A Network Coding Approach to Overlay Network Monitoring

Christina Fragouli
École Polytechnique Fédérale de Lausanne
christina.fragouli@epfl.ch

Athina Markopoulou
Stanford University
amarko@stanfordalumni.org

September 28, 2005

Abstract

Monitoring and diagnosis of network conditions is a central problem in networking. As such, it has received a lot of attention in the Internet community in general and in the context of overlay networks in particular. Independently, recent advances in network coding have shown that it is possible to increase network capacity and better share the available resources by allowing intermediate nodes to perform processing operations, in addition to just forwarding packets. In this work, we propose the use of network coding techniques to improve several aspects of network monitoring in overlay networks. As a specific application, we use our approach for the well-known problem of network tomography, and in particular for inferring link loss rates from end-to-end measurements. We demonstrate that our approach can decrease the bandwidth used by probes, improve the accuracy of estimation, and decrease the complexity of selecting paths or trees to send probes.

1 Introduction

Distributed Internet applications often use overlay networks, that enable them to detect and recover from failures or degraded performance of the underlying Internet infrastructure. To achieve this high-level goal, it is necessary for the nodes in the overlay to monitor Internet paths, assess and predict their behavior, and eventually make efficient use of them. Clearly, accurate monitoring at minimum overhead and complexity is of crucial importance for the operation of all networks, and particularly for overlay networks [1].

In the past decade, several approaches have been proposed for inferring network characteristics of interest (such as topology, packet loss rate, delay, and failures) using end-to-end measurements [2, 3, 4]; this class of problems is commonly referred to as *network tomography*. Active measurement techniques have been proposed that send sequences of probe packets from a set of sources to a set of receivers, and infer link-level metrics of interest from the received packets. Some techniques send probes over unicast paths [5] while others use multicast trees [2, 3]; to cover the entire network, a mesh of paths and/or trees is needed. The bandwidth efficiency of these methods can be measured by the number of probe packets needed to estimate the metric of interest within a desired accuracy. It depends both on (i) the choice of paths/trees over which sequences of probes are sent and on (ii) the number of probes in each sequence. Clearly, there is a tradeoff between bandwidth efficiency and estimation accuracy; it is desirable to improve both as well as to keep computational complexity low.

In this work, we propose the use of network coding techniques [6, 7] to improve several aspects of network monitoring. The basic idea of network coding is to allow intermediate nodes to process the incoming packets before forwarding them. The set of operations that intermediate nodes perform are referred to as a network code; typically, linear codes are used [7]. The idea of network coding (albeit difficult to apply to today's Internet routers) can be gracefully applied to overlay networks, where the network designer has control over the intermediate nodes in the overlay; furthermore, we envision the use of network coding only for special probe packets and not for forwarding regular traffic.

Allowing nodes in an overlay network to perform network coding can improve all aspects of network tomography, namely bandwidth usage, estimation accuracy, and complexity in choosing which paths to monitor. More specifically, the use of network coding allows to (i) eliminate the overlap between paths and/or trees needed to cover the entire network (ii) use less probes per sequence to achieve a certain accuracy, by intelligently using not only the number, but also the content of received probes and (iii) reduce the complexity of choosing which paths to monitor. In general, we believe that the idea of combining network coding with network tomography techniques is very promising. As a concrete example, we show how to use simple linear coding to improve the inference of link-level loss rate from end-to-end measurements.

The structure of the paper is as follows. Section 2 summarizes related work in monitoring and network coding. Section 3 discusses a motivating example, which demonstrates the key points of our approach. Section 4 discusses more formally the application of network coding to network tomography. Section 5 presents simulation results and section 6 concludes the paper.

2 Related Work

Within the broad area of network monitoring, we are interested in network tomography for overlay monitoring, i.e., in using active (unicast or multicast) probes between overlay nodes to infer characteristics of the Internet paths between them. There has been a significant amount of work in this area. Our novel contribution is the application of ideas from network coding.

