Network tomography via network coding

G. Sharma
Department of Electrical Engg.
Indian Institute of Tech. Bombay
Mumbai, India, 400 076
gauravsharma@iitb.ac.in

S. Jaggi
Department of Information Engineering
Chinese University of Hong Kong
Shatin, N.T., Hong kong
jaggi@ie.cuhk.edu.hk

B. K. Dey
Department of Electrical Engg.
Indian Institute of Tech. Bombay
Mumbai, India, 400 076
bikash@ee.iitb.ac.in

***Abstract*— In this work we show how existing network coding algorithms can be used to perform network tomography, i.e., estimate network topology. We first examine a simple variant of the popular distributed random network codes proposed by Ho et al. [14], and show how it can enable each network node to passively estimate the network topology upstream of it at no cost to throughput. The delays introduced by each upstream node and link can also be similarly estimated. We then consider the scenario wherein an adversary hidden in the network wishes to disrupt the estimation of network topology. We show how network error-correcting codes can be used to reliably perform network tomography if the network has sufficient connectivity, and demonstrate that network tomography is impossible otherwise.**

Index terms: **Network coding, network tomography, topological identification, Byzantine adversaries.**

I. I

Network tomography attempts to characterizes the internal properties of large networks via measurements at the boundaries. Clearly, internal nodes have a large influence on network performance; routers inside the network determine the paths that information flows in, packets losses inside networks affect transmission rates, and cumulative delays incurred on nodes and links determine the end-to-end delay of transmissions. However, since currently implemented networks often do not allow clients to access the internal infrastructure, all such characteristics must be inferred by analyzing end-to-end transmissions.

We consider a model wherein each network node v has a unique identification number $i(v)$. (For instance, the IP address of a node, its GPS coordinates, or a factory stamp could function as such identification numbers.) Prior to communication, each node knows only its own identification numbers, and those of the nodes immediately upstream and downstream of it. The goal of this work is for a node to determine the identification numbers of *all* the nodes upstream of it and the set of links between them. To aid in this process, nodes are allowed to pass on arbitrary messages from upstream nodes to downstream nodes. We also consider an adversarial setup wherein malicious nodes hidden in the network attempt to disrupt this process (for instance by lying about their own node number, or by corrupting the messages passed on by upstream nodes). The focus of this paper is to show that current results and algorithms in the network coding literature generate network tomography schemes with interesting properties.

In the first part of this work, we note that such network tomography can be done at no extra cost when implementing a classical distributed network code. The distributed network codes suggested by Ho et al. [14] are implemented by each node performing random linear operations over a finite field on incoming packets to generate outgoing packets. Each packet is appended with a short header that encodes the linear transform thus far. We show that a result of Ho et al [15] implies that for network codes over large enough fields, with high probability the headers themselves can act as signatures that are distinct for distinct networks. The receiver(s) (and indeed every node) can then passively infer the upstream network topology via the headers on incoming packets, without the tomography disrupting the information transmission. We note that this scheme can even be directly extended to the problem of determining dynamic network properties, such as packet delays incurred at each node and link.

We then consider the problem of actively estimating network topology in the presence of a malicious Byzantine adversary who wishes to disrupt the estimation process. For instance, there might be a malicious router who wishes to disrupt internet communication by falsifying routing tables in the other routers. As a first step, we examine whether a node v can reliably estimate whether a particular directed edge (u, u') upstream of it exists or not. That is, if the nodes with indices u and

u' are both known to be in the network, whether they are connected via an edge directed from u to u'. The node v can estimate the topology of the entire network upstream of it by repeating this process at most $|\mathcal{V}|^2$ times (where $\mathcal{V}$ is the set of nodes in the network). Let $\mathcal{Z}$ be a set of links of size Z that the adversary controls (controlling a node is equivalent to controlling all links outgoing from it). We show that if Z is at least half the capacity from $\{u, u'\}$ to v, then regardless of the scheme used the adversary can always force a probability of error of at least $1/2$. Conversely, if the capacity from $\{u, u'\}$ to v is strictly greater than $2Z$, network error-correcting codes recently discussed in the literature can be used to reliably estimate the existence of (u, u') in a computationally efficient manner.

