Network coding for network tomography

Srinivasan, R.

Award date:
2006

Eindhoven University of Technology
Department of Electrical Engineering
Signal Processing Systems

TU/e technische universiteit eindhoven

Network coding for network tomography

by Ramya Srinivasan

Master of Science thesis

Project period: January 2006 – December 2006

Report Number: 37-06

Supervisors:
Name (TU/e): Dr. Frans Willems, Prof. Jan Bergmans
Name (The school of computer and communication sciences, EPFL, Lausanne):
Prof. Suhas Diggavi,
Prof. Christina Fragouli,
Prof. Athina Markopoulou

Network coding for network tomography

M.Sc. Thesis

Ramya Srinivasan

The school of computer and communication sciences,
EPFL, Lausanne

Electrical Engineering Department
Technical University of Eindhoven

Report Date: October 14, 2006
Thesis Committee: Dr. Frans Willems, Prof. Jan Bergmans
Prof. Suhas Diggavi, Prof. Christina Fragouli
Prof. Athina Markopoulou

Abstract

Robust measurements of network dynamics (such as topology, packet loss rate, delay, and failures) are becoming increasingly important to the design and operation of large inter-networks like the Internet. Distributed Internet applications thus try to detect and recover from failures or degraded performance of the underlying Internet infrastructure.

One approach to doing this requires using special purpose network support in order to gain access to statistics collected at internal nodes, wherefrom network performance can be gauged. However, such a solution can be difficult or even infeasible since it involves introducing measurement mechanisms into a wide range of routers in large and administratively diverse networks. Therefore, devising methods for inference of desired network characteristics by means of end-to-end measurements, which obviate the need for access to internal nodes, forms a very important class of problems. It is this methodology that has been termed as 'network tomography'.

In the past decade, several approaches have been proposed towards this end, 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. Until recently, the traditional methods of doing this have been by sending probes either over unicast paths or by using multicast trees to cover the entire network, or some desired subset of links. 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. 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.

This project explores the novel method of inferring the success rates of links (link probabilities) in a network by employing network coding techniques. The area of network coding emerged in 2000, and since then, has attracted a lot of interest 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. It has been shown that the network coding approach to tomography can decrease the bandwidth used by probes, improve the accuracy of estimation, and decrease the complexity of selecting paths or trees through which to route probes, when compared to unicast or multicast based inference.

Given a network topology whose link probabilities must be estimated, there exist a number of possible solutions to the network coding based inference prob-

lem depending on the choices of nodes that serve as sources, receivers and coding points (nodes at which coding is done). Our observation is that each such solution may lead to a different estimation accuracy. We will call each solution a configuration. In this project, we develop a number of heuristics for estimation of link probabilities for any arbitrary coding scheme (i.e. the configuration) that is applied to a topology. Our starting point towards these heuristics is the maximum likelihood (ML) estimator and its simplified implementation for multicast probes over tree topologies presented in [1].

In deriving some of these heuristics we use an estimator that had been previously developed for multicast based systems as a convenient starting point. The configuration that a given network topology is made to assume together with the heuristic employed for the estimation therefore constitutes the solution to the inference problem.

Various networks are studied during the course of this thesis, where different solutions, including the multicast method to the inference problem are sought. These are then characterised by means of suitable metrics and a comparative study is carried out. The simulation results very conclusively show that in general, well chosen network coding estimators outperform multicast based estimators in terms of the quality of estimation (as well as bandwidth efficiency).

Contents

Abstract i

Table of Contents 1

1 Introduction 3
- 1.1 Traditional methods of inference . . . 4
  - 1.1.1 Unicast based inference . . . 4
  - 1.1.2 Multicast based inference . . . 6
- 1.2 Network coding based inference . . . 7
  - 1.2.1 Minimum Cost Covering Theorem . . . 9
  - 1.2.2 Identifiability theorem . . . 11
- 1.3 Problem motivation . . . 12

2 Link probability estimation 15
- 2.1 Model . . . 15
- 2.2 Maximum Likelihood Estimation . . . 16
- 2.3 Performance metrics . . . 20
- 2.4 Problem Statement . . . 21
- 2.5 Contributions of this project . . . 22

3 Reversibility theorem 23
- 3.1 Dual Configurations and Reversibility . . . 23
- 3.2 Measuring Directional Networks: . . . 26
- 3.3 Advantages of network coding based tomography . . . 27

4 Network coding based inference 29
- 4.1 Estimation problem: basic configurations and other small networks 29
  - 4.1.1 Basic configuration 1 . . . 30
  - 4.1.2 Basic configuration 2 . . . 34
  - 4.1.3 Basic configuration 3 . . . 34
  - 4.1.4 Basic configuration 4 . . . 36
  - 4.1.5 Other small sized networks : 4-,5-,7- and 9-link trees . . . 38
- 4.2 Estimation problem : observations and conclusions . . . 40
- 4.3 Choice of coding scheme: basic configurations and other small networks . . . 45
- 4.4 Convergence properties . . . 51

5 Estimation algorithms **55**
5.1 ML estimator for a multicast tree 55
5.2 ML estimator for an inverse multicast tree 57
5.3 ML estimator for an inverse multicast-multicast type tree 59
5.4 Estimation of a tree with an arbitrary coding configuration . . . 60
5.4.1 Suboptimal estimator 1 61
5.4.2 Suboptimal estimator 2 : subtree decomposition 68
5.5 Suboptimal estimator 3 : Message passing algorithm 70

6 Simulation results **81**
6.1 45 link network 82
6.2 200 link network 93
6.3 Conclusions 94

7 Cyclic networks **101**

8 Conclusions and recommendations **107**

A Fisher Information Matrix is non-singular **109**
A.1 Multicast and inverse multicast trees 109
A.2 General coding scheme on a tree 111

Bibliography **118**

1

Introduction

The design, operation, control and management of large internetworks like the Internet requires robust measurements of network dynamics for measuring their performance. There are two different approaches to monitoring network performance. These are :

(i) Collecting statistics at internal nodes and using network management packages to generate link-level performance reports.

(ii) Characterizing network performance based on end-to-end behavior of point-to-point traffic such as that generated by TCP or UDP.

In large scale networks, end-systems cannot rely on the network itself to cooperate in characterising its own behaviour. This is because gaining access to a wide range of routers in an administratively diverse network can be difficult, thus making it impractical to monitor every link on an end-to-end path. Moreover, introducing new measurement mechanisms into the routers themselves is difficult as well, because it requires persuading large companies to alter their products. This therefore creates a significant drawback to the first approach.

A large volume of research has stemmed from adopting the second of the two approaches, that explore feasible solutions to characterising network performance by means of end-to-end measurements. It is this methodology that has been termed as 'network tomography', and it is defined as the problem of identifying topology and inferring link-level performance parameters such as packet drop rate or delay variance using only end-to-end measurements. Network tomography thus, is a vital component in current efforts to transform large scale internetworks into well-understood and predictable systems.

In this thesis, the focus is active tomography where sources send probe packets towards the receiver. Our application is to estimate the success rates on individual links within a given network. The success probability (or link probability) of a link e is denoted as α_e and this is one minus the packet drop rate i.e. one minus the loss probability of the link.

Section 1.1 contains some literature study of traditional multicast-based and unicast-based inference methods. Section 1.2 introduces the concept of network coding based inference, which is the subject of study of this project. Section 1.3 explains the motivation behind this research work.

1.1 Traditional methods of inference

Till recently, the two traditional methods of estimating network internal characteristics were by the measurement of end-to-end (i) unicast traffic and (ii) multicast traffic. While many networks do not currently support multicast due to its scalability limitations (routers need to maintain per group sate), and lack of access control, unicast inference in contrast is easily deployed on a scalable commercial infrastructure. On the other hand, it is difficult to determine the performance characteristics of individual links from end-to-end measurements of unicast traffic, while multicast traffic provides a more suitable framework for inference. Moreover, multicast solutions are more bandwidth efficient.

In inferring link probabilities of a network, it is assumed that the topology of the network is known in advance. The analysis of networks with a single source and multiple receivers that can be represented as an acyclic graph (tree) $G = (V, E)$, consisting of a set of nodes V that represent the source, routers and receivers in the network and a set E of links has been done in previous works such as [1], [3] and [5]. The level $l(k)$ of some node k in the tree is defined as the number of links in the path from the (single) source to node k. The set of all link probabilities of the tree is denoted by α.

End-to-end measurements made on an isolated subpath (a subpath consisting of two or more links in which internal nodes have only one child) do not provide sufficient information to resolve the individual losses in the isolated subpath. Thus, if isolated subpaths exist in the network under study, we remove the internal subpath nodes and use a single composite link to represent the isolated subpath. This forms what is called a 'logical tree', and is demonstrated in Figure 1.2.

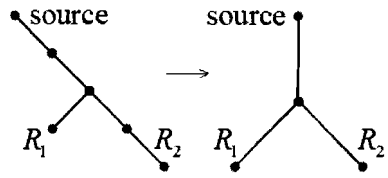


Figure 1.1: An example of how a the original tree representation of a two-receiver network is converted into its logical equivalent, as required by the inference algorithm for identifiability.

All nodes in the logical tree thus have at least two children, apart from the source (one child) and the receivers (no children). All the links in such a tree are then said to be 'identifiable', because there is a one-to-one mapping between the measurements and the set of link probabilities, α, of the logical tree. Figure 1.1 depicts the logical tree of the simplest non-trivial network (a triad).

1.1.1 Unicast based inference

The inference of network losses using end-to-end unicast traffic is researched extensively in [5]. It is shown that inference based solely on measurement of unicast traffic is not possible, and resolving losses requires identifying and incorporating reasonable prior information or constraints. The inference procedure

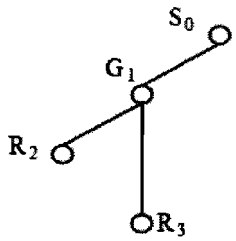


Figure 1.2: A simple network involving a source S_0, a router G_1 and two receivers R_2 and R_3.

requires setting up a network modeling framework based on the correlation between unconditional (single packet) losses and conditional (back-to-back) packet losses. Based on internet loss observations and theoretical results from queue models, a coupling between the unconditional and conditional loss probabilities is then established and this coupling is used in the inference procedure.

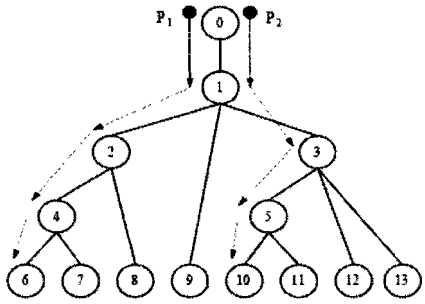


Figure 1.3: Unicast based inference of network internal characteristics : An example of a depth-4 tree. The arrows depict the paths traversed by the first packet (P_1) and second packet (P_2) of a $(6, 10)$ packet pair.

Measurements made using back-to-back pairs provide an opportunity to generate spatio-temporal statistics of the system. By back-to-back packet pairs we mean two packets that are sent one after the other by the source, possibly destined for different receivers, but sharing a common set of links in their paths. The reason for introducing back-to-back measurements is that the correlation between the link-level success probabilities of closely-spaced packets traveling along the same links can be anticipated and it is this correlation can be used for the estimation purpose.

If we wish to estimate the success probabilities for the entire tree, then for each pair of receivers (r_j, r_k), we gather the following statistics using back-to-back probing pairs. If we use the term "(r_j, r_k)-pairs" to denote those pairs in which the first packet was destined for receiver r_j and the second for receiver r_k, then the nature of a $(6, 10)$ pair for example is shown by the two arrows in Figure 1.3. The first packet P_1 is sent to receiver 6 and the second packet P_2 to receiver 10. In the case of the (r_j, r_k) receiver pair, the measurements to be made are: the number $n_{j,k}$ of trials wherein the first packet is received by receiver r_j, the number $m_{j,k}$ of trials wherein both the first packet is received by r_j and the second packet is received by r_k. Similarly, $n_{k,j}$ and $m_{k,j}$ are

measured. In general, if there are N receivers, then there are N^2 back-to-back receiver pairings.

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. An advantage of inference using multicast traffic as opposed to unicast traffic is that it obviates the need for back-to-back probing pairs.