In the context of resilient overlay networks (RON) proposed in [1], $O(n^2)$ paths are monitored, where n is the number of end-hosts. The authors in [5] chose a basis of $k << n^2$ paths to measure and compute the properties of all n^2 paths. Different studies have measured a variety of characteristics, including loss metrics [2, 3], delay-metrics, or generic distance metrics [8, 9]; in this paper, we are interested in inferring packet loss rates.

The problem of inferring the link loss rates from end-to-end measurements is typically underconstrained. Network tomography [2, 3] originally used multicast probes to exploit correlation on shared parts of the path. Later on, techniques were developed to use unicast probes instead. Good estimators have been developed to infer link loss rates from the multicast probes observed at the receivers. The trees over which multicast probes are sent are either considered given, or are selected by solving a covering problem [10], which is NP–hard. [4] developed a technique for jointly estimating the topology and the link characteristics. In contrast, one of the results of this paper is that using network coding makes the selection of probe routes an LP problem.

The area of network coding emerged in 2000 [6, 7], and since then it has attracted a lot of interest [11] due to its potential for contributions to the theory and practice of networks. The core idea in network coding is to allow intermediate nodes to combine packets before forwarding them. In particular, it is well known that maximizing the throughput when multicasting (a problem known as packing Steiner trees) is NP-hard. In [12], it was shown that by combining independent network/information flows at intermediate nodes, the throughput can be maximized using polynomial-time algorithms. We use this idea to choose routes (over which we send probe packets) that cover the network we want to monitor; the solution can now be found in polynomial time, which is an improvement over [10]. The difference from [12] is that instead of maximizing throughput, we are interested in minimizing the cost of sending probes.

In practice, multicast is not widely supported and unicast probes are used instead. In order to emulate multicast behavior and exploit correlation, ideas such as back-to-back packets have been proposed. Unfortunately, ensuring that two packets will stay back-to-back until their destination is impractical, as it requires perfect synchronization, knowledge of delays in every network element and no cross-traffic. Network coding offers an alternative solution: two incoming packets are forced to share fate downstream, not by keeping them back-to-back but by combining them into a single packet, using network coding at the junction node.

Although network coding cannot, and should not, be widely applied to Internet routers, it could naturally be used in overlay networks where nodes have enhanced functionality; this has

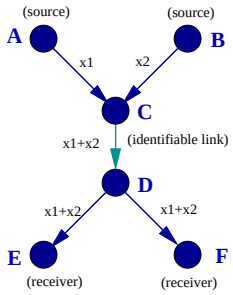

Figure 1: Main example. A and B are sources, E and F are receivers, C adds (or *xor*-s incoming packets, D copies incoming packet to both outgoing links.

also been proposed by [13]. To the best of our knowledge, our work is the first to use network coding for measurements and inference in overlay networks. Independently, passive inference of network characteristics from an already established multicast, network-coded connection has been recently investigated in [14].

3 Motivating Example

Consider the network depicted in Fig. 1. Nodes A and B send probes and nodes E and F receive them. The intermediate nodes C and D can look at the content of the incoming packets and form packet(s) to forward to their outgoing link(s). Every link loses a packet according to an i.i.d. Bernoulli distribution, with probability unknown to us. We are interested in estimating these loss probabilities in all links, namely $p_{AC}, p_{BC}, p_{CD}, p_{DE}, p_{DF}$.

The basic idea of our scheme is the following. Node A sends to node C a probe packet with payload that contains the binary string $x_1 = [1\ 0]$. Similarly, 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 bit-wise xor operation. Node D sends the incoming packet x_3 to both outgoing links DE and DF. All operations happen in one time slot, that will be defined later.

In every time period, probe packets (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 p_{link}. The possible outcomes observed at nodes E and F are summarized in the left two columns of Table 1. The five right columns at the same table show the combination of loss and success events in 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 $(1-p_{AC})p_{BC}(1-p_{CD})(1-p_{DE})(1-p_{DF})$. Similarly, we can write the probability of each of the 10 observed events as a function of the link loss probabilities.