II. B

A. Network Coding

Linear network coding is the paradigm of encouraging each node to perform linear operations on received packets to generate transmissions. The seminal work of [4] and [5] showed that this results in information-theoretically optimal codes. But arguably at least as important a contribution of the field of network coding is the paradigm shift it caused in diverse fields such as network algorithm design ([11], [14]), coding theory ([13], [7]) and efficient wireless communications ([3], [6]), among others. This work demonstrates the possible benefits of network coding for the field of network tomography.

B. Network Tomography

The term network tomography arises from the analogy to a process (commonly used in medical and industrial imaging) used to construct a model of the cross-section of an object. This imaging technique passes a multitude of rays of rays in different directions through the object to be measured, and reconstructs its two-dimensional cross-section by solving the corresponding estimation problem. The idea behind network tomography is quite similar, in that end-to-end communication parameters (such as round-trip times, packet losses, headers) are used to estimate the internal characteristics of the network.

One of the first works to consider *network tomography* was by Vardi [19]. A survey of several results in the networking literature can be found in Castro et al [20]. Commonly used schemes typically require active network probing, with packets specially injected for the purpose of tomography. The schemes usually require careful choice of injected packets, and heuristic probabilistic or statistical signal-processing approach to estimating the topology.

C. Network Coding Tomography

Network codes, by design, require nodes inside the network to perform computations on their received packets to generate transmitted packets. In the case of a popular class of distributed linear network codes [14], each packet is also appended with a small header that internal nodes use to transmit information about their linear coding operations. The fact that internal nodes are allowed to perform (linear) computations, and that packet headers contain signatures of these linear operations, act as an additional resource for passive network tomography. This idea has been leveraged by some authors. The work in [1] considers peer-to-peer systems, and uses subspace ideas similar to those proposed in [7] to identify local bottlenecks. In another work [2] the authors assume global knowledge of the network code, and based on this strong assumption passively infer the topology of the network.

The first results (Theorem 4 and 5) in our paper most closely resembles the work of [9]. The main result in [9] shows that a single random network code can be simultaneously robust against a set of failure patterns (i.e., when some links go down), and also distinguish between each of those failure patterns. However, they do not consider the problem of network identification, and also cannot distinguish between certain classes of networks. Our work can distinguish between any pair of networks, and are also generalized to estimate the *delay topology* of the network; i.e., the amount of delays incurred by transmissions at any node or link.

Our second set of results (in Theorem 6) build directly on the recent advances in network error-correcting codes ([13], [7]). Such codes can be used to reliably communicate over a network containing a Byzantine adversary as long as certain network connectivity conditions are met. In particular, reliable communication is possible if and only if the mincut from the source to the sink is larger than twice the mincut from the adversary to the sink. The pair of nodes $\{u, u'\}$ use these network error-correcting codes to signal existence or non-existence of the edge (u, u') to v.

III. P N T

In the *passive* tomography case nodes estimate the topology of the network upstream of them using the

headers of network codes that are already being used for information transmission.

We use a general model that encompasses both wired and wireless networks. To simplify notation, we consider only the problems where a single source wishes to communicate with a single destination over an acyclic network. However, using standard reductions our techniques extend to general multicast communication networks (multiple sources, multiple sinks, delays at nodes and edges, delays at nodes and edges, cycles) over which random linear network codes are employed. Our code design is almost identical to that in [14]. The crucial difference is in the decoder – our codes allow the decoder not only to retrieve the source's information, but also to retrieve the network topology with no penalty to the communication rate. For completeness, we also outline the distributed network code design of [14].

