$. We then exhaustively considered several values of α (same on all links) and n (the number of probes). The results are shown in Fig. 9. We can see that, even with suboptimal estimation, using two sources consistently outperforms a single multicast source, even with MLE estimation. This is apparent in Fig. 9 where the ENT metric for the single source (drawn in bold lines) is consistently above the other two algorithms.[5]

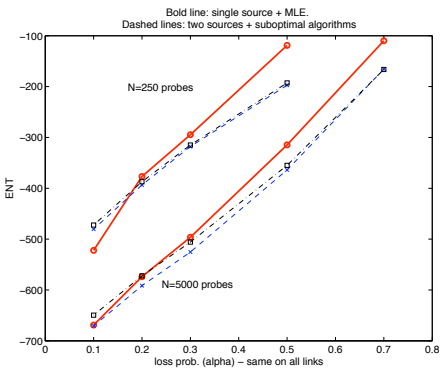
Finally, we present results for the belief propagation algorithm. In Figure **??**, we compare the MINC like heuristic and the message passing algorithms, over the 45 link network, with respect to the ENT measure. We set all links in this network to $\alpha = 0.5$ and varied the number of probes from 250 to 5000.

Both heuristics yield better performance (lower ENT values) as the number of sources increases from 1 to 5. However, the message passing heuristic is inferior to the MINC like estimator for the multicast tree (for which this is an ML estimator) as well as for the 2-source tree. However,

[5]Two notes about the ENT metric. First, the differences in the value of ENT are significant, although this is not visually obvious; recall that ENT is defined by taking the sum of the log of the MSE's. Second, ENT can be < 0, it is the differential entropy that matters.

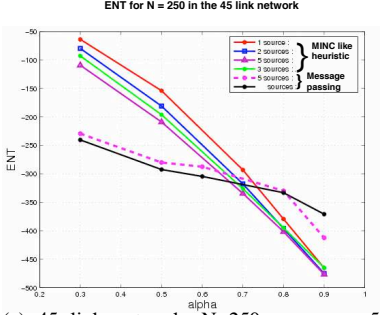


(a) ENT vs. number of probes

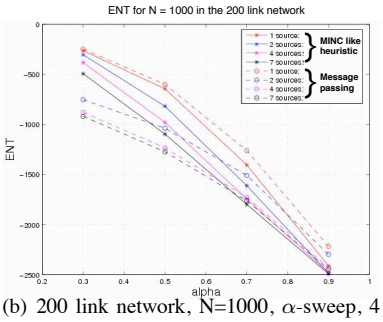


(b) ENT vs. loss probability

Fig. 9. Comparison of one source (with MLE to two sources (with a suboptimal estimation: MINC-like heuristic or subtree decomposition). The comparison is in terms of metric ENT



(a) 45 link network, N=250, α-sweep, 5 different coding schemes



(b) 200 link network, N=1000, α-sweep, 4 different coding schemes

Fig. 10. α-sweep for 45 and 200 link networks

it does a great deal better than the latter for the 3-source case and even more so in the 5-source case. This trend can be explained by looking at the number of cycles in the factor graph. A cycle is created in the factor graph of a network configuration when (1) two different paths have more than one link in common and (2) a set of m paths, say W_m, covers a set E_m of m links, with each of the paths in W_m containing at least two links in E_m. As the factor graph becomes more and more cyclic, the performance of the sum-product algorithm degrades. Given a specific network configuration, a raw estimate of the extent of cyclicity of the factor graph can be made by looking at the 'average number of links contained per path' and the 'average number of paths that a link is contained in'.

We applied the message passing algorithm to a 45 link- and 200 link-network and show that it performs better on those configurations that result in sparsely connected factor graphs. The results obtained for an α-sweep for the 45 link network with $N = 250$ is shown in Figure 10(a). The results obtained for the 200 link network for $N = 1000$ is shown in Figure 10(b).

We simulated for the average ENT:$\log_2(\frac{1}{E}\sum_{e\in E}[\hat{\alpha}_e - \alpha_e]^2)$ of the 45 link- and 200-link multicast trees, obtained from both the ML estimator and the message passing algorithms. The results are plotted in Figure 11. ENT_{av} of the 45 link tree is better (lower) than that of the 200 link tree for a given number of probes. Also, the message passing heuristic performs relatively better in the 45 link case, since the tree is more sparse than in the 200 link case.

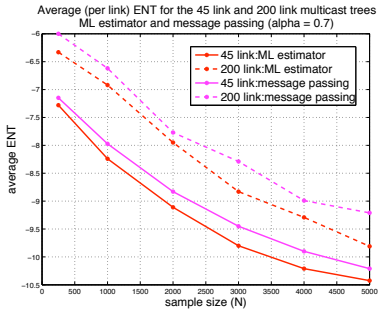


Fig. 11. Average ENT: multicast estimator in 45 link, 200 link topologies.

VI. Discussion

A. From Trees to General Topologies

Throughout the paper, we illustrated our ideas on trees. However, they can naturally be extended to more general topologies. The only difference is that intermediate nodes may need to perform slightly more elaborate operations, for example over finite fields.