1.1.2 Multicast based inference

The inference of network losses using end-to-end multicast traffic is researched extensively in [1]. The key to this approach is that multicast traffic introduces correlation in the end-to-end losses measured by receivers. This correlation can, in turn, be used to infer the loss behaviour of the links within the multicast routing tree spanning the sender and receivers.

Using this approach, a maximum likelihood estimator (MLE) of the link loss rates within a multicast tree connecting the sender of the probes to a set of receivers has been developed in [1]. These estimates are derived under the assumption that link losses are described by independent Bernoulli losses, in which case the problem is that of estimating the link loss rates given the end-to-end losses for a series of n probes.

The subtree rooted at some node $k \in V$ is denoted by $G(k) = (V(k), E(k))$. If the set of receivers is denoted as R, then the 'receiver subset' of node k is $R \bigcap V(k)$ and is denoted by $R(k)$. If we wish to estimate the success probabilities for the entire tree, the measurements required to be made are as follows: for each node $k \in V$, measure the number of trials, say, $\gamma(k)$ wherein a packet is received by at least one receiver in $R(k)$. It is shown that the estimates derived are strongly consistent (converge almost surely to the true loss rates) and moreover, the asymptotic normality property of MLEs allows deriving an expression for their rate of convergence to the true rates as n increases.

Another strong advantage of using multicast rather than unicast traffic is efficiency. N multicast servers produce a network load that grows at worst linearly as a function of N. On the other hand, the exchange of unicast probes can lead to local loads which grow as N^2, depending on the topology. We illustrate this in Figure 1.4 In this example, $2N$ servers exchange probes. For unicast probes, the load on central link grows as N^2; for multicast probes it grows only as $2N$.

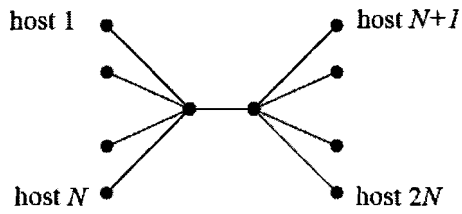


Figure 1.4: $2N$ servers exchange probes. For unicast probes, the load on the central link grows as N^2; for multicast probes, it grows only as $2N$.

The multicast based inference method will be studied in detail during the course of this project because it will serve as a starting point for our analysis

of network coding based inference procedures. The inference problem requires two questions to be addressed, which are outlined below.

Given a set of links $L \subseteq E$ whose behaviour is of interest, a set of minimum cost multicast trees must be chosen within the network which cover the links in L in a way that their associated link loss rates can be inferred. This is called the 'minimum cost tree cover problem'. Resolution of this problem is especially important as poorly designed sets of measurements can easily overwhelm network resources. It has been established [9] that the cover problems are NP-hard and that in some cases, finding an approximation within a certain factor of optimal is also NP-hard.

The next question is whether the set of trees that cover the links in L will allow one to determine the behaviour of all the links in L. This is called the 'identifiability problem'.

In connection with these two issues, three versions of these problems are defined : the weak, strong, and medium cover problems. Briefly, the weak cover problem is based on the assumption that it is sufficient that each link of interest appear in at least one tree. The strong cover problem requires that each link occur between two branching points in at least one tree. The medium cover problem relaxes this last requirement and instead requires that the set of trees covering the link provide enough information to determine the link measure (link probability, in this project) of interest.

If we wish to identify the links in the topology shown in Figure 1.5 by means of a strong cover, we see that only 4 of the links are identifiable. Moreover, we need to perform the experiment 4 times on 4 different multicast trees that will cover these four links in a strong cover.

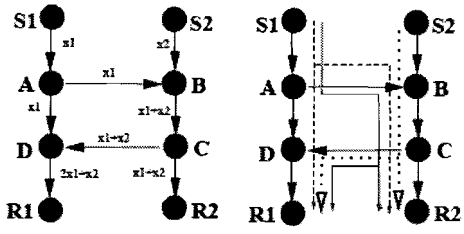


Figure 1.5: A network wth a cycle : all links are identifiable with the network coding approach; only links S_1A, S_2B, DR_1, CR_2 are identifiable (from a strong cover).

Now the purpose of the discussion above is to compare the solution to the two problems explained as offered by the traditional multicast technique, that has just been explained, against the solutions offered by the network coding technique. The concept of network coding is briefly introduced below and then we address the problems of minimum cost covering and identifiability and show why network coding is advantageous to the traditional multicast approach.

1.2 Network coding based inference

The area of network coding emerged in 2000 [6, 7], and since then it has attracted a lot of interest [13] due to its potential for contributions to the theory and

practice of networks. The application of network coding to network tomography was introduced in [2]. In this work, we use this work as a starting point and study the use of network coding techniques 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.

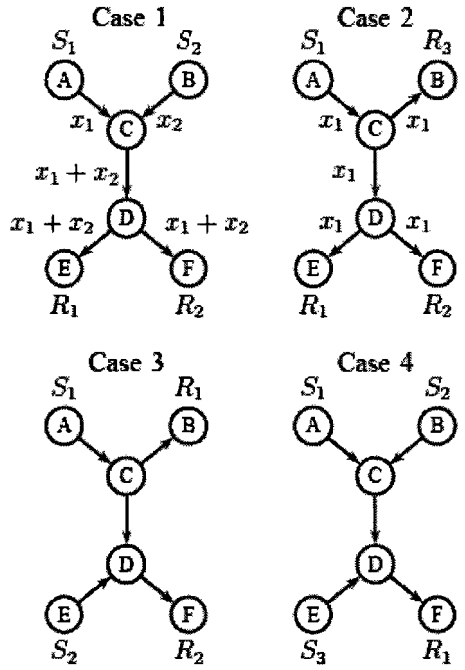


Figure 1.6: Four 5-link configurations which lead to all 5 links being identifiable.

Consider as an example, the first network depicted in Figure 1.6.

Example 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 the corresponding success probabilities in all links, namely α_{AC}; α_{BC}; α_{CD}; α_{DE}; α_{DF} from the contents of the received probes at nodes E and F.*

The basic idea of applying network coding to this case is the following. Node A sends to node C a probe packet with payload that contains the binary string $x_1 = [10]$. Similarly, node B sends probe packet $x_2 = [01]$ 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 $x_3 = [11]$, and forwards it to node D; more generally,

$x_3 = x_1 \oplus x_2$, where $\oplus$ 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[1].

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: the transmitted packet traverses every link in $\{AC;BC;CD;DE;DF\}$, with probability α_{link}. The possible outcomes observed at nodes E and F are summarized in the left two columns of Table 4.1. The next five columns at 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 $\alpha_{AC}(1-\alpha_{BC})(1-\alpha_{CD})(1-\alpha_{DE})(1-\alpha_{DF})$. Similarly, we can write the probability of each of the 10 observed events as a function of the link loss probabilities. The last column gives each of these event probabilities. The goal is to estimate $\alpha_{AC}; \alpha_{BC}; \alpha_{CD}; \alpha_{DE}; \alpha_{DF}$ from the contents of the received probes at nodes E and F. By repeating the experiment a number of times, we can observe how many times each event occurs and then use standard Maximum Likelihood (ML) estimation to infer the underlying link success rates. The ML estimator identifies the link-success 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. The network coding 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). The network coding approach solves this problem.*
- *In every experiment we send exactly one probe on every link, which is the minimum possible required to cover the entire graph. In contrast, the two multicast trees would overlap and thus send two probes on 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.*

1.2.1 Minimum Cost Covering Theorem

With the constraints of network topologies and link capacities, achieving the optimal end-to-end throughput in data networks (a problem known as packing Steiner trees) with single or multiple unicast, multicast and broadcast sessions has been known to be a fundamental but computationally hard problem. Thus, the problem of achieving optimal throughput in such data networks, leads to solving NP-complete problems. However, in [8] it is shown that in networks that employ network coding, computing the strategies to achieve the optimal end-to-end throughput can be performed in polynomial time. In [8], it was shown that by combining independent network/information flows at intermediate nodes, the throughput can be maximised using polynomial-time algorithms. This idea can then be used [2] to choose routes (over which we send probe packets) that cover

[1]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 network we want to monitor; the solution can now be found in polynomial time, which is an improvement over [9]. The difference from [8] is that instead of maximizing throughput, we are interested in minimizing the cost of sending probes.

The goal is to estimate the loss probabilities for all links in L at the minimum bandwidth cost. First, a cost is associated that is proportional to the flow through a link, and then the links to be utilized to estimate α_e for all $e \in L$ are selected so as to minimise the total cost. Then, the minimum cost cover problem can be formulated as a Linear Program (LP) (done in [2]), which allows to solve it in polynomial time, provided that intermediate nodes can combine probes. The LP is reproduced below.

Notation: Each link $e_i \in L$ is associated with a conceptual flow f^i and conceptual flows of different links in L share links without contending for link capacity. If the total flow through an edge is f, then $f^i \leq f$ expresses the fact that each packet in f might be a combination of several packets of conceptual flows. Let $C : E \rightarrow R^+$ denote the cost function that associates a non-negative cost $C(e)$ with each link e. The total cost $\sum_e C(e)f(e)$ must be minimised, where $f(e)$ is the flow through e. $f_{in}(v)/f_{out}(v)$ is the total incoming/outgoing flow of vertex v and $f_{in}(e)/f_{out}(e)$ is the total incoming/outgoing flow to edge e. All nodes in $\{S_i\}$ are connected to a common source node $\mathcal{S}$ through a set of infinite-capacity and zero-cost links $E_S = \{\mathcal{S}S_i\}$. Similarly, nodes in $\{R_i\}$ are connected to a common node $\mathcal{R}$ by means of infinite-capacity and zero-cost links $E_R = \{\mathcal{R}R_i\}$.

Algorithm 1: *LP Program*

$\min \sum_e C(e)f(e)$
$f(e) \leq a \quad \forall e \in E - S_E - R_E$
$f(e) = a \quad \forall e \in L$
Each conceptual flow f corresponding to $e_i = u_i v_i$, which satisfies the constraints:
$f^i(e) \leq f(e) \quad \forall e \in E - e_i$
$f^i(e) \geq 0 \quad \forall e \in E$
$f^i_{in}(S) = 0$
$f^i_{out}(R) = 0$
$f^i_{in}(u) = f^i_{out}(u) \quad \forall u \in V - \{\mathcal{S},\ \mathcal{R},\ u_i,\ v_i\}$
$a \leq f^i_{in}(u_i) \leq 2a$
$a \leq f^i_{out}(v_i) \leq 2a$
$f^i_{in}(u_i) + f^i_{out}(v_i) \geq 4a$

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 LP problem. 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 1.6. This simple scheme utilizes each edge of the graph exactly once per time slot and thus has the minimum total bandwidth cost.

In the example shown in Figure 1.5, as opposed to the multicast scenario wherein only 4 links could be strongly covered, we see in that with network coding employed, all links are in a strong cover and we need only use each link

once per time slot for the estimation of the link probability.

1.2.2 Identifiability theorem

Consider 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 Figure 1.6, 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 (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 a loss rate α_{AC}, of the associated path.

The following theorem [2] 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.

Definition 1 *A link $e \in E$ is said to be* identifiable *if it is possible to estimate the associated loss rate of the link by sending probing packets from nodes in $\mathcal{S}$ to nodes in $\mathcal{R}$.*

Figure 1.6 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.

Theorem 1 *Given $G = (V, E)$ and sets $\mathcal{S}$ and $\mathcal{R}$, a link CD is identifiable $\Longleftrightarrow$ 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: A link CD is identifiable if C is a source or a branching point, and D is a receiver or a branching point, otherwise the link loss rate of edge CD will be undistinguishable from the loss rate of an ascendent or a descendant edge. These are the structures depicted in 1.6, 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 linkloss rate associated with link CD cannot be distinguished from the link lossrate associated with link e.

■

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 α-rate 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].

For the configurations in Figure 1.6, the links that are identifiable when network coding is employed and when only multicast is used are listed in the Table 1.1.

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 1.1: Links that are identifiable in the 4 basic configurations when network coding and simply multicast is used.

1.3 Problem motivation

Simply stated, our objective is to estimate the link probabilities of a desired set $L \subseteq E$ of links in any given network at minimum bandwidth cost. The three fundamental approaches to the estimation problem using end-to-end measurements are by means of using unicast traffic, multicast traffic and network coding.

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. However, the same assumption is made in all cost covering formulations in the literature, wherein the cost only reflects the bandwidth consumed by the probe packets, and does not take into account how the estimation accuracy (of α) is affected by the placement of sources and receivers.