Our goal is to estimate $p_{AC}, p_{BC}, p_{CD}, p_{DE}, p_{DF}$ from the contents of the received probes at nodes E and F. By repeating the experiment a number of times, we observe how many times each event occurs. We can then use standard Maximum Likelihood (ML) estimation to infer the underlying link loss rates. The ML estimator identifies the link-loss rates that would, with higher probability, result in obtaining our particular set of data.

In contrast, the multicast-based tomography approach would use two multicast trees rooted at nodes A and B and ending at E and F, in order to cover all five links at least once. Our approach has the following advantages:

• The two multicast trees approach would not distinguish the loss-rates between links AC and CD (or similarly BC and CD). Our approach solves this problem. As we will see in Section 4.2, Fig. 1 provides the intuition for identifying a link even in general topologies.

• In every experiment we send exactly one probe on every link, which is the minimum possible required to cover the entire graph. As we will see in Section 4.3, this observation holds for any general graph. In contrast, the two multicast trees would overlap and thus send two probes on

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 1: 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.

each one of the links CD, DE and DF.

• Finally, by looking not only at the number of received probes but also at their contents, we are able to infer additional information.

The example in this section demonstrated the key ideas and benefits of our approach. The aforementioned observations gracefully generalize in general graphs.

4 Application to Network Tomography

4.1 Problem Statement

We are given an overlay network represented as a directed graph $G = (V, E)$, and each link (or edge) $e \in E$ of the network has loss probability p_e, $0 \leq p_e < 1$. We assume that a packet traversing a link e is lost with probability p_e, and that losses are independent. We are also given a set of $S \subseteq V$ of nodes that can act as sources of probe packets, a set $R \subseteq V$ of nodes that can act as receivers of probe packets, and a set of links $L \subseteq V$. The goal is to estimate the link loss probabilities $\{p_e, e \in L\}$, by sending probe packets from nodes in S to nodes in R.

Our performance measure is a cost function proportional to the link utilization required to estimate $\{p_e, e \in L\}$ with a desired accuracy. We will assume that the desired accuracy can be achieved by using a rate of α probe packets. Without loss of generality we can assume that each edge of our graph has capacity α.

This is a typical problem statement in the network monitoring literature [10]. The new idea in this paper is that we assume that nodes in V have the capability to linearly combine incoming packets; also, in estimating the link loss-rates we take into account not only the number of received probe packets but also their contents.

Requirements. To deploy our approach we need nodes in the overlay V to have the following capabilities:

1. A node can look at the contents of several packets arriving from different incoming links, and linearly combine them to create an outgoing packet.
2. A node can send replicate and transmit a copy of the same packet to several outgoing links.
3. The node operates in time slots. If a packet does not arrive within the time slot, it is considered lost.

The first assumption is necessary for schemes that use network coding in overlay networks [13]. The second assumption is equivalent to overlay multicast. The third is a design issue: the duration of the time slot (time that the node waits for incoming packets before declaring them lost) should be carefully chosen based on the frequency of probes, the network delays, the synchronization between sources etc. The assumed capabilities of the nodes are realistic in the context of overlay networks, where the capabilities and operations of a node are completely

controlled by the designer/operator of the overlay. This is in contrast to Internet routers, on which the end-user has no control. Furthermore, we envision using network coding only for measurement probes and possibly other control traffic, and not necessarily for the bulk of regular traffic.

Problem Decomposition. The problem can be decomposed into the following parts:
1. Identifiability. For each link $e \in L$, decide whether its loss probability can be inferred.
2. Minimum Cost Covering. Select the paths through which probe packets will be routed, and the nodes at which they will be linearly combined, so that the links of interest are covered at minimum cost (to be defined).
3. Packet Design. Select the contents of the packets, and the operations performed at intermediate nodes.
4. Estimation. Process the collected probes at the receivers and estimate loss-rates for $l \in L$.