The network is modeled as a hypergraph with edge set $\mathcal{E}$ and vertex set $\mathcal{V}$ [17]. There is a source, Alice, who communicates to a destination, Bob, over a wired or wireless network. Each packet transmission corresponds to a hyperedge (henceforth simply called an edge) directed from the transmitting node to the set of observer nodes. The hypergraph model captures both wired and wireless networks. For wired networks, the edge is a simple point-to-point link. For wireless networks, each such edge is determined by instantaneous channel realizations (packets may be lost due to fading or collisions) and connects the transmitter to all nodes that hear the transmission. The hypergraph is unknown to Alice and Bob prior to transmission. Each node v is assumed to have a unique identification number $i(v)$ We assume that for each v, $i(v) \in \{1, \ldots, M\}$, where $M \geq |\mathcal{V}|$ is an upper bound on the network size known in advance by each network node.

The *network capacity*, denoted by C, is defined as usual the time-average of the maximum number of packets that can be delivered from Alice to Bob over the network, i.e., the max flow of the hypergraph representing the network. It can be also expressed as *the min-cut from the source to the destination*. (For the corresponding multicast case, C is defined as the minimum of the min-cuts over all destinations.) It is assumed that prior to code-design, the value of C, or at least a good upper bound h to it, is known to the source.

Source: Alice generates incompressible data that she wishes to deliver to Bob over the network. To do so, Alice encodes her data as dictated by the encoding algorithm (described in subsequent sections). She divides the encoded data into batches of b packets.

A packet contains a sequence of n symbols from the finite field $\mathbb{F}_q$. All arithmetic operations henceforth are done over symbols from $\mathbb{F}_q$. The choice of the finite field $\mathbb{F}_q$ is a design parameter for each network code. Out of the n symbols in Alice's packet, b symbols are redundancy added by the source.

Alice organizes the data in each batch into a matrix $[\mathbf{X}]$ as shown below.

$$[\mathbf{X}] = \begin{bmatrix} x(1,1) & x(1,2) & \cdots & x(1,n) \\ x(2,1) & x(2,2) & \cdots & x(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ x(b,1) & x(b,2) & \cdots & x(b,n) \end{bmatrix}$$

The i^{th} row in the matrix $[\mathbf{X}]$, denoted $[\mathbf{X(i)}]$, is just the i^{th} packet in the batch. As in standard distributed network codes [14], the redundancy in each packet is devoted to a length-b unit vector, hence the matrix $[\mathbf{X}]$ is of the form $[\mathbf{X_1}\ \mathbf{I}]$, where $[\mathbf{I}]$ denotes a $b \times b$ identity matrix, and $\mathbf{X_1}$ denotes the message part of the packets.

Encoders: As in [14], Alice takes linear combinations of the rows of $[\mathbf{X}]$ to generate her transmitted packets. In particular, for each edge e leaving the source s the packet $Y(e)$ traversing e is generated via the linear transform $\vec{\beta}_{\mathbf{e}}[\mathbf{X}]$. Here $\vec{\beta}_{\mathbf{e}}$ (called the *local coding vector for e*) is a vector over $\mathbb{F}_q$. Its components are the *local coding variables* $\beta_{i,e}$, for all i in $\{1, \ldots, b\}$.

As the packets traverse the network, the internal nodes also apply linear transforms to received packets to generate the packets they broadcast. For each edge e leaving a node v, the packet $Y(e)$ traversing e is generated via the linear transform $\vec{\beta}(e)[\mathbf{Y(v)}]$. Here $\vec{\beta}(e)$ (called the *local coding vector for e*) is a vector over $\mathbb{F}_q$. Its components are the *local coding variables* $\beta_{e',e}$, for all edges e' that are incoming to v.

The network code, i.e., the choice of local coding vectors for each edge, is described in Section III-A.