Looking at the 4 configurations in Figure 1.6, we observe that the choice of sources and receivers impacts the accuracy of the estimator; that is, for a fixed number of probes, each configuration leads to a different estimation accuracy; equivalently, to achieve the same mean squared error (MSE), we may need to use a different number of probes for each configuration.

This was our motivation behind the problem that is the subject of this thesis, and that is precisely formulated in Section 2.4. Clearly, the dependence

of estimation accuracy on the number of sources and receivers would extend 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. Therefore, in the forthcoming chapters, we will quantify these various dependencies of the estimation accuracy.

2

Link probability estimation

As described in the previous chapter, given a network represented as a directed graph $G = (V, E)$, our purpose is to estimate the link probabilities of some set of links $L \in E$. Estimates of links can be obtained by maximising the likelihood function associated with the observation space of the network being studied. We find that this is a reasonable approach only for very small networks (we have worked out the likelihood formulae for various networks having up to 9 links). Some basic definitions in connection with the networks we estimate are given in Section 2.1. In Section 2.2, we formulate the estimation problem as a maximum likelihood equation. In Section 2.3, we define metrics by which to obtain performance measures in order to compare different topologies and configurations; this will be illustrated by studying two simple 3-link networks. Section 2.4 defines the research problem of this thesis, and finally the contributions made in this project are listed in Section 2.5.

2.1 Model

We study inference of links in networks that can be represented as a tree[1] $G = (V, E)$. Moreover, as elaborated in Section 1.1, it must be a logical tree as required by the identifiability theorem. All nodes in the logical tree have at least two children, apart from the source (one child) and the receivers (no children). All the links are identifiable, because there is a one-to-one mapping between the measurements and the set of link probabilities, α, of the logical tree.

We distinguish between three types of nodes in the graph (see Figure 5.3).

- Multicasting nodes : nodes that have one parent and two or more children. They merely forward the incoming packet as is, to all their children.
- Coding nodes : nodes that have 2 or more parents and 1 child. They linearly combine (according to the coding scheme and field size) incoming packets before forwarding the new packet to its child.

[1] In Chapter 8, we study networks that contain cycles. A cycle, sometimes also called a circuit, is a subset of the edge set of that forms a path such that the first node of the path corresponds to the last.

- Multicasting cum coding nodes : Nodes that are a combination of the 2 types above. They have 2 or more parents and 2 or more children. They linearly combine incoming packets and then simultaneously send (multicast) the new packet to all their children.

The parent(s) of a node k is (are) denoted by $f(k)$ and its child(or children) is (are) denoted by $d(k)$. Similarly, the origin node of a link e is $f(e)$ and the destination node is $d(e)$.

2.2 Maximum Likelihood Estimation

The observation space is denoted by Ω and every outcome $x \in \Omega$, is observed with a frequency denoted by $n(x)$. The outcome observed at some time instant m is x^m. If the sample size (number of trials in the experiment) is 'n', then the probability of n independent observations parameterised by α is

$$p(x^1, \ldots, x^n; \alpha) = \prod_{m=1}^{n} p(x^m; \alpha) = \prod_{x \in \Omega} p(x; \alpha)^{n(x)} \tag{2.1}$$

Our task is to estimate the value of α from a set of experimental data $(n(x))_{x \in \Omega}$. We do this by finding the value $\check{\alpha}$ that maximises the log-likelihood function

$$L(\alpha) = \log\, p(x^1, \ldots, x^n; \alpha) = \sum_{x \in \Omega} n(x) \log\, p(x; \alpha) \tag{2.2}$$

for the data $x^1, \ldots, x^n$. The log likelihood function for basic configuration 1 is plotted in Figure 2.1 as an example. We see that it is a well behaved function (in the $[0, 1]$ interval), that is it does not have any local maxima and minima . The same holds true for the other configurations as well.

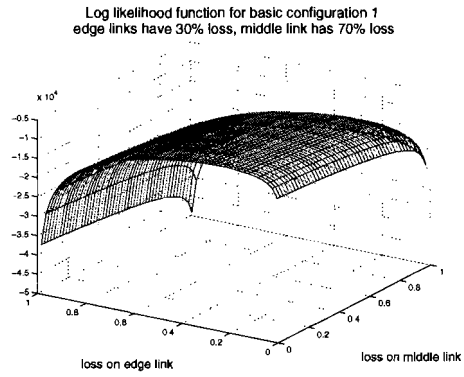


Figure 2.1: Log likelihood function for basic configuration 1.

Now, let the observed event counts be contained in the vector $\vec{O} = (n_1, n_2, \ldots n_K)$ where K is the number of events and n_1 is the number of times event 1 occur and so on. The log-likelihood function is then

$$L(\alpha) = \log(p_1^{n_1}\ p_1^{n_2} \ldots p_1^{n_K}) \tag{2.3}$$

Finally, we attempt to find an analytical solution to

$$\breve{\alpha} = \text{argmax}_{\alpha} L(\alpha) \tag{2.4}$$

This will be done for the two 3-link networks shown in Figure 2.2.

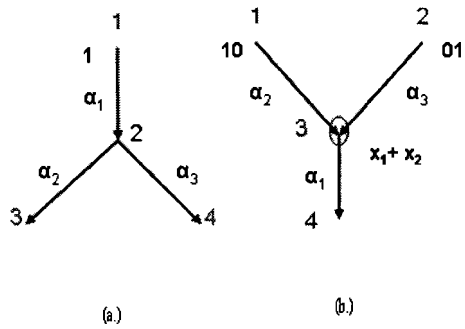


Figure 2.2: 2-leaf topology with multicasting (a) and network coding (b)

Example 2 *The observation space, underlying link loss patterns and event probabilities are shown in Tables 2.1 and 2.2. A link state is '1' if it functions, a '0' if it fails and '-' represents the don't care condition.*

Note that the event spaces are identical for these two configurations[2]*. Hence, their log-likelihood functions are the same as well, and the estimator would perform identically for corresponding links in the two cases.*

Observation		Link loss pattern			Event probability
R_1	R_2	link 1	link 2	link 3	
1	0	1	0	1	$p_1 = \alpha_1 * (1 - \alpha_2) * \alpha_3$
0	1	1	0	1	$p_2 = \alpha_1 * \alpha_2 * (1 - \alpha_3)$
1	1	1	1	1	$p_3 = \alpha_1 * \alpha_2 * \alpha_3$
0	0	1	0	0	$p_4 = \bar{\alpha_1} + \alpha_1 * \bar{\alpha_2} * \bar{\alpha_3}$
		0	-	-	

Table 2.1: Binary 2-leaf tree with multicasting. R_1 is node 3 and R_2 is node 4.

Multicast case

Consider the multicasting case, i.e.configuration (a). Let us denote by

- γ_2, the fraction of events wherein packet 1 is received by at least one of the receiver nodes 3 or 4 (i.e. by the receiver set of node 2).
- γ_3, the fraction of events wherein packet 1 reaches receiver node 3.
- γ_4, the fraction of events wherein packet 1 reaches receiver node 4.

[2]This is a consequence of Theorem 2 (reversibility) which was discussed in Section 3.1

Observation	Link loss pattern			Event probability
R	link 1	link 2	link 3	
10	1	0	1	$p_1 = \alpha_1 * (1-\alpha_2) * \alpha_3$
01	1	0	1	$p_2 = \alpha_1 * \alpha_2 * (1-\alpha_3)$
11	1	1	1	$p_3 = \alpha_1 * \alpha_2 * \alpha_3$
00	1	0	0	$p_4 = \bar{\alpha_1} + \alpha_1 * \bar{\alpha_2} * \bar{\alpha_3}$
	0	-	-	

Table 2.2: Binary 2-leaf tree with network coding. Nodes 1 and 2 are sources, node 4 is the receiver.

The ML solution turns out to be

$$\begin{aligned} \breve{\alpha}_1 &= \frac{(n_2+n_3)(n_1+n_3)}{n_3} \\ &= \frac{\gamma_3\gamma_4}{\gamma_3+\gamma_4-\gamma_2} \qquad (2.5) \\ \breve{\alpha}_2 &= \frac{n_3}{n_1+n_3} \\ &= \frac{\gamma_3+\gamma_4-\gamma_2}{\gamma_4} \qquad (2.6) \\ \breve{\alpha}_3 &= \frac{n_3}{n_2+n_3} \\ &= \frac{\gamma_3+\gamma_4-\gamma_2}{\gamma_3} \qquad (2.7) \\ & \qquad (2.8) \end{aligned}$$

The measurements that must be made at the receivers to obtain this solution are illustrated in Figure 2.3(a).

The expressions for the α_i's are intuitively obvious. For instance, the correlation between the measurements made at the 2 receivers is a measure of how lossy link 1 is. As seen from the figure, α_1 is estimated as

$$\frac{\text{(frequency with which node 3 receives a '1')(frequency with which node 4 receives a '1')}}{\text{(frequency with which node 3 and node 4 receives a '1')}}$$

Therefore, the better is this link (higher the value of α_1), the more decorrelated these measurements become, and $\gamma_3\gamma_4$ approaches $\gamma_3+\gamma_4-\gamma_2$.

Network coding case

Consider the network coding case, i.e. configuration (b.). Let us denote by

- γ_3, the fraction of events wherein any packet (10 or 01 or 11) is received by at least one of the receiver nodes 3 or 4 (i.e. by the receiver set of node 2).
- γ_3^1, the fraction of events wherein 10 reaches receiver node 4 (i.e. by the receiver set of node 3).
- γ_3^2, the fraction of events wherein 01 reaches receiver node 4 (i.e. by the receiver set of node 3).

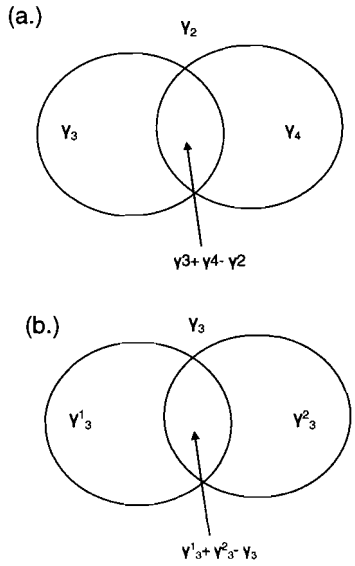


Figure 2.3: 2-leaf topology. Measurements for ML estimation (a) multicast (b) network coding

The ML solution turns out to be

$$\breve{\alpha}_1 = \frac{(n_2+n_3)(n_1+n_3)}{n_3} = \frac{\gamma_3^1\gamma_3^2}{\gamma_3^1+\gamma_3^2-\gamma_3} \quad (2.9)$$

$$\breve{\alpha}_2 = \frac{n_3}{n_1+n_3} = \frac{\gamma_3^1+\gamma_3^2-\gamma_3}{\gamma_3^2} \quad (2.10)$$

$$\breve{\alpha}_3 = \frac{n_3}{n_2+n_3} = \frac{\gamma_3^1+\gamma_3^2-\gamma_3}{\gamma_3^1} \quad (2.11)$$

(2.12)

The measurements that must be made at the receivers to obtain this solution are illustrated in Figure 2.3(b).

Again, the expressions for the α_i's are intuitively obvious. For instance, the correlation between the measurements made at the 2 receivers is a measure of how lossy link 1 is. As seen from the figure, α_1 is estimated as

$$\frac{\text{(frequency with which node 4 receives a '10')(frequency with which node 4 receives a '01')}}{\text{(frequency with which node 4 receives a '11')}}$$

Therefore, the better is this link (higher the value of α_1), the more decorrelated these measurements become, and $\gamma_3^1\gamma_3^2$ approaches $\gamma_3^1 + \gamma_3^2 - \gamma_3$. Both these cases always have a single, valid solution i.e. $\{\breve{\alpha} \in (0,1)^3\}$.

In Chapter 3, we solve the likelihood equation for the 4 basic configurations as well as some other small sized networks (4-, 5-, 7- and 9-link trees).

2.3 Performance metrics

The quality of the estimation for a single link e is captured by the *mean-squared error* metric, *i.e.*, $MSE = [|\hat{\alpha}_e - \alpha_e|^2]$, 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

$$\text{ENT} = \sum_{e \in E} \log_2 \left([\hat{\alpha}_e - \alpha_e]^2 \right), \tag{2.13}$$

which is a shifted version of the entropy of independent Gaussian random variables with the given variances [11]. 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.