4.2 Identifiability Problem

Similarly to [3], we say that a link $e \in E$ is *identifiable* if it is possible to estimate the associated loss-rate p_e by sending probing packets from nodes in S to nodes in R. The following theorem gives necessary and sufficient conditions for identifiability.

CD is the directed link from node C to node D; (C, D) is a path from C to D.

Theorem 1. *Given $G = (V, E)$ and sets S and R, a link CD is identifiable if and only if both conditions hold:*
Condition 1: *At least one of the following holds:*
(a) $C \in S$.
(b) *There exist two edge disjoint paths (X_1, C) and (X_2, C) that do not employ edge CD with $X_1, X_2 \in S$.*
(c) *There exists two edge disjoint paths (X_1, C) and (C, X_2) that do not employ CD with $X_1 \in S$, $X_2 \in R$.*
Condition 2: *At least one of the following holds:*
(a) $D \in R$.
(b) *There exist two edge disjoint paths (D, X_1) and (D, X_2) that do not employ edge CD with $X_1, X_2 \in R$.*
(c) *There exists two edge disjoint paths (X_1, D) and (D, X_2) that do not employ CD with $X_1 \in S$, $X_2 \in R$.*

Sketch of Proof: We claim that a link CD is identifiable if C is a source or a branching point, and D is a receiver or a branching point. These are the structures depicted in Fig. 2, where we want to identify the link-loss rate associated with edge CD and interpret the remaining edges as possibly corresponding to paths. It is easy to see that if both conditions are satisfied link CD is identifiable. Conversely, assume the first condition is not satisfied. Then C can only receive one stream of probe packets, since it is connected to one source only. There exists an edge e through which this stream of probe packets arrives to node C. The link–loss rate associated with link CD cannot be distinguished from the link loss–rate associated with link e. □

The above theorem generalizes the intuition of the motivating example. Similar intuition has been independently developed in [4], where a 2x2 topology was studied extensively as the building block of more general topologies; in all previous work, spacing between packets is important (either keeping them back-to-back or exploiting their distance Δ). The underlying requirement of this theorem, e.g. for condition (1b), is not necessarily that probes come to node C through two distinct links, but that two rate–α flows (that have undergone failures i.i.d. Bernoulli distributed) arrive to node C. This is enforced by the α-capacity links. Under these assumptions and $p_e < 1$, identifiability is a topological property of the graph that does not depend on the loss-rate values p_e, as was also discussed in [3].

Example. Keeping the same topology shown in Fig. 1, we now vary the sets of sources and receivers. Fig. 2 depicts four configurations for which link CD is identifiable. Case 1 is our motivating example; Case 2 is similar to a single multicast tree rooted at A; Case 3 uses sources

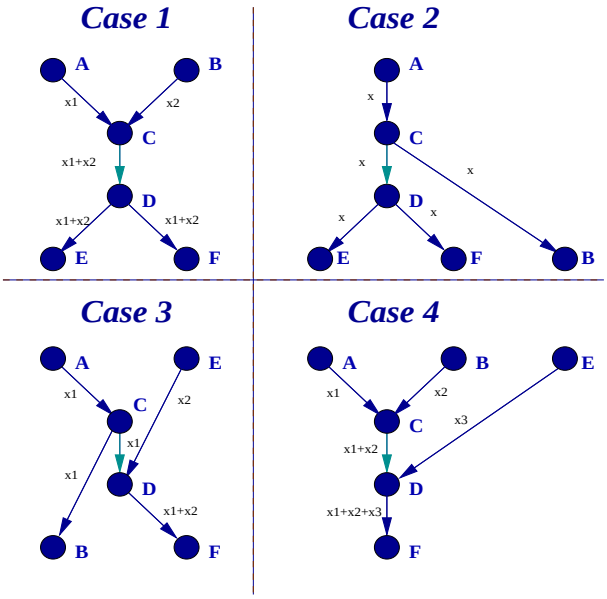

Figure 2: Four configurations that lead to identifiable edge CD (for the same topology).