Decoder: Analogously to how Alice generates $[\mathbf{X}]$, the destination Bob organizes the received packets into a matrix $[\mathbf{Y}]$. The i^{th} received packet corresponds to the i^{th} row of $[\mathbf{Y}]$. Note that the number of received packets, and therefore the number of rows of $[\mathbf{Y}]$, is a variable dependent on the network topology. The rank of $[\mathbf{Y}]$, however, must be at least b for Bob to decode successfully. Bob attempts to reconstruct Alice's information, $[\mathbf{X}]$, using the matrix of received packets $[\mathbf{Y}]$.

Since each packet transmitted by an internal node is a linear combination of its incoming packets, the effect

of the network at the destination can be summarized as follows.

$$[\mathbf{Y}] = [\mathbf{T}][\mathbf{X}] = [\mathbf{T}\mathbf{X_1}\ \mathbf{T}], \quad (1)$$

The matrix [**T**] refers to the linear transform from Alice to Bob. The identity matrix in the last b columns of [**X**] incurs the same transform as the rest of the batch. Thus, Bob receives a description of the network transform in [**T**].

Theorem 1 of [10], whose statement is reproduced below in Theorem 1, shows that the probability that [**T**] does not have full column rank is asymptotically negligible. With high probability, therefore, Bob can decode [**X**] as $[\mathbf{T}]^{-1}[\mathbf{Y}]$.

We can now state the well-known prior results of [14], [10].

Theorem 1: (Theorem 1, [14], [10]) Choosing local coding variables uniformly at random from $\mathbb{F}_q$ results in a network code that allows Alice to communicate with Bob at a rate asymptotically equaling h and with a probability of error less than $(1 - 1/q)^{|\mathcal{E}|}$.

In addition, Theorem 3 in [8] is useful, since it gives an explicit characterization of [**T**] in terms of the local coding variables. Let F be the *line graph matrix*, i.e., the $|\mathcal{E}| \times |\mathcal{E}|$ matrix whose $(i, j)^{th}$ entry equals $\beta(e_i, e_j)$. Let A be the *input matrix*, i.e., the $b \times |\mathcal{E}|$ matrix whose $(i, j)^{th}$ entry equals β_{i,e_j}. Let μ be the number of edges incoming to the sink, and B be the *output matrix*, i.e., the $|\mathcal{E}| \times \mu$ matrix whose $(i, j)^{th}$ entry equals $\beta_{e_i,j}$, where e_i is the ith incoming edge to the sink.

Theorem 2: (Theorem 3, [8]) The transfer matrix [**T**] from Alice to Bob equals $B(I - F)^{-1}A$.

A. Topology Identification

We are now in a position to state our code design and the corresponding results for passive tomography.

First, each vertex v_j pares the network of excess edges, so that there are no more than h edges incoming to it from any other vertex v_i, nor more than h edges outgoing from it to any other vertex v_k. Since it is assumed that $h \geq C$, this does not change the achievable rate of the network code. In the resultant pared network, each node may still need up to hM local coding vectors, each of length at most hM. The required vectors are chosen by selecting $(hM)^2$ elements uniformly i.i.d. from $\mathbb{F}_q$. Unlike [14], the $(hM)^2$ local coding variables for each node v in the network are assumed to be generated and known in advance by all nodes in the network, even though the network topology itself is not known in advance to anyone. The underlying assumption is that there is a common source of randomness available to each network node, and the network code choices are a deterministic function of this common randomness. For example, consider the the phone book of a random city (typically available to all residents) as analogous to the common randomness, and the phone number of each resident being analogous to the local coding vector.

For any edge e let $\vec{\mathbf{X}}_\mathbf{e}$ denote the information passing through edge e (represented as a length-n vector). Then the message through an edge e' outgoing from the head of e is encoded as

$$\vec{\mathbf{X}}_{\mathbf{e}'} = \sum_{e:head(e)=e'} \beta_{e,e'} \vec{\mathbf{X}}_\mathbf{e}. \quad (2)$$

Let β equal the set of all the local coding variables of all the nodes in a network. Two networks are considered *detectably different* if the subgraphs induced by the nodes taking part in at least one path from the source to the sink are different. As an example of the implications of our definition, even if one network can be transformed into another simply by renumbering some nodes' identification numbers, they are considered to be detectably different. Prior work on network coding tomography does not distinguish between some such classes of networks. Lemma 3 shows that the overall transfer matrix T with entries viewed as polynomials of the components of β are different for detectably different networks.