In Chapter 6, we will study the performance of various estimation algorithms as a function of network size (a 45 link and a 200 link network), i.e. the number of nodes /links in the network. In order to do this, we define another metric called the 'average' ENT as

$$\text{ENT}_{av} = \log_2 \left(\frac{1}{E} \sum_{e \in E} [\hat{\alpha}_e - \alpha_e]^2 \right) \tag{2.14}$$

Under some regularity conditions (see for example Chapter 7 in [10]), the scaled (by sample size n) asymptotic covariance of the optimal estimator is lower-bounded by the Cramer-Rao bound I^{-1}, where the Fisher information matrix I is defined as

$$I_{p,q}(\alpha) = \left[\frac{\delta}{\delta \alpha_p} \log p(x^1 \ldots x^n; \alpha) \frac{\delta}{\delta \alpha_q} \log p(x^1 \ldots x^n; \alpha) \right] \tag{2.15}$$

and α_p, α_q are the loss probabilities of two links, defining a square matrix with size equal to the number of links. In particular, under the regularity conditions, the MLE is asymptotically efficient, *i.e.*, it asymptotically (in sample size) achieves this lower bound. [3] Hence the asymptotic error covariance of the MLE

[3] 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 to be so in our simulations.

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.

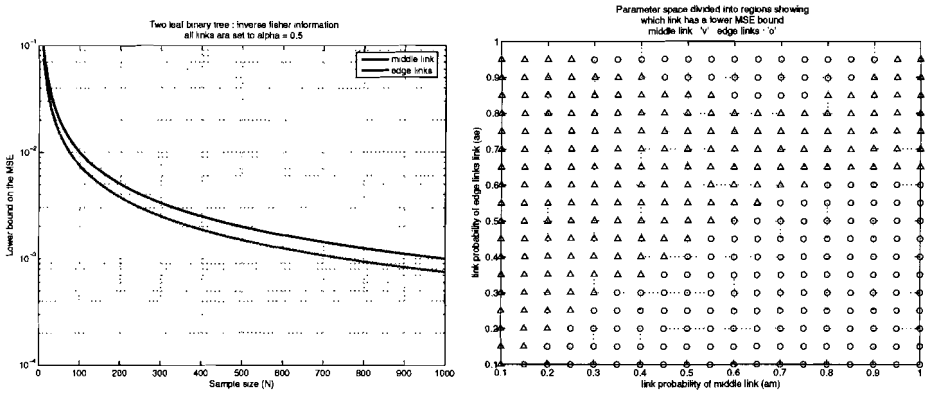
We go back to the example of the two simple 3-link configurations in Figure 2.2 and show how the Fisher matrix can be used to describe the performance of an estimator.

Example 3 *As mentioned in Example 2, the log-likelihood functions of the two networks are equal, and indeed, if we evaluate the Inverse Fisher information matrices for these two configurations, we see that they are equal to each other and is*

$$\mathcal{I}^{-1} = \frac{1}{n}\begin{pmatrix} \frac{\alpha_1(\bar{\alpha}_3-\alpha_2(1+\alpha_3(\alpha_1-2)))}{\alpha_2\alpha_3} & -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_3} & -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_2} \\ -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_3} & -\frac{\bar{\alpha}_2\alpha_2}{\alpha_1\alpha_3} & -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_1} \\ -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_2} & -\frac{\bar{\alpha}_2\bar{\alpha}_3}{\alpha_1} & -\frac{\bar{\alpha}_3\alpha_3}{\alpha_1\alpha_2} \end{pmatrix} \tag{2.16}$$

where n is the sample size.

Consider first, the 2-leaf multicast tree. All 3 links are set to have equal α. The Cramer Rao bound for the middle and edge links are plotted versus sample size in Figure 2.4(a) and in (b) we perform a 2-D sweep and plot regions where the middle link and edge links are better estimated.



(a) (b)

Figure 2.4: (a) Cramer Rao for the middle and the leaf link(s) when all three links have the same $\alpha = 0.5$ (b) Middle link set to α_m and edge links to α_e, regions where each link is better estimated are depicted.

2.4 Problem Statement

Given a network whose link probabilities must be estimated, various coding schemes can be applied, each of which, along with the algorithm chosen for the estimator, constitutes the solution to the estimation problem. Given a network

with $|V_l|$ leaves (degree 1 nodes), the number ρ of coding schemes that can be applied is

$$\rho \geq \frac{1}{2} \sum_{m=1}^{|V_l|-1} \binom{|V_l|}{m} \tag{2.17}$$

The factor $\frac{1}{2}$ results from eliminating schemes that are equivalent by the reversibility theorem. Equality in the above relation holds when an additional constraint is imposed that specifies which sources reach which receivers (or equivalently, which nodes are chosen as coding nodes). Note that a multicast approach offers $|V_l|$ solutions.

The estimation problem is defined by specifying the following quantities.

1. The adjacency matrix A of an undirected graph, $G = (V, E)$, representing the network whose links are to be estimated
2. Nodes in V which act as sources (these are leaf nodes)
3. Nodes in V which act as receivers (these are leaf nodes)
4. The set of nodes $V_c \in V$ where coding is performed

All other nodes are taken to be multicast nodes. All nodes in $\{V_c \bigcup V_m\}$, i.e. all coding and multicast nodes must be of degree 3 or higher,that is, we assume the network to contain only logical links since this is a necessary condition for identifiability (see Chapter 1). Given which nodes act as sources, which nodes as receivers, and as coding nodes, there is a unique solution to the flow in the network.

2.5 Contributions of this project

Our contributions in this project include the following.

1. For a given network topology, devise various coding schemes for the estimation purpose. These could in principle range from the traditional multicast inference scheme with a single source, to schemes with as many as $L - 1$ sources, where L is the number of leaves.
2. Develop algorithm(s)/estimator(s) to estimate the link probabilities of a network, for any coding scheme that is applied to it. We have not yet derived an exact maximum likelihood estimator for the case when any arbitrary coding scheme is applied (though for specific configurations, we have done so). Therefore, we have developed a number of simplified (sub-optimal) estimators. The solution to the estimation problem is therefore a combination, both of the coding scheme employed and the heuristic used for deriving link probability estimates. We study trees of various sizes and the dependence of the estimator on the heuristic employed and the coding scheme: number and placement of sources, receivers and coding nodes.
3. Extend the procedures developed to general graphs, that is networks that contain cycles.

3

Reversibility theorem

This chapter contains initial results on network coding based inference. Section 3.1 defines dual networks, illustrated first by a motivating example, and then the reversibility theorem which states that dual networks perform identically, is proved. Section 3.2 looks at networks that are directional (loss rates along the two directions of a link are different). The chapter is concluded in Section 3.3 where some advantages offered by the network coding approach over the multicast approach are listed.

3.1 Dual Configurations and Reversibility

Example 4 *We noted in Example 3 that the 3-link multicast tree and inverse multicast tree (in Figure 2.2) have the same event space, the same likelihood function and Fisher information matrices. Also, if we look at cases 2 and 4 in Figure 1.6, whose event spaces are written put in Tales 4.2 and 4.4, we see that the event spaces and hence likelihood functions of these two configurations are equal. We say that such configurations are the 'dual' of each other. Clearly, given a configuration, its dual can be obtained by converting all sources to receivers and vice versa and converting coding nodes to multicast nodes and vice versa. The flow through each link in a configuration is thus the negative of that in the corresponding link of its dual.*

Consider a tree with $\mathcal{L}$ leave nodes, where $\mathcal{L}_1$ act as sources and the remaining $\mathcal{L}_2 = \mathcal{L} - \mathcal{L}_1$ act as receivers of probes, and a given orientation of the links. We refer as "dual configuration" to the configuration that results from reversing the orientation of all links in the network, and from having the $\mathcal{L}_1$ sources become receivers, while the $\mathcal{L}_2$ receivers act as sources. In this section, 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 Figure 1.6). In Chapter 4 , we will see in Figure 4.14(a) and (b) that these dual configurations result in the same mean square error bound. We also see in Figures 4.16 and 4.17 that the regions of superior performance (lowest inverse Fisher information) corresponding to basic configurations 2(red) and 4(pink) are symmetrical to each other. In fact, we observed that their

associated ML estimator functions coincide. We now generalize 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 an observable outcome that occurs with the same probability in G^d and vice-versa.

With every edge $e = (j, k)$ of G, we can associate a set of sources $S(j) \subset V$ that flow through this edge, and a set of receivers $R(k) \subset V$ that observe the flow through e. Our main observation is that the pair $\{S(j),\ R(k)\}$ 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 $\{R(k),\ S(j)\}$. If in G edge e fails while all other edges do not, the receivers $R(k)$ will not receive the contribution in the probe packets of the sources $S(j)$. If in G^d edge e fails while all other edges do not, the receivers $S(j)$ will not receive the contribution in the probe packets of the sources $R(k)$. 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.

We make some remarks here. If two networks are the dual of each other it implies that they necessarily have the same observation space and hence likelihood functions. However, the converse is not true i.e., it is possible that given a topology with a specific configuration, there can exist another configuration for the same topology which has the same observation space, but which is not the dual of the first. This is because the event spaces underlying the observation spaces are not the same. This is illustrated in Figure 3.1.

Writing out the events with their event probabilities for the three cases shown in the figure shows us that the likelihood functions for the first two configurations are identical, but that that of the third is not, because though it has the same observation space, the underlying event space is not the same. This means that if an experiment is performed to estimate the links of the network, then the (maximum likelihood) link estimates obtained when configuration 1 is used are the same as that when configuration 2 is used, whereas the same link estimates are not obtained from configuration 3. We write this as

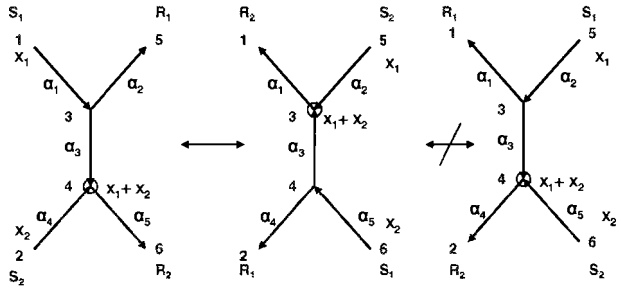


Figure 3.1: The first and second configurations are the dual of each other, but the third is not. ??.

$$\begin{aligned}(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3, \breve{\alpha}_4, \breve{\alpha}_5)_1 &= (\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3, \breve{\alpha}_4, \breve{\alpha}_5)_2 \\ &\neq (\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3, \breve{\alpha}_4, \breve{\alpha}_5)_3 \end{aligned} \tag{3.1}$$

where the subscripts denote the configuration that is used for estimation.

We illustrate this by writing out the event probabilities for the three configurations in Table 3.1. In the Figure 3.1, x_1 is the packet produced by source S_1 and x_2 by S_2. The two receivers R_1 and R_2 for the three cases are indicated as well.

Observation		configuration 1	configuration 2	configuration 3
R_1	R_2			
ϕ	x_1	$\alpha_1\ \bar{\alpha}_2\ \alpha_3\ \bar{\alpha}_4\ \alpha_5$	$\alpha_1\ \bar{\alpha}_2\ \alpha_3\ \bar{\alpha}_4\ \alpha_5$	$\bar{\alpha}_1\ \alpha_2\ \alpha_3\ \alpha_4\ \bar{\alpha}_5$
ϕ	x_2	$\alpha_4\ \alpha_5\ (\alpha_1\ \bar{\alpha}_2\ \bar{\alpha}_3 + \bar{\alpha}_1)$	$\alpha_1\ \alpha_2\ (\alpha_5\ \bar{\alpha}_3\ \bar{\alpha}_4 + \bar{\alpha}_5)$	$\alpha_4\ \alpha_5\ (\alpha_3\ \bar{\alpha}_1\ \bar{\alpha}_2 + \bar{\alpha}_3)$
ϕ	$x_1 \oplus x_2$	$\alpha_1\ \bar{\alpha}_2\ \alpha_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \alpha_3\ \bar{\alpha}_4\ \alpha_5$	$\bar{\alpha}_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5$
x_1	ϕ	$\alpha_1\ \alpha_2\ (\alpha_5\ \bar{\alpha}_3\ \bar{\alpha}_4 + \bar{\alpha}_5)$	$\alpha_4\ \alpha_5\ (\alpha_1\ \bar{\alpha}_2\ \bar{\alpha}_3 + \bar{\alpha}_1)$	$\alpha_1\ \alpha_2\ (\alpha_4\ \bar{\alpha}_3\ \bar{\alpha}_5 + \bar{\alpha}_4)$
x_1	x_1	$\alpha_1\ \alpha_2\ \alpha_3\ \bar{\alpha}_4\ \alpha_5$	$\alpha_1\ \bar{\alpha}_2\ \alpha_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \bar{\alpha}_5$
x_2	x_2	$\alpha_1\ \alpha_2\ \bar{\alpha}_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \bar{\alpha}_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \alpha_3\ \bar{\alpha}_4\ \alpha_5$
x_1	$x_1 \oplus x_2$	$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5$	$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5$
ϕ	ϕ	$1 - (\sum_{i=1}^{7} p_i)$	$1 - (\sum_{i=1}^{7} p_i)$	$1 - (\sum_{i=1}^{7} p_i)$

Table 3.1: Event space and event probabilities for the three configurations in Figure 3.1.