As a concrete example, consider the configuration in Fig. 12, where two sources inject probes x_1 and x_2. Assume that intermediate nodes can perform operations over the field $\mathcal{F}_4$. Node A combines the probes to create packet x_1+x_2. Node B creates and sends $2x_1 + x_2$. Note that we can directly apply the heuristic algorithms discussed in Sec. V. E.g., we can decompose the network into parts through which the same "type" of probe packets flows. Each such part is a tree, and we can use a different estimator for each part, in the

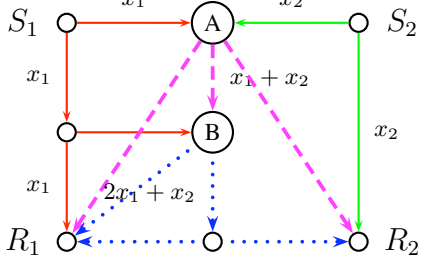


Fig. 12. A network configuration that is not a tree.

exact same way as in Sec. V-A. Similarly, we can use belief-propagation by decomposing the graph into observable paths and creating a bipartite graph as in Sec. V-C.

B. Comparison to other Monitoring Techniques

Network-coding enhanced active monitoring offers several advantages compared to previous techniques. Compared to classic tomography, we are able to identify a larger number of links and improve their estimation accuracy. The reasons behind this improvement are: (i) combining probes using network coding (e.g. $x_1 + x_2$ on link $C - D$) carries more information than x_1, x_2 sent separately) and (ii) using XOR to combine different probes results in exactly one probe per link, thus allowing us to get all the benefits from multiple sources without any increase in bandwidth. Compared to per-link measurements, our approach allows to keep internal nodes simple (assuming that they already implement network coding/XOR) and delegate all complexity to special nodes at the edge; furthermore, it can also be preferable to per-link measurements in dynamic environments where estimates need to be frequently reported, thus causing an extra overhead. Finally, appropriate choice of multiple sources and receivers can evenly distribute probe traffic across the network.

VII. Conclusion

In this paper, we studied link-loss monitoring using multiple sources of probes from the edge and network coding capabilities in the middle of the network. We showed that is possible to significantly improve link-loss estimation over previous purely tomographic techniques by combining three elements (i) multiple sources and receivers (ii) careful selection of the number and placement of sources at the edge of the network ("network points of view") and (iii) network-coding functionality at intermediate nodes (which eliminates the bandwidth overhead from multiple sources and also enriches the information carried by each probe). Overall, in networks where network coding is already deployed, these ideas can potentially improve monitoring; this potential can be fully exploited by using the right points of view and efficient estimators.

References

[1] R. Caceres, N. G. Duffield, J. Horowitz and D. Towsley, "Multicast-based inference of network-internal loss characteristics", IEEE Trans. in Inf. Theory, vol. 45, pp. 2462–2480, 1999.

[2] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurements", *ITC Seminar on IP traffic, Measurements and Modeling*, Monterey, CA, Sept. 2004.

[3] T.Bu, N.Duffield, F.Presti, and D.Towsley, "Network tomography on general topologies," *in Proc. ACM Sigmetrics, 2002*.

[4] M. Rabbat, R. Nowak and M. Coates, "Multiple source, multiple destination network tomography", *in Proc. of IEEE Infocom 2004*.

[5] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, July 2000.

[6] S-Y. R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Information Theory*, vol. 49, 2003.

[7] C. Fragouli. J. Widmer and J.Y. LeBoudec, "Network coding: an instant primer", CCR ACM SIGCOM 2006, vol. 36, is. 1.

[8] "The network coding webpage," *http://www.netcod.org*.

[9] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution", Infocom March 2005.

[10] "Avalanche: File Swarming with Network Coding", *http://research.microsoft.com/ pablo/avalanche.aspx*

[11] S.Katti, D.Katabi, W.Hu, H. Rahul, M.Medard, "The importance of being opportunistic; practical network coding for wireless environments", *in Proc. of Allerton Conference*, Sept. 2005.

[12] D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient overlay networks," *in Proc. of ACM SOSP*, Oct. 2001.

[13] Y. Chen, D. Bindel, H.Song and R.Katz, "An algebraic approach to practical and scalable overlay network monitoring," *in Proc. ACM SIGCOMM 2004*.

[14] Y. Zhu, B. Li, J. Guo, "Multicast with network coding in application-layer overlay networks," *in IEEE JSAC, Special Issue on Service Overlay Networks*, 4^{th} Quarter, 2003.

[15] C. Fragouli, A. Markopoulou, "A network coding approach to overlay network monitoring", *in Proc.of 43rd Allerton*, Sept. 2005.

[16] Y. Mao, F. R. Kschischang, B. Li and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks", *in IEEE JSAC, Special Issue on Self-Organizing Distributed Collaborative Sensor Networks*, vol. 23, pp. 820-829, April 2005.

[17] T.Cover,J.Thomas,"Elements of Information Theory",Wiley 1991.

[18] E L. Lehmann, *Elements of large-sample theory*, *Springer*, 1999.