Lemma 3: The transfer matrices T for detectably different networks are distinct.

Proof: For two given networks, let $N_{(v,v')}$ and $N'_{(v,v')}$ denote the number of edges from node v to node v' in the two networks. Since the networks are different, there exist nodes v and v' such that $N_{(v,v')} > N'_{(v,v')}$, and in the first network there is a path from the source to v to v' to the sink. By Theorem 1 of [16], there is a component in the transfer matrix T for the first network where the degree of some $\beta_{i,j}$ corresponding to the extra edge is nonzero, whereas the degree of the same variable in every component of the transfer matrix of the second network is zero. ■

Lemma 3 compares the transfer matrices corresponding to different networks in terms of the variables in the local encoding vectors. However, the linear transforms observed by each node depends on the values of the local coding variables chosen by each node rather than the variables themselves. The following theorem shows that if the values of each of these variables is chosen by generating them i.i.d. according to a uniform distribution from a sufficiently large field $\mathbb{F}_q$, then with high proba-

bility the transfer matrices of *all* the different networks will also be distinct. The proof technique is very similar to that employed in [9], where it is shown that different failure patterns of networks can be identified using the packet headers.

Theorem 4: If each local coding variable is generated i.i.d. with a uniform distribution over $\mathbb{F}_q$, then the probability that all the different unicast networks with at most $|\mathcal{V}|$ nodes and at most $|\mathcal{E}|$ edges will have distinct transfer matrices is at least $1 - |\mathcal{V}|^{4|\mathcal{E}|}\left(1 - (1 - \frac{1}{q})^{|\mathcal{V}|}\right)$.

Proof: There are $|Vertices|$ possible starting points and $|Vertices|$ possible ending points for each edge. Therefore the number of networks with at most $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges is bounded from above by $|\mathcal{V}|^{2|\mathcal{E}|}$. Each path in any of the networks have at most $|\mathcal{E}|$ edges. Since the networks are acyclic, each variable has degree at most one in every component of the transfer matrix, and every component has the total degree at most $|\mathcal{V}|$. Consider two different unicast networks $N_{(v,v')}$ and $N'_{(v,v')}$, and the corresponding transfer matrices $T(N_{(v,v')})$ and $T(N'_{(v,v')})$. By Lemma 3 at least one of the polynomials corresponding to an entry of the difference of the transfer matrices $T(N_{(v,v')}) - T(N'_{(v,v')})$ is non-zero. By Lemma 4 in [15], which is a result of recursive application of the Schwartz-Zippel Lemma [18], the probability that this polynomial equals zero for a particular choice of the variables β is at most $1 - (1 - \frac{1}{q})^{|\mathcal{V}|}$. Using the union bound over all possible networks gives us the required result. ∎

The above proof can be directly extended for the case where each node wishes to estimate the entire network upstream of it. For wireless networks (modeled via hypergraphs) we consider hyperedges instead of edges. The code construction and identification is exactly the same, except that each node codebook will be bigger since there are $2^{|\mathcal{V}|-1}$ possible edges outgoing from any node compared to $|\mathcal{V}| - 1$ possible edges.

B. Delay topology identification

If the edges in the network have different delays, then the delay topology can also be estimated along with the network topology using the same technique as in Theorem 4. Suppose the delay at each edge is known at the head node (the receiver) of that edge (up to the nearest time unit). For every pair (e, e') of incoming and outgoing edges, a node uses a different local variable β_{e,e',d_e} for different delay d_e of e. Thus the message for the edge e' is encoded as

$$\vec{\mathbf{X}}_{e'} = \sum_{e:head(e)=m} \beta_{e,e',d_e}\vec{\mathbf{X}}_e. \tag{3}$$

Typically, an upper bound D on the edge delay will be known a priori by Bob. The following result can then be proved in the same way as Theorem 4. In this theorem, two networks with same nodes and edges are also considered different if at least one edge has different delays in the two networks.