It must be noted here that when looking at the observation spaces of configurations 1 and 2, for every observation in the observation space of configuration 1, there is a unique observation in the space of configuration 2 with the same event probability. These two events represent the functionality of the same set of paths in the two configurations. For example, the second event in case 1 occurs with the same probability as the fourth event in case 2. Both require links 4 and 5 to function (marked in red in the figure) and the other three links to collectively fail. The second event in case 3 requires links 4 and 5 to function and the other three links to collectively fail; however, this event probability is not equal to that corresponding to cases 1 and 2 (obviously because the failure pattern on the other three links is different).

Again, the inverse Fisher information matrices of dual configurations are equal whereas this is not true of cases 1 or 2 and case 3. There would be a correspondence between the diagonal elements of the 2 matrices (of cases 1 and 3 for instance) only if the apriori link probabilities were symmetrically chosen. In the figure, this would be the case if the following mapping were applied : $\alpha_1 \rightarrow \alpha_2$ and $\alpha_4 \rightarrow \alpha_5$.

As mentioned in Chapter 2, we have not yet been able to derive a heuristic for ML estimation of arbitrary coding schemes, but have developed several sub-optimal methods instead. Now whereas an ML estimator yields identical performance for a configuration and its dual (when link probabilities are symmetrical of course), this would not be the case for sub-optimal estimation in general. We will study three suboptimal methods in Chapter 5, the first two methods for which the above applies, but the third method called the message passing algorithm, operates identically on networks that have equal event spaces, and hence, yields identical performance for any network configuration and its dual, even for an arbitrary coding scheme applied.

3.2 Measuring Directional Networks:

We examine the case where the loss rates are different in the two directions of a link. The 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 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 and one on its dual.

■

This 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.

3.3 Advantages of network coding based tomography

From our study of multicast inference, the subsequent formulation of the network coding based inference problem and our initial results as regards identifiability and reversibility, we conclude that network coding offers several benefits when compared against multicasting.

1. Minimum cost cover : While this is an NP hard problem when a multicast solution is sought, it reduces to a problem that can be cast as a linear program which is solved in polynomial time when network coding is employed.

2. Identifiability : Given a network that contains only logical links, network coding based inference always results in the maximum possible number of links in the graph being included in a strong cover.

   This is definitely not the case when multicast based inference is used. We showed by means of the example network in Figure 1.5 that all links were identifiable by means of network coding based inference, but this was not the case when multicast was used.

3. Measuring directional networks with a network coding approach allows an $\mathcal{L}$-fold decrease in measurement bandwidth compared to the multicast approach.

4

Network coding based inference

As an initial analysis, we address the problem of estimating a single link in the network in Section 4.1. In connection with this, we study four tomographic schemes applied to a 5-link topology, resulting in what we call the 'basic configurations'. Along with this, we also study some small sized networks here (ranging from 5 to 9 links). Based on our study of these topologies, we make some observations and draw some conclusions as regards the estimation problem given an arbitrary coding scheme in Section 4.2. In Section 4.3, we state some properties of the MLE and its convergence behaviour and then use this for a performance evaluation of some small networks, including the basic configurations. Hence, our purpose is twofold: (i) to compare the basic configurations against each other and (ii) make an initial demonstration of how network monitoring depends on your points of view : how a choice of coding scheme applied to the same topology can entail different solutions, in terms of their performance and analysis. Finally, in Section 4.4, we study the convergence of the MLE to the true value for any tree with an arbitrary coding scheme applied to it.

4.1 Estimation problem: basic configurations and other small networks

From the identifiability conditions that were established in Chapter 1, a link 'e' within the network which has a degree 'm' origin node $f(e)$ and a degree 'n' destination node $d(e)$ can be estimated by the general configuration shown in Figure 4.1.

The identifiability theorem requires the node degrees m and n to be at least 3 (unless $f(e) \in S$ or $d(e) \in R$. The curved lines denote paths from (to) the source subset of $f(e)$ (the receiver subset of $d(e)$).

Clearly, link e can be identified for all configurations wherein

- The number of incident incoming links of node $f(e)$ lies in $\{1, m-1\}$, the remaining being incident outgoing links.

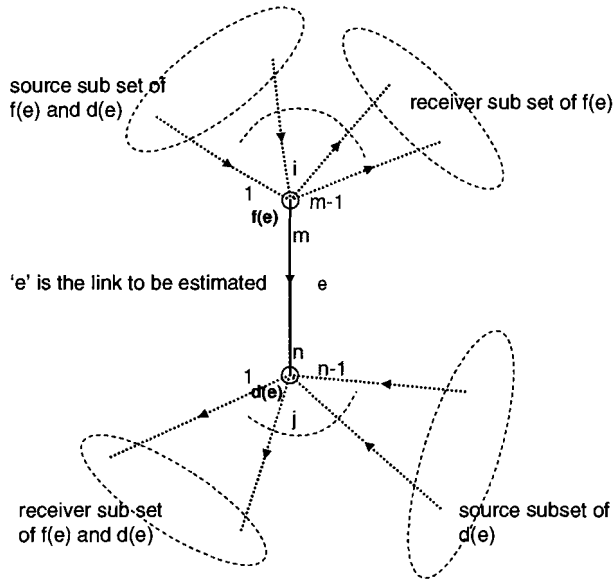


Figure 4.1: Estimating a single link 'e' in the network. The source and receiver subsets of the origin node ($f(e)$) and destination node ($d(e)$) are depicted .

- The number of incident outgoing links of node $d(e)$ lies in $\{1, n-1\}$, the remaining being incident incoming links.

Whether each of the nodes $f(e)$ and $d(e)$ are coding nodes or multicast nodes depends on the choice of configuration. Our aim is to characterise the performance of these various configurations in terms of the quality of estimation of the link probability of 'e'. Toward this end, we consider the simplest instance of this scenario, wherein the node degrees m and n are both set to 2. This results in 4 'basic configurations', shown in Figure 1.6 in Chapter 1.

Configurations 1, 3 and 4 employ network coding techniques. 1 and 3 have one coding node each, whereas all the nodes in configuration 4, which is an instance of an inverse multicast tree, are coding nodes. Configuration 2 results from application of the classical multicast technique to the topology, and all the nodes here are of the multicast type.

Now, these 4 configurations are associated with different event spaces, which translates into different likelihood functions associated with the observations. The estimate of the link probability of the desired link(s) is obtained as the maximising argument of the likelihood function. Hence, we write out the event space, event probabilities and finally the likelihood functions for each of the 4 configurations in the Tables 4.1, 4.2, 4.3 and 4.4. Configuration 1 has 10 possible outcomes summarised in table 4.1. Configuration 2 has 8 possible outcomes summarised in 4.2. Configuration 3 has 8 possible outcomes summarised in 4.3. Configuration 4 has 8 possible outcomes summarised in 4.4.

4.1.1 Basic configuration 1

Now, let us denote by

Observation		Link loss pattern					Event probability
R_1	R_2	L1	L2	L3	L4	L5	
ϕ	x_1	1	0	1	0	1	$\alpha_1 * (1-\alpha_2) * \alpha_3 * (1-\alpha_4) * \alpha_5$
ϕ	x_2	0	1	1	0	1	$p_2 = (1-\alpha_1) * \alpha_2 * \alpha_3 * (1-\alpha_4) * \alpha_5$
ϕ	$x_1 \oplus x_2$	1	1	1	0	1	$p_3 = \alpha_1 * \alpha_2 * \alpha_3 * (1-\alpha_4) * \alpha_5$
x_1	ϕ	1	0	1	1	0	$p_4 = \alpha_1 * (1-\alpha_2) * \alpha_3 * \alpha_4 * (1-\alpha_5)$
x_1	x_1	1	0	1	1	1	$p_5 = \alpha_1 * (1-\alpha_2) * \alpha_3 * \alpha_4 * \alpha_5$
x_2	x_2	0	1	1	1	1	$p_6 = (1-\alpha_1) * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5$
x_1	$x_1 \oplus x_2$	1	1	1	1	1	$p_7 = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5$
x_2	x_2	0	1	1	1	0	$p_8 = (1-\alpha_1) * \alpha_2 * \alpha_3 * \alpha_4 * (1-\alpha_5)$
x_2	x_2	1	1	1	1	0	$p_9 = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * (1-\alpha_5)$
ϕ	ϕ	multiple possibilities					$1 - (p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9)$

Table 4.1: Basic configuration 1 : Observation space, underlying link loss patterns and event probabilities.

Observation			Link loss pattern					Event probability
R_1	R_2	R_3	L1	L2	L3	L4	L5	
ϕ	ϕ	x	1	0	1	0	1	$p_1 = \alpha_1 * (1-\alpha_2) * \alpha_3 * (1-\alpha_4) * \alpha_5$
ϕ	x	ϕ	1	0	1	1	0	$p_2 = \alpha_1 * (1-\alpha_2) * \alpha_3 * \alpha_4 * (1-\alpha_5)$
ϕ	x	x	1	0	1	1	1	$p_3 = \alpha_1 * (1-\alpha_2) * \alpha_3 * \alpha_4 * \alpha_5$
x	ϕ	ϕ	1	1	0	-	-	$p_4 = \alpha_1 * \alpha_2 * ((1-\alpha_3) + \alpha_3 * (1-\alpha_4) * (1-\alpha_5))$
			1	1	1	0	0	
x	ϕ	x	1	1	1	0	1	$p_5 = \alpha_1 * \alpha_2 * \alpha_3 * (1-\alpha_4) * \alpha_5$
x	x	ϕ	1	1	1	1	0	$p_6 = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * (1-\alpha_5)$
x	x	x	1	1	1	1	1	$p_7 = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5$
ϕ	ϕ	ϕ	multiple possibilities					$p_0 = 1 - (p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7)$

Table 4.2: Basic configuration 2 : Observation space, underlying link loss patterns and event probabilities.

Observation		Link loss pattern					Event probability
R_1	R_2	L1	L2	L3	L4	L5	
ϕ	x_1	1	0	1	0	1	$p_1 = \alpha_1\ (1-\alpha_2)\ \alpha_3\ (1-\alpha_4)\ \alpha_5$
ϕ	x_2	1	0	0	1	1	$p_2 = \alpha_4\ \alpha_5\ (\alpha_1\ (1-\alpha_2)\ (1-\alpha_3) + (1-\alpha_1))$
		0	-	-	1	1	
ϕ	$x_1 \oplus x_2$	1	0	1	1	1	$p_3 = \alpha_1\ (1-\alpha_2)\ \alpha_3\ \alpha_4\ \alpha_5$
x_1	ϕ	1	1	0	0	1	$p_4 = \alpha_1\ \alpha_2\ (\alpha_5\ (1-\alpha_3)\ (1-\alpha_4) + (1-\alpha_5))$
		1	1	-	-	0	
x_1	x_1	1	1	1	0	1	$p_5 = \alpha_1\ \alpha_2\ \alpha_3\ (1-\alpha_4)\ \alpha_5$
x_2	x_2	1	1	0	1	1	$p_6 = \alpha_1\ \alpha_2\ (1-\alpha_3)\ \alpha_4\ \alpha_5$
x_1	$x_1 \oplus x_2$	1	1	1	1	1	$p_7 = \alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5$
ϕ	ϕ	multiple possibilities					$p_0 = 1 - (p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7)$

Table 4.3: Basic configuration 3 : Observation space, underlying link loss patterns and event probabilities.