Case	Network Coding	Multicast Probes
1	all links	DE, DF
2	all links	all links
3	all links	AC, CB
4	all links	no links

Table 2: Identifiable links for the cases in Fig. 2.

A and E and linear combinations whenever two flows meet; Case 4 does the same for sources A, B and E and is equivalent to an inverse multicast tree (with sink at F). Table 2 lists which links are identifiable in all four cases, if we use our network coding approach and if we use the classic multicast probe approach. While our approach is able to identify all links for any sets of sources and receivers, this is not always the case for the classic multicast approach.

It is straightforward to see that we can check in polynomial-time whether a link is identifiable or not, by applying Theorem 1 and examining min-cut conditions on the graph $G - \{CD\}$. For the rest of the paper we will assume that all links in L are identifiable.

4.3 Minimum Cost Covering Problem

Our goal is to estimate the loss probabilities for all links in L at the minimum bandwidth cost. First, we associate a cost proportional to the flow through a link, and select which links to utilize to estimate p_e for all $e \in L$, so as to minimize the total cost.[1] Then, we formulate the minimum cost cover problem as a Linear Program (LP), which allows to solve it in polynomial time, provided that intermediate nodes can combine probes. This is an improvement over the same problem without network coding, which is NP-hard [10].

Intuition. Following an approach similar to [12], we introduce conceptual flows that can share a link without contending for link capacity. We associate with each edge $e_i \in L$ one such conceptual flow f^i. We would like each f^i to bring probe packets to link $e_i = u_i v_i \in L$ in a manner consistent with the conditions of Theorem 1 for edge e_i. Conceptual flows corresponding to different edges e_i share edges without contention, and a total flow f measures the utilization of edges by probe packets. We will use the condition $f^i \leq f$ to express the fact that each packet in f might be the linear combination of several packets of conceptual flows.

[1] This cost function assumes that a desired accuracy for p_e can be achieved by utilizing rate–α probe packets in a manner consistent with the conditions in Theorem 1, but independently of the exact paths through which the probe packets are routed. This is a simplifying, yet standard assumption in network monitoring literature.

Algorithm 1 LP program

$\min \sum_e C(e)f(e)$

$f(e) \leq a \quad \forall e \in E - S_E - R_E$

$f(e) = a \quad \forall e \in \mathcal{L}$

Each conceptual flow f^i, corresponding to $e_i = u_i v_i$, satisfies the constraints:

$f^i(e) \leq f(e) \quad \forall e \in E - e_i$

$f^i(e) \geq 0 \quad \forall e \in E$

$f^i_{in}(\mathcal{S}) = 0$

$f^i_{out}(\mathcal{R}) = 0$

$f^i_{in}(u) = f^i_{out}(u) \quad \forall u \in V - \{\mathcal{S}, \mathcal{R}, u_i, v_i\}$

$a \leq f^i_{in}(u_i) \leq 2a$

$a \leq f^i_{out}(v_i) \leq 2a$

$f^i_{in}(u_i) + f^i_{out}(v_i) \geq 4a$

Notation. Let $C : E \rightarrow R^+$ be our cost function that associates a non-negative cost $C(e)$ with each edge e. We are interested in minimizing the total cost $\sum_e C(e)f(e)$, where $f(e)$ is the flow through edge e. We also denote by $f_{in}(v)/f_{out}(v)$ the total incoming/outgoing flow of vertex v and with $f_{in}(e)/f_{out}(e)$ the total incoming/outgoing flow to edge e. We connect all nodes in $S = \{S_i\}$ to a common source node $\mathcal{S}$ through a set of infinite-capacity and zero-cost edges $E_S = \{\mathcal{S}S_i\}$. Similarly, we connect the nodes in $R = \{R_i\}$ to a common node $\mathcal{R}$ using an infinite-capacity and zero-cost set of edges $E_R = \{R_i\mathcal{R}\}$.