Theorem 5: If each local coding variable β_{e,e',d_e} is generated i.i.d. with a uniform distribution over $\mathbb{F}_q$, then the probability that all the different unicast networks with at most $|\mathcal{V}|$ nodes, at most $|\mathcal{E}|$ edges and maximum edge delay D will have distinct transfer matrices is at least $1 - |\mathcal{V}|^{4|\mathcal{E}|}D^{2|\mathcal{E}|}\left(1 - (1 - \frac{1}{q})^{|\mathcal{V}|}\right)$.

Note: If there are delays at both links and nodes, then every node u may also scale all its coding variables by another variable α_{d_u} depending on its delay d_u. Again, using the technique of Theorem 4 it is then possible, with high probability, to identify the network topology and the delay topology of the network from the received transfer matrix.

IV. A N T

In the *active* tomography case some or all the internal nodes are allowed to perform any general computation of messages on incoming links to generate messages on outgoing links. Such schemes may be harder to implement, and may be more disruptive to network communication. However, since this paradigm is more general than that of passive network tomography, more powerful schemes can be designed.

Under this paradigm, we consider the harder problem of discovering the network topology upstream of a node, given that the network may contain one or several Byzantine adversaries intent on disrupting the tomography process. To this end, these adversaries may corrupt all information coming from or passing through them.

The adversary controls an arbitrary set edges of size Z. The set of links he controls is denoted $\mathcal{Z}$. He can transmit arbitrary messages on the edges in $\mathcal{Z}$. This model also encompasses adversarial control of nodes, since controlling a set of nodes is equivalent to controlling all edges outgoing from them. The adversary knows in advance the network tomography protocol that network nodes use. In addition, we can even allow him to know the entire network topology, and the messages transmitted on each honest link since (as we show below) this does not change the result.

We reduce the tomography problem to that of detecting whether a single edge is in the network or not. More precisely,

Problem 1**. Edge Detection in the presence of an adversary:** A node v wishes to reliably estimate whether two nodes u and u', known by v to be in the network upstream of it, have an edge (u, u') between them. All communication is over a directed network containing a Byzantine adversary containing a set $\mathcal{Z}$ of links.

If Problem 1 can be reliably solved, repeating the process at most M^2 times allows the node to estimate the existence or non-existence of each link.

As in the passive tomography case, for simplicity we consider directed acyclic wired networks (though the results can be directly to wired networks with cycles).

We note that Problem 1 defined above can further be reduced to the following network communication problem.

Problem 2**. Network communication in the presence of an adversary:** The set of nodes $\{u, u'\}$ (if they are in the network) wish to communicate a single bit reliably to the node v over a network containing a Byzantine adversary controlling the set $\mathcal{Z}$ of links.

The reason is that the only part of the network that knows whether or not the directed edge (u, u') is in the network are the two nodes u and u'. The single bit they wish to communicate corresponds to the existence or non-existence of this link.

The reduction of Problem 1 to Problem 2 allows us to use the set of recent results concerning network communication in the presence of Byzantine adversaries.

In particular, Theorem 2 [13] is relevant. Let C be the min-cut from the source comprising the two nodes u and u' to the node v.

Theorem 6: The nodes u and u' can communicate reliably to a sink over a network with an adversary who controls the links $\mathcal{Z}$ if and only if $C > 2Z$.

Sketch of Proof: If Z is at least half of C, then the adversary can choose his edges $\mathcal{Z}$ to lie in a min-cut from $\{u, u'\}$ to v. He can then mimic the tomography protocol that the nodes $\{u, u'\}$ are using, but flip the bit of information that they wish to communicate.