- γ_3, the fraction of events wherein any packet x_1, x_2 or $x_1 \oplus x_2$ is received by at least one of the receiver nodes 5 or 6 (i.e. by the receiver set of node 4).

Observation	Link loss pattern					Event probability
R_1	L1	L2	L3	L4	L5	
x_1	1	0	1	0	1	$p_1 = \alpha_1 * (1-\alpha_2) * \alpha_3 * (1-\alpha_4) * \alpha_5$
x_2	0	1	1	0	1	$p_2 = (1-\alpha_1) * \alpha_2 * \alpha_3 * (1-\alpha_4) * \alpha_5$
x_3	0	0	1	1	1	$p_3 = \alpha_4 * \alpha_5 * (\alpha_3 * (1-\alpha_2) * (1-\alpha_1) + (1-\alpha_3))$
	-	-	0	1	1	
$x_1 \oplus x_2$	1	1	1	0	1	$p_4 = \alpha_1 * \alpha_2 * \alpha_3 * (1-\alpha_4) * \alpha_5$
$x_1 \oplus x_3$	1	0	1	1	1	$p_5 = \alpha_1 * (1-\alpha_2) * \alpha_3 * \alpha_4 * \alpha_5$
$x_2 \oplus x_3$	0	1	1	1	1	$p_6 = (1-\alpha_1) * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5$
x	1	1	1	1	1	$p_7 = \alpha_1 * \alpha_2 * \alpha_3 * \alpha_4 * \alpha_5$
$x_1 \oplus x_2 \oplus x_3$	multiple possibilities					$p_0 = 1 - (p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7)$

Table 4.4: Basic configuration 4 : Observation space, underlying link loss patterns and event probabilities.

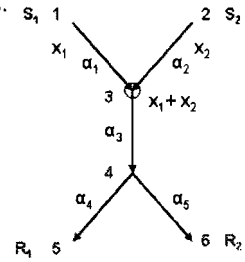


Figure 4.2: Basic configuration 1

Note that we could call this quantity as γ_4 as well, since nodes 3 and 4 have the same receiver set.

- γ_3^1(or γ_4), the fraction of events wherein x_1 reaches at least one of the receiver nodes 5 or 6.
- γ_3^2, the fraction of events wherein x_2 reaches at least one of the receiver nodes 5 or 6.
- γ_5, the fraction of events wherein any packet x_1, x_2 or $x_1 \oplus x_2$ is received by receiver node 5.
- γ_6, the fraction of events wherein any packet x_1, x_2 or $x_1 \oplus x_2$ is received by receiver node 6.

The ML solution turns out to be

$$\begin{aligned} \breve{\alpha}_1 &= \frac{n_2 + n_7 + n_9}{n_2 + n_3 + n_6 + n_7 + n_8 + n_9} \\ &= \frac{\gamma_3^1 + \gamma_3^2 - \gamma_3}{\gamma_3^2} \end{aligned} \tag{4.1}$$

$$\begin{aligned} \breve{\alpha}_2 &= \frac{n_2 + n_7 + n_9}{n_1 + n_3 + n_4 + n_5 + n_7 + n_9} \\ &= \frac{\gamma_3^1 + \gamma_3^2 - \gamma_3}{\gamma_3^1} \end{aligned} \tag{4.2}$$

The expression for $\breve{\alpha}_3$ in terms of the n_i's is rather large, hence, it is written directly in terms of our definitions above

$$\breve{\alpha}_3 = \frac{\frac{\gamma_5\gamma_6}{\gamma_5+\gamma_6-\gamma_4}}{\frac{\gamma_3}{\gamma_3^1\gamma_3^2}(\gamma_3^1+\gamma_3^2-\gamma_3)} \tag{4.3}$$

$$\begin{aligned}\breve{\alpha}_4 &= \frac{n_5+n_6+n_7}{n_1+n_2+n_3+n_5+n_6+n_7} \\ &= \frac{\gamma_5+\gamma_6-\gamma_4}{\gamma_6}\end{aligned} \tag{4.4}$$

$$\begin{aligned}\breve{\alpha}_5 &= \frac{n_5+n_6+n_7}{n_4+n_5+n_6+n_7+n_8+n_9} \\ &= \frac{\gamma_5+\gamma_6-\gamma_4}{\gamma_5}\end{aligned} \tag{4.5}$$

We have the following equalities which are satisfied

$$\gamma_4 = (1-\bar{\breve{\alpha}}_1\bar{\breve{\alpha}}_2)\breve{\alpha}_3(1-\bar{\breve{\alpha}}_4\bar{\breve{\alpha}}_5) \tag{4.6}$$

$$\gamma_4^1 = \breve{\alpha}_1\breve{\alpha}_3(1-\bar{\breve{\alpha}}_4\bar{\breve{\alpha}}_5) \tag{4.7}$$

$$\gamma_4^2 = \breve{\alpha}_2\breve{\alpha}_3(1-\bar{\breve{\alpha}}_4\bar{\breve{\alpha}}_5) \tag{4.8}$$

$$\gamma_5 = (1-\bar{\breve{\alpha}}_1\bar{\breve{\alpha}}_2)\breve{\alpha}_3\breve{\alpha}_4 \tag{4.9}$$

$$\gamma_6 = (1-\bar{\breve{\alpha}}_1\bar{\breve{\alpha}}_2)\breve{\alpha}_3\breve{\alpha}_5 \tag{4.10}$$

Hence, in the maximum likelihood estimate, the probability P(Receiver set of node 3 receives any packet) is estimated by the quantity γ_3; the probability P(Receiver set of node 3 receives x_1) is estimated by the quantity γ_3^1 and so on.

Note that, if we denote by

- γ_5^1, the fraction of events wherein receiver node 5 receives x_1
- γ_5^2, the fraction of events wherein receiver node 5 receives x_2
- γ_6^1, the fraction of events wherein receiver node 6 receives x_1
- γ_6^2, the fraction of events wherein receiver node 6 receives x_2

we find the following inequalities

$$\gamma_5^1 \neq \breve{\alpha}_1\breve{\alpha}_3\breve{\alpha}_4 \tag{4.11}$$

$$\gamma_5^2 \neq \breve{\alpha}_2\breve{\alpha}_3\breve{\alpha}_4 \tag{4.12}$$

$$\gamma_6^1 \neq \breve{\alpha}_1\breve{\alpha}_3\breve{\alpha}_5 \tag{4.13}$$

$$\gamma_6^2 \neq \breve{\alpha}_2\breve{\alpha}_3\breve{\alpha}_5 \tag{4.14}$$

Hence, in the maximum likelihood estimate, the probability P(Receiver set of node 5 receives x_1) is *not* estimated by the quantity γ_5^1 and so forth.

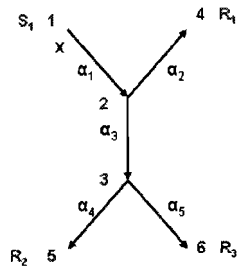


Figure 4.3: Basic configuration 2.

4.1.2 Basic configuration 2

$$\breve{\alpha}_1 = \frac{(n_4+n_5+n_6+n_7)(n_1+n_2+n_3+n_5+n_6+n_7)}{n_5+n_6+n_7} = \frac{\gamma_3\gamma_4}{\gamma_3+\gamma_4-\gamma_2} \quad (4.15)$$

$$\breve{\alpha}_2 = \frac{n_5+n_6+n_7}{n_1+n_2+n_3+n_5+n_6+n_7} = \frac{\gamma_3+\gamma_4-\gamma_2}{\gamma_3} \quad (4.16)$$

$$\breve{\alpha}_3 = \frac{(n_5+n_6+n_7)}{(n_3+n_7)} \frac{(n_2+n_3+n_6+n_7)(n_1+n_3+n_5+n_7)}{(n_4+n_5+n_6+n_7)(n_1+n_2+n_3+n_5+n_6+n_7)} = \frac{\frac{\gamma_3+\gamma_4-\gamma_2}{\gamma_3\gamma_4}}{\frac{\gamma_5+\gamma_6-\gamma_3}{\gamma_5\gamma_6}} \quad (4.17)$$

$$\breve{\alpha}_4 = \frac{n_3+n_7}{n_1+n_3+n_5+n_7} = \frac{\gamma_5+\gamma_6-\gamma_3}{\gamma_6} \quad (4.18)$$

$$\breve{\alpha}_5 = \frac{n_3+n_7}{n_2+n_3+n_6+n_7} = \frac{\gamma_5+\gamma_6-\gamma_3}{\gamma_5} \quad (4.19)$$

Hence, we have the following equalities which are satisfied

$$\gamma_2 = \breve{\alpha}_1[1-\bar{\breve{\alpha}}_2(\bar{\breve{\alpha}}_3+\breve{\alpha}_3\bar{\breve{\alpha}}_4\bar{\breve{\alpha}}_5)] \quad (4.20)$$

$$\gamma_3 = \breve{\alpha}_1(1-\bar{\breve{\alpha}}_4\bar{\breve{\alpha}}_5) \quad (4.21)$$

$$\gamma_4 = \breve{\alpha}_1\breve{\alpha}_2 \quad (4.22)$$

$$\gamma_5 = \breve{\alpha}_1\breve{\alpha}_3\breve{\alpha}_4 \quad (4.23)$$

$$\gamma_6 = \breve{\alpha}_1\breve{\alpha}_3\breve{\alpha}_5 \quad (4.24)$$

4.1.3 Basic configuration 3

In this case, it turns out to be intractable to find a closed form solution to the system of equations $\frac{\partial L(\alpha)}{\partial \alpha_i} = 0 \ , i = \{1,2,3,4,5\}$.

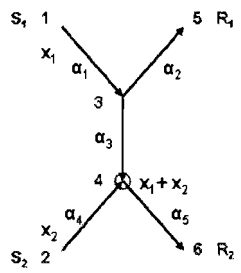


Figure 4.4: Basic configuration 3

However, the equations can be solved numerically, and we find that this is a cubic system of equations, 1 root is always valid, lying in the $(0, 1)$ interval, the other 2 roots always being invalid. The existence of a unique valid solution has not been proven rigourously, but has been verified numerically.

Now, if we denote by

- γ_5, the fraction of events wherein receiver node 5 receives x_1.
- γ_6^1, the fraction of events wherein receiver node 6 receives x_1.
- γ_6^2, the fraction of events wherein receiver node 6 receives x_2.
- γ_6, the fraction of events wherein receiver node 6 receives any packet (x_1 or x_2 or $x_1 + x_2$).
- γ_3, the fraction of events wherein the receiver set of node 3, i.e. receiver nodes 5 and 6 receive x_1.

we find the following equalities and inequalities

$$\gamma_5 = \breve{\alpha}_1\breve{\alpha}_2 \tag{4.25}$$
$$\gamma_6^2 = \breve{\alpha}_4\breve{\alpha}_5 \tag{4.26}$$
$$\gamma_6^1 \neq \breve{\alpha}_1\breve{\alpha}_3\breve{\alpha}_5 \tag{4.27}$$
$$\gamma_6 \neq \breve{\alpha}_5(1 - \bar{\breve{\alpha}}_4(1 - \breve{\alpha}_1\breve{\alpha}_3) \tag{4.28}$$
$$\gamma_3 \neq \breve{\alpha}_1(1 - \bar{\breve{\alpha}}_2(1 - \breve{\alpha}_3\breve{\alpha}_5) \tag{4.29}$$

Note that none of the γ's defined here above are maximum likelihood estimates of event probabilities which involve α_3.

Much time was spent trying to find estimates of probabilities of other events that depend on the state of link 3, but was of no avail. It is thus not clear what measurements must be made in order to derive a maximum likelihood solution to $\breve{\alpha}$.