Algorithm 1 summarizes the LP program. The idea is to lower-bound the probe rate $f(e)$, in edge e, given the conceptual flows and the condition $f^i(e) \leq f(e)$. The full proof is omitted here and will be provided in a longer version [15].

A useful special case. If we want to estimate the loss-rate on *all* identifiable edges of the graph (as opposed to a restricted set L) we do not even need to solve the above LP. We can simply have each source send a probe and each intermediate node forward a combination of its incoming packets to its outgoing edges, as in Fig. 2. This simple scheme utilizes each edge of the graph exactly once per time slot and thus has the minimum total bandwidth cost.

4.4 Packet Design Problem

Given the set of flows previously identified, design the content of the probe packets and the processing intermediate nodes should perform. In particular, if intermediate node A receives incoming probe packets $x_1, x_2, \ldots x_l$, we want the linear operation of each different subset of these packets to be distinct. Interestingly, this is related to the problem of designing pn-sequences for Code-Division-Multiple Access Schemes (CDMA), and to the problem of designing training sequences for MIMO channels [15]. In general, intermediate nodes can do operations over a finite field F_q, by treating q bits of each binary probe packets as a symbol of F_q.

A Useful Special Case. If the graph G is a tree, with a subset of the leaves serving as sources and a subset of the leaves serving as receivers of probe packets, there exists a simple packet design solution. For n sources we simply use binary probe packets of length n. Source 1 sends the probe packet [1 0 . . . 0], source 2 the packet [0 1 . . . 0] and generally source i the packet that has 1 in i^{th} position. When incoming packets meet at a node, the node sends a packet to every outgoing edge, whose paylaod is the binary *xor*, i.e. the union of all 1's.

4.5 Estimation Problem

As illustrated in our motivating example, each experiment consists of a set of probe packets sent simultaneously (within the same time slot) from the source nodes, traversing the network and resulting in one of the observed outcomes at the receiver nodes. These outcomes have

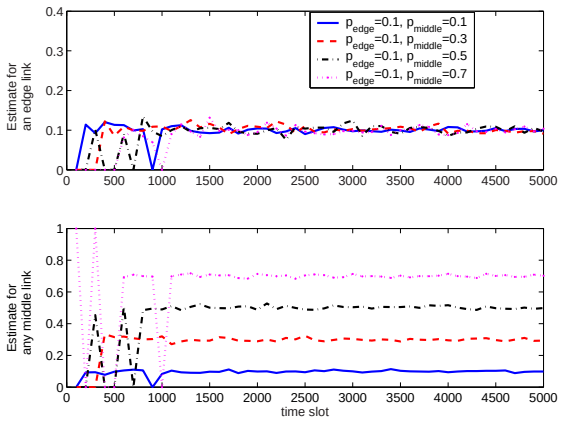

Figure 3: Maximum likelihood estimation of loss rates for the links in Fig.1. Edge links $\{AC, BC, DE, DF\}$ have $p = 0.1$. The middle link (CD) has $p_{CD} \in \{0.1, 0.3, 0.5, 0.7\}$.

a probability distribution that depends on the underlying link loss probabilities. Note that receivers will need to share information about received probes, e.g. by sending a summarizing report to a central node for processing. After a sufficiently large number of experiments, we can count the number each event appeared.

The estimation problem is to estimate the loss probabilities of the links of interest, based on the collected data. So far, we have used the Maximum Likelihood (ML) estimation, which calculates the link-loss rates that would most likely result to the observed appearances.[2] To avoid the complexity of ML, one could use the EM algorithm proposed in [3], to approximate the optimal ML solution. Moreover, one can use heuristic methods to approximate the exact solution, that try to pack for example the core structures in Fig. 2 and use the respective ML estimator for each "simplified" network; we are currently working on such methods in [15].

5 Simulation Results

In this section, we present preliminary simulation results for interesting special cases.