On the other hand, if Z is less than half of C, then $\{u, u'\}$ can use a network error-correcting code (for example [13], [7]) to communicate reliably with v. These codes are polynomial-time and distributed. In fact, as outlined in [13], they can also be overlaid on the distributed network codes of Ho et al – the only difference lies in the fact that the nodes $\{u, u'\}$ need to add some redundancy to their transmissions, and the node v needs to solve a set of linear equations. Thus the only active part of this tomography scheme is for the nodes u, u' and v' – all other nodes can perform the distributed network coding scheme of Ho et al, oblivious to the presence of an adversary.

.....

.....

V. C

In Section III we outline passive network tomography schemes that can estimate both the static and dynamic topology of the network. These schemes can be implemented via existing network codes, at no cost to the throughput. The trade-off is that the tomography schemes we propose have high complexity of implementation at the decoder – we are currently investigating whether this can be lowered. In Section IV we we provide matching necessary and sufficient conditions the problem of estimating the existence of a single edge despite the presence of an adversary who wishes to obfuscate this process. It is possible that the necessary conditions on estimating the existence of multiple edges simultaneously are less stringent, since if the adversary attempts to disrupt the tomography process for the estimation of one edge, he may be unable to do so for another. This is an area of current investigation. We are also investigating what can be achieved by purely passive network coding tomography in the presence of an adversary.

A

G. Sharma was supported by a Direct Grant and the MS-CU-JL grant during his internship at the Chinese University of Hong Kong. We wish to thank Raymond Yeung for his suggestion that our schemes could also estimate the delay topology of the network.

R

[1] M. Jafarisiavoshani, C. Fragouli, S. Diggavi, and C. Gkantsidis, “Bottleneck discovery and overlay management in network coded peer-to-peer systems,” in *SIGCOMM INM*, 2007.

[2] M. Jafarisiavoshani, C. Fragouli, and S. Diggavi. “Subspace properties of randomized network coding,” presented at the *Information Theory Workshop*, 2007.

[3] P. A. Chou and Y. Wu, “Network coding for the Internet and wireless networks,”*IEEE Signal Processing Magazine*, vol. 24, issue 5, pp. 7785, 2007.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216,July 2000.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear Network Coding,” *IEEE transactions on Information theory*, vol. 49, no. 2, pp. 371, February 2003.

[6] S. Katti, S. Gollakota, and D. Katabi, “Embracing Wireless Interference: Analog Network Coding,” *ACM SIGCOMM 2007*.

[7] R. Koetter and F. R. Kschischang, “Coding for Errors and Erasures in Random Network Coding,” submitted to the *IEEE Transactions on Information Theory*, March, 2007.

[8] R. Koetter, and M. Médard, "An Algebraic Approach to Network Coding," *IEEE Transactions on Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

[9] T. Ho, B. Leong, Y. Chang, Y. Wen, and R. Koetter, "Network Monitoring in Multicast Networks Using Network Coding," ISIT 2005, pp. 1977-1981.

[10] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," Invited Paper, 41st Allerton Annual Conference on Communication, Control, and Computing, 2003.

[11] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973-1982, June 2005.

[12] S. Jaggi, P. A. Chou, and K. Jain, "Low Complexity Algebraic Network Multicast Codes," ISIT 2003.

[13] S. Jaggi, M. Langberg, M, S. Katti, T. Ho, D. Katabi, M. Medard, "Resilient Network Coding in the Presence of Byzantine Adversaries," INFOCOM 2007.

[14] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," ISIT 2003.

[15] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413-4430, October 2003.

[16] T. Ho, D. R. Karger, M. Médard, and R. Koetter, "Network Coding from a Network Flow Perspective," ISIT 2003.

[17] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-Cost Multicast over Coded Packet Networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2608–2623, June 2006.

[18] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[19] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, March 1996.

[20] R. Castro, M. J. Coates, G. Liang, R. Nowak, and B. Yu, "Internet tomography: Recent developments," *Statistical Science*, vol. 19, no. 3, pp. 499–517, 2004.