Another method of forming an estimate (not ML but MMSE) of the link probabilities of this configuration was attempted as described here. The observed event counts are contained in the vector $\vec{O} = (n_1, n_2, \ldots n_K)$ where K is the number of events. The estimate of a particular link in the network α_e,

given the event-count vector is the expectation of the posterior

$$E[\alpha_e|\vec{O}] = \int \alpha_e P(\alpha_e|\vec{O})\, d\alpha_e \tag{4.30}$$

$$= \int_{\vec{\alpha}} \alpha_e \frac{P(\vec{O},\alpha)}{P(\vec{O})}\, d\vec{\alpha_e} \tag{4.31}$$

The joint probability of the observed event counts and the link probabilities (i.e. the likelihood function) $\vec{\alpha_e}$ is given by

$$P(\vec{O},\vec{\alpha}) = p_1^{n_1}\ p_1^{n_2} \ldots p_1^{n_K} \tag{4.32}$$

and the probability of the observation is

$$P(\vec{O}) = \int_{\vec{\alpha}} P(\vec{O},\vec{\alpha})\, d\vec{\alpha} \tag{4.33}$$

Therefore, our estimate of the link probability α_e is

$$E[\alpha_e|\vec{O}] = \int_{\vec{\alpha}} \alpha_e \frac{P(\vec{O},\vec{\alpha})}{\int_{\vec{\alpha}} P(\vec{O},\vec{\alpha})\, d\vec{\alpha}}\, d\vec{\alpha} \tag{4.34}$$

Again, finding a closed form solution to this integral analytically is not possible. It is thus this configuration which is of particular interest. Despite having the same topology, configuration 3 has a likelihood equation which has 3 roots, whereas cases 1,2 and 3 are all linear systems (with a single, valid root). Now cases 2 (multicast) and 4 (inverse multicast) are identical from an analytical perspective by the reversibility theorem. On the other hand, Case 1 can be seen as an inverse multicast tree connected by a single link (link 3) to a multicast tree, hence, we can only but expect its solution to be of the same nature (requiring measurements of the same nature) as that of cases 2 and 4.

We will study other simple topologies in the course of this chapter that employ coding schemes which are intractable to solve for analytically.

4.1.4 Basic configuration 4

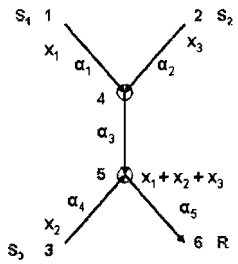


Figure 4.5: Basic configuration 4

Now let us denote by

- γ_4, the fraction of events wherein receiver node 6 receives any of the 7 possible packets.

- γ_4^1, the fraction of events wherein receiver node 6 receives a packet from source node 1.
- γ_4^2, the fraction of events wherein receiver node 6 receives a packet from source node 2.
- γ_5, the fraction of events wherein receiver node 6 receives any of the 7 possible packets.
- γ_5^3, the fraction of events wherein receiver node 6 receives a packet from source node 3.
- γ_5^4 (which is the same as $\gamma_4$in this case), the fraction of events wherein receiver node 6 receives a packet from its parent node 4.

$$\breve{\alpha}_1 = \frac{n_4+n_7}{n_2+n_4+n_6+n_7} = \frac{\gamma_4^1+\gamma_4^2-\gamma_4}{\gamma_4^2} \tag{4.35}$$

$$\breve{\alpha}_2 = \frac{n_4+n_7}{n_1+n_4+n_5+n_7} = \frac{\gamma_4^1+\gamma_4^2-\gamma_4}{\gamma_4^1} \tag{4.36}$$

$$\breve{\alpha}_3 = \frac{(n_5+n_6+n_7)}{(n_4+n_7)} \frac{(n_1+n_4+n_5+n_7)(n_2+n_4+n_6+n_7)}{(n_3+n_5+n_6+n_7)(n_1+n_2+n_4+n_5+n_6+n_7)} = \frac{\frac{\gamma_5^3+\gamma_4-\gamma_5}{\gamma_4^1+\gamma_4^2-\gamma_4}}{\frac{\gamma_4^1\gamma_4^2}{\gamma_4\gamma_5^3}} \tag{4.37}$$

$$\breve{\alpha}_4 = \frac{n_5+n_6+n_7}{n_1+n_2+n_4+n_5+n_6+n_7} = \frac{\gamma_5^3+\gamma_4-\gamma_5}{\gamma_4} \tag{4.38}$$

$$\breve{\alpha}_5 = \frac{(n_3+n_5+n_6+n_7)(n_1+n_2+n_4+n_5+n_6+n_7)}{(n_5+n_6+n_7)} = \frac{\gamma_5^3\gamma_4}{\gamma_5^3+\gamma_4-\gamma_5} \tag{4.39}$$

Hence, we have the following equalities which are satisfied

$$\gamma_4 = (1-\bar{\alpha_1}\bar{\alpha_2})\alpha_3\alpha_5 \tag{4.40}$$

$$\gamma_4^1 = \alpha_1\alpha_3\alpha_5 \tag{4.41}$$

$$\gamma_4^2 = \alpha_2\alpha_3\alpha_5 \tag{4.42}$$

$$\gamma_5 = \alpha_5(1-\bar{\alpha_4}(\bar{\alpha_3}+\alpha_3\bar{\alpha_1}\bar{\alpha_2})) \tag{4.43}$$

$$\gamma_5^3 = \alpha_4\alpha_5$$

4.1.5 Other small sized networks : 4-,5-,7- and 9-link trees

Given a 4-link topology, there are 3 different configurations that it can assume :(i) a tertiary multicast tree with 1 source and 3 receivers, (ii) a multicast cum coding tree with 2 sources and 2 receivers and (iii) a tertiary coding tree with 3 sources and 1 receiver.

We study (i) and (ii) here ((iii) is equivalent to (i) by the reversibility theorem (see Chapter 3)). See Figure 4.6.

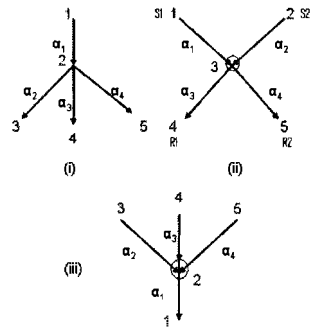


Figure 4.6: (i)multicast tree : 2 roots to the ML equation (ii) coding cum multicast tree : 5 roots to the ML equation (iii) inverse multicast (equivalent to (i))

The multicast case (i) can be solved analytically, and has 2 roots and it is verified that there is always 1 valid solution (i.e. $\breve{\alpha} \in (0,1)^3$), the other solution lying outside this interval.

It is the configuration in (ii) that is of interest; gain, no closed form solution was obtained, however, the system was solved for numerically. The ML equation has 5 roots, with 1 valid root (lying in $(0,1)^4$).

We can see what happens if a third receiver is added to topology (ii) above. See Figure 4.7.

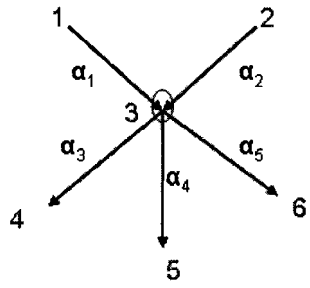


Figure 4.7: (i)multicast tree : 2 roots to the ML equation (ii) coding cum multicast tree : 7 roots to the ML equation

This system has 7 roots. It is seen that each additional link added to the 4-link case (ii), either above the center node or below it, increases the order of the system by 2.

Now we look at what would happen if an additional coding node were added to basic configuration 1 as shown in Figure 4.8.

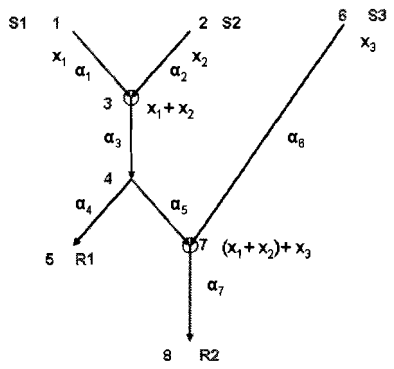


Figure 4.8: 2 links and 1 coding node added to basic configuration 1 : ML equation has 3 roots. The red path has consecutively, a coding-, multicast- and coding- node.

Again, non analytical solution was obtained; numerical evaluation shows that the system has 3 roots.

If we now add an additional multicast node to basic configuration 3, as shown in Figure 4.9, we again see that no analytical solution can be obtained, but numerical evaluation shows that the ML equation has 3 roots.

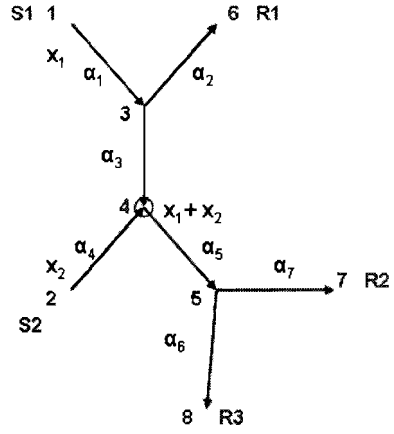


Figure 4.9: 2 links and 1 multicast node added to basic configuration 3 : ML equation has 3 roots. The red path has consecutively, a multicast-, coding-, and multicast- node.

This can be expected from the *reversibility* theorem (see Chapter 3). The configuration in Figure 4.9 is equivalent to that in Figure 4.8 since by replacing every source with a receiver and vice-versa, the 2 topologies translate into one another.

Now consider Figure 4.10. We add an additional coding node to basic configuration 3. We see cannot solve this system numerically for all possible roots, however, it is confirmed that the ML equation has more than 5 roots.

Finally, we look at a 9-link topology, see Figure 4.11. Again, we are unable to solve the ML equation for all its roots, but it can be verified that the system

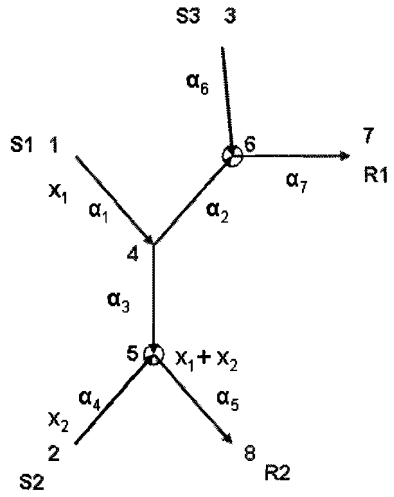


Figure 4.10: 2 links and 1 coding node added to basic configuration 3 : ML equation has more than 5 roots. The 2 red paths each has consecutively, a multicast- and coding- node.

has more than 5 roots.

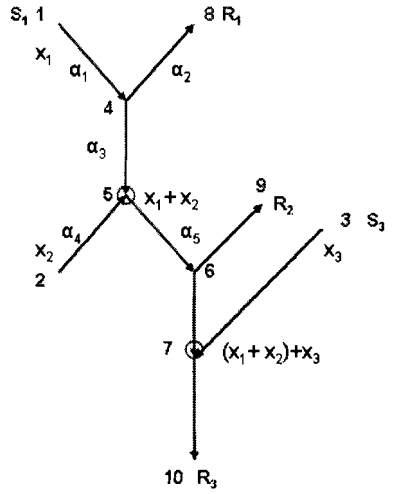


Figure 4.11: 9-link network with 3 sources, 2 receivers and 2 coding points. The red path is of the form coding node-multicast node-coding node -multicast node. The system has more than 5 roots.

4.2 Estimation problem : observations and conclusions

Based on our study of the networks in the previous section, we make some observations here that will prove useful in chapter 3 where we develop heuristics

for estimation of link probabilities of a network with any arbitrary coding scheme applied to it. We note the following.

1. A multicast tree can always be solved for analytically. A multicast node with out-degree 2 results in a linear ML system of equations. Hence, a link that is incident to 2 nodes, each with out-degree 2, would have a single solution from the ML equation.

   A multicast node with out-degree 3, results in a quadratic ML system of equations, hence a link that is incident to 2 nodes, one of which has out-degree 2 and the other, an out-degree of 3, will have 2 solutions (one of them a valid root) from the ML equation.

   A multicast node with out-degree 4 results in a cubic ML system of equations and so forth.

2. An inverse multicast tree can always be solved for analytically. The same remarks made for a multicast tree hold here by virtue of the *reversibility* theorem. A coding node with in-degree 2 results in a linear ML equation, coding nodes with in-degree of 3 results in a quadratic ML equation and so forth.

3. Nodes that are of the coding cum multicast type always result in a non-linear ML equation, for which no analytical solution has been obtained. Even the simplest node of this type : with in-degree 2 and out-degree 2 (see Figure 4.7) results in a 5^{th} order system. As mentioned before, every link added to the topology (either above or below the center node) in this figure, increases the order of the system by 2. This can be expected since the same node (Figure 4.7) performs both coding and multicast operations and is equivalent to a network that has 2 source-receiver paths, each with a multicast node and a coding node in succession. This is further elaborated upon in the following.