Main example. Consider the main example of Fig. 1. Let us assume that all edge links (AC, BC, DE, DF) have the same loss prob. $p = 0.1$ and that the middle link (CD) has higher loss prob. p_{CD}; let us consider 4 scenarios, corresponding to different values of $p_{CD} \in \{0.1, 0.3, 0.5, 0.7\}$, while keeping p=0.1 on all other links. The goal is to estimate the loss probability for all links. In Fig. 3, we show the maximum likelihood estimates for an edge link (AB) and for the middle link (CD). The same color in the top and the bottom plots indicate that the estimates are obtained for the same case, i.e. value of p_{CD}).

This simple figure confirms what we intuitively expected. The estimates successfully approximate the actual loss probabilities ($p_{edge} = 0.1$ for the edge link and p_{CD} for the middle link, respectively). The MLE oscillates in the first time slots, and converges after it has collected enough probes, (here, in roughly 1000 timeslots, but the number depends on p_{edge} and p_{CD}). Of course, this is just a proof of concept and a more thorough study of the estimation error need to be provided in a longer version [15].

Comparing different configurations. For the same topology, let us now consider the four possible configurations for sending probes, shown in Fig. 2. These can be thought as core-structures for link-loss estimation. In each case, we use MLE to process the probes at the receivers. From the simulations, we made the following observations (we omit the plots due to lack of space.) First, our method was able to identify the loss rates for all links, in all configurations, as predicted in Table 2. Second, the four configurations in Fig. 2 lead to

[2] I.e., given an observed number of appearances of events (in our case, number of times each outcome is observed at the receivers), estimate the model parameters (in our case, link-loss rates), such that the probability of observing the set of appearances is maximized.

Table 3: Possible outcomes and events for Case 2

Received at			Is link ok?				
E	F	B	AC	BC	CD	DE	DF
–	–	0	Multiple possibilities				
–	–	x_1	1	1	Multiple possibilities		
–	x_1	–	1	0	1	0	1
–	x_1	x_1	1	1	1	0	1
x_1	–	–	1	0	1	1	0
x_1	–	x_1	1	1	1	1	0
x_1	x_1	–	1	0	1	1	1
x_1	x_1	x_1	1	1	1	1	1

different estimation accuracy and convergence time, despite the fact that the topology is the same and each link is used exactly once per time slot.

The reason is that the four configurations differ in the possible observed outcomes and in the formula used in MLE (omitted here due to lack of space). For example, Table 3 shows the possible outcomes, and the events that led to these outcomes, for Fig. 2 - Case 2; clearly, this is a different set from Table 1 that corresponded to Fig. 2 - Case 1. Which configuration performs better, depends on the value of the loss probabilities on all links.

General Topologies. It is straightforward to apply our approach to any *tree topology*. This was demonstrated in all cases of Fig. 2. Given some leaves that act as sources and receivers, there is a unique orientation of the links. Source S_i sends probe $x_i = [0 \dots 1\ 0 \dots 0]$ (all 0's and an 1 at the i^{th} position). Each intermediate node forwards the *xor* of its incoming probes to all outgoings. Receivers estimate link loss rates using MLE.

However, a general topology may contain *cycles*. An example is shown in Fig. 4: a simple xor of packets can cancel out those appearing for an even number of times in the summation. To solve this problem, we can allow linear operations (here addition) over a larger field. E.g. S_1 and S_2 send x_1 and x_2; intermediate nodes add incoming packets, e.g. B sends $x_1 + x_2$, ... D sends $2x_1 + x_2$. Notice that if D used xor instead, R_1 would get confused: $(x_1 \otimes x_2) \otimes x_1 = x_2$.

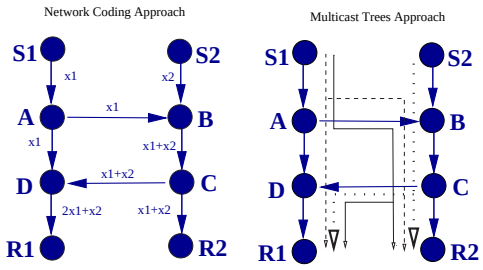

Figure 4: Example of network with a cycle.

In Fig. 5, we show preliminary simulations for this example. First, the ML estimates for the network coding approach seem to approximate well the actual link loss rates. Second, we are able to identify all links; in contrast, the multicast approach - shown in right side of Fig. 4- would fail, even three multicast trees, as the problem would still be under-constrained. Finally, we send one probe per link, while the multicast trees would overlap in this topology.

6 Conclusion and Future Work

In this paper, we proposed the use of network coding to improve network tomography in overlay networks. We demonstrated the potential for improving several aspects of the problem:

• Identifiability: we can estimate the loss rate of any identifiable link - while the multicast trees approach heavily depends on the topology and the set of sources and receivers.

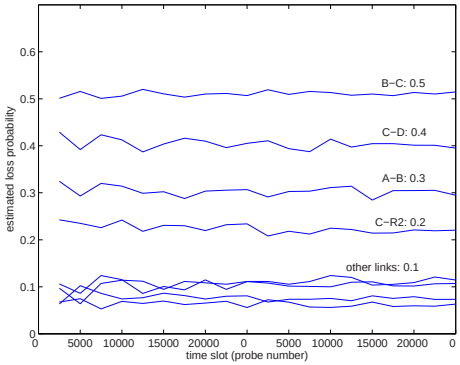

Figure 5: Estimates for the example with cycle, using network coding and MLE. Actual loss rates: p_{CR_2}=0.2, p_{AB}=0.3, p_{CD}=0.4, p_{BC}=0.5, all other p=0.1.

• Bandwidth efficiency: exactly one probe is transmitted over every link per time slot, which is the minimum possible; this is achieved by combining multiple probes into one when flows overlap, and by intelligently using the content of the probes for inference.
• Complexity: choosing paths for sending probes got reduced from NP-hard to an LP problem (to identify a subset of links), or eliminated (to identify all links).
• Estimation: there is potential for improving accuracy, by using the content, not only the number, of received probes.
These benefits come at the overhead of some additional processing at overlay nodes. We are currently working on extending several aspects of this work, [15], including: probe packet design algorithms; simplified calculations of ML estimation over acyclic networks; inference of metrics other than loss rates; and simulations over realistic Internet topologies.

References

[1] D.Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient overlay networks," *in Proc. of* 18^{th} *ACM SOSP*, Canada, Oct. 2001.

[2] 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.

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

[6] 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.

[7] S-Y. R. Li, R.W. Yeung, N. Cai, "Linear network coding," *IEEE Trans. on Inf. Theory,* Vol. 49, Feb. 2003.

[8] P. Francis, S. Jamin, C. Jin, Y. Jin, V. Paxson, D. Raz, Y. Shavitt, L. Zhang, "IDMaps: A global internet host distance estimation service," *in IEEE/ACM Trans. on Networking*, Oct. 2001.

[9] T.S.E. Ng and H.Zhang, "Predicting internet network distance with cordinated-based approaches," *in Proc. of IEEE INFOCOM 2002.*

[10] M. Adler, T. Bu, R. K. Sitaraman and D. Towsley, "Tree layout for internal network characterizations in multicast networks," *in Proc. of ACM NGC 2001*, Nov. 2001.

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

[12] Z. Li, B. Li, D. Jiang, L. C. Lau, "On achieving optimal throughput with network coding," *in Proc. of IEEE INFOCOM 2005.*

[13] 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.

[14] T.Ho, B. Leong, Y. Chang, Y. Wen and R. Koetter, "Network monitoring in multicast networks using network coding", *in International Symposium on Information Theory (ISIT) 2005.*

[15] C. Fragouli and A. Markopoulou, "Network coding for network tomography", *technical report in preparation.*