4. We saw that basic configurations 1, 2 and 4 are all linear systems and yielded analytical solutions. Configuration 3 on the other hand, is a cubic system, which could not be solved for analytically. Of the 3 coding schemes applied to this topology (configurations 1, 3 and 4), it is case 3 which has a general/arbitrary coding scheme and is analytically different from the others. Now case 2 (multicast) and case 4 (inverse multicast) are identical by virtue of the reversibility theorem. Case 1, is an inverse multicast tree connected by a single link to a multicast tree, and has an ML system of equations of the same nature as cases 2 and 4.

   Whereas in case 1, the coded stream is transmitted to both receivers, in case 3, the coded stream is transmitted to only one of the receivers (R_2, see Figure 4.4). The first source stream (originating at S_1) is multicast to receiver R_1 before being coded at node 4 and forwarded to R_2.

5. We saw that in basic configuration 3, we were unable to form an estimate of any event probability that involves the state of link 3. We saw that the fraction of events wherein R_2 receives x_1 from S_1 was not equal to the product of the link probabilities of the corresponding path $w = (1-3-6)$.

If we denote this fraction as $\gamma_{R_2}^{S_1}$

$$\gamma_{R_2}^{S_1} \neq \prod_{e \in w} \breve{\alpha}_e \tag{4.44}$$

We see the same thing again in figures 4.8, 4.9 and 4.10 where for the paths marked in red, the ML estimate of the probability that the path is functional is not equal to the fraction of events wherein a measurement is made at the receiver of the path.

Note: *Whenever a network contains a path 'w' (from source s to receiver r) with both a coding node and a multicast node, we have*

$$\gamma_r^s \neq \prod_{e \in w} \breve{\alpha}_e \tag{4.45}$$

In Chapter 5, we develop heuristics that find the maximum likelihood estimates of link probabilities for networks that are

- multicast trees (all nodes are of the multicast type)
- inverse multicast trees (all nodes are of the coding type)
- inverse multicast tree connected by a single link to a multicast tree. **Note**: *Here, we have both multicast and coding nodes, but that all the multicast nodes are preceded by coding nodes.*

We have not yet derived ML estimators for networks

- which contain nodes of the coding cum multicast type.
- which have an arbitrary coding scheme, i.e. wherein a path has *multicast nodes that precede coding nodes.*

6. *A path that has a multicast node with out-degree 2, followed by a coding node with in-degree 2, results in a cubic ML equation.* Hence, paths of the 3 types below, all render 3 roots to the ML equations.

   - (multicast-coding)
   - (coding-multicast-coding)
   - (multicast-coding-multicast)

   We see in Figure 4.11 that the red path is of the form (multicast-coding-multicast-coding), and that this renders at least 5 roots.

   Also, Figure 4.10 has 2 connected paths, both of the (multicast-coding) type, and this system has at least 5 roots as well.

7. In Chapter 5, where we develop heuristics for the estimation purpose, we see that

- In a multicast tree, maximum likelihood estimates of a link outgoing from some node 'k', requires measurements to be made *only* at the receiver subset of k.

  Measurements made at all receivers outside this receiver subset do not affect the estimates of the desired link. Therefore, only receivers that occur 'below' the link are relevant here,

- In an inverse multicast tree, maximum likelihood estimates of a link, that is incoming to some node 'k', requires *only* those measurements to be made at the receiver/sink that contain packets generated by sources in the source subset of k.

  Measurements involving packets generated by all other sources outside this source subset do not affect the estimate of the desired link. Therefore, only sources that occur 'above' the link are relevant here,

Hence, in multicast, inverse multicast inverse multicast-multicast type trees, only measurement of information flows that traverse through a link are relevant for estimation purposes. In short, we need only look at receivers that occur below a link and sources that occur above a link for estimating that link.

8. For a tree with arbitrary coding scheme applied, the above does not hold true. That is,
   When a path contains a multicasting node(s) preceding a coding node(s), maximum likelihood estimation of a link (k,l) in the path requires measurements to be made at receivers above the link (in addition to the receiver subset) and of sources below the link (in addition to those in the source su

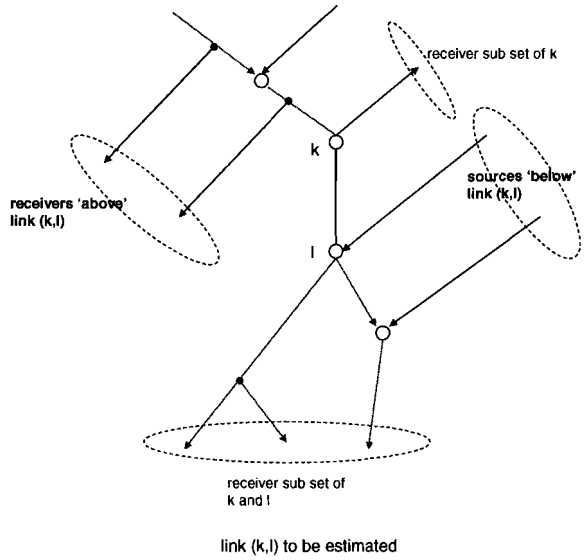
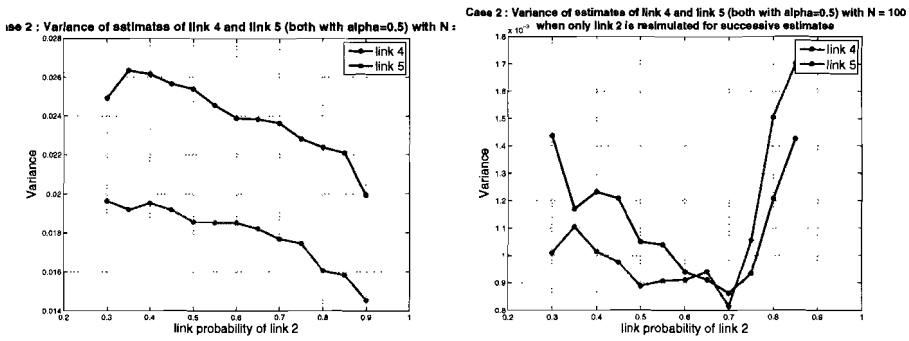


Figure 4.12: Arbitrary coding scheme wherein link (k,l) is to be estimated. Receivers above the link and sources below the link are relevant for maximum likelihood estimation.

The path in red is a 'mixed path' with both types of nodes. We will show that receivers above the link and sources below the link are relevant for maximum likelihood estimation.

We demonstrate this dependance by a simulation experiment using basic configuration 3. We do two things,

(i) Simulate for the variance of the ML estimates of the link probabilities of links 4 and 5 as a function of the link probability of link 2. Note that link 2 is a path to receiver R_1 that occurs 'above'links 4 and 5 (both of whose receiver subset is R_2). Now, if the maximum likelihood estimates of links 4 and 5 were to be independent of link 2 (which does not carry information flows that traverse either of links 4 or 5), the variance plot (see Figure 4.13(a)), would be a horizontal line, i.e., variance of these two links would be a constant, irrespective of the value on link 2.



(a) Basic case 3: Variance of links 4 and 5 as a function of the link probability of link 2.

(b) Basic case 3: Variance of links 4 and 5 as a function of the link probability of link 2, when only link 2 is re-simulated for successive estimates

Figure 4.13: Basic configuration 3 in Figure4.4 : ML estimates of links 4 and 5 requires measurements to be made at receiver R_1.

However, we see that the variance improves (decreases) as link 2 improves (α_2 increases).

(ii) Simulate for the variance of the ML estimates of the link probabilities of links 4 and 5 as a function of the link probability of link 2 (when only link 2 is re-simulated for successive estimates). This gives us some sort of measure of the sensitivity of the estimates of links 4 and 5 to the measurement made at R_1. Hence, what we see in Figure 4.13(b) is the variance of links 4 and 5 due only to the statistics of link 2. We compare the order of this against that in plot (a) and observe that the sensitivity is considerable.

Hence, finding an ML solution to the estimation problem is very complicated in a general coding scheme. As the number of alternating multicast-coding nodes in a path increases, and as the number of such 'mixed' paths in a network increases, the order of the ML system of equations grows and

as we saw earlier, even for 7- and 9- link topologies, all the roots cannot be determined.

9. For arbitrary coding schemes, it might become difficult to numerically solve the maximum likelihood equation, in which case, an iterative algorithm can be used. Often, variants of the Newton-Raphson algorithm can be used. Given a starting value $\alpha^\dagger$, we expand the ML equation, $\frac{\partial L(\breve{\alpha})}{\partial \alpha} = 0$ by a Taylor series about $\alpha^\dagger$ to obtain

$$0 \doteq \frac{\partial L(\breve{\alpha})}{\partial \breve{\alpha}} = \frac{\partial L(\alpha^\dagger)}{\partial \breve{\alpha}} + \frac{\partial^2 L(\alpha^\dagger)}{\partial \alpha \alpha^T}(\breve{\alpha} - \alpha^\dagger) \tag{4.46}$$

On rearranging, we get

$$\breve{\alpha} \doteq \alpha^\dagger + J(\alpha^\dagger)^{-1} U(\alpha^\dagger) \tag{4.47}$$

where $U(\alpha) = \frac{\partial L(\alpha)}{\partial \alpha}$ is the 'score vector' and $J(\alpha)$ is the 'observed information' matrix.

4.3 Choice of coding scheme: basic configurations and other small networks

The fundamental object controlling convergence rates of the MLE $\breve{\alpha}$ is the 'Fisher information matrix' at α. This is defined for each $\alpha \in (0,1)^{|E|}$ as the $|E|$-dimensional real matrix

$$I_{j,k}(\alpha) = \mathrm{Cov}(\frac{\partial L}{\partial \alpha_j}(\alpha), \frac{\partial L}{\partial \alpha_k}(\alpha)) \tag{4.48}$$

If the regularity conditions hold (see Section 4.4), I is equal to the more convenient expression

$$I_{j,k}(\alpha) = -E\frac{\partial^2 L}{\partial \alpha_j \partial \alpha_k} \tag{4.49}$$

We use the following theorem for all further analysis. This is proved in Section 4.4.

Theorem 1
When $\alpha_e \in (0,1]$, $e \in E$, $I(\alpha)$ is non-singular, and as $n\infty$, $\sqrt{n}(\breve{\alpha} - \alpha)$ converges in distribution to a $|E|$-dimensional gaussian random variable with mean 0 and covariance matrix $I^{-1}(\alpha)$.

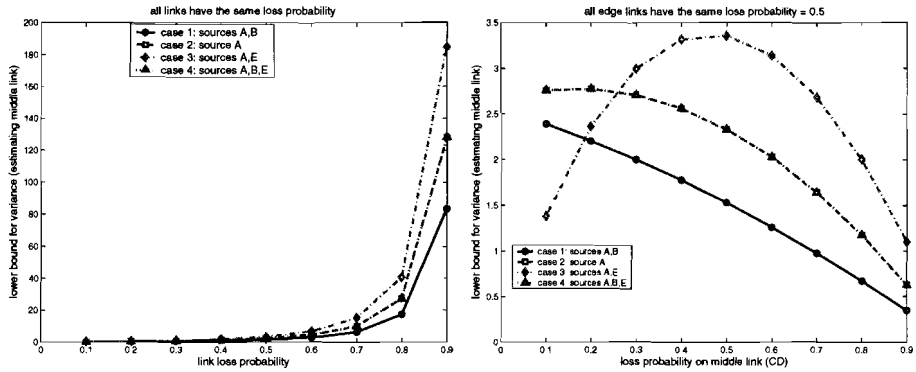
■

Hence, the inverse Fisher information matrix provides a lower bound for the mean squared error of an unbiased estimator. This is known as the 'Cramer rao bound'. This helps us in estimating the number of probes required to obtain an estimate with a desired accuracy.

Additionally, it helps us, for example, to associate confidence intervals with our estimates. For asymptotically large n, $\breve{\alpha_e}$ will lie, with probability $1-\delta$, between the points $\alpha_e \pm z_{\frac{\delta}{2}}\sqrt{\frac{I_{kk}^{-1}(\alpha)}{n}}$.

We begin by comparing the 4 basic configurations based on the Cramer Rao bound of the link estimates. We start by

i setting all links in the basic topology to have the same loss probability. The inverse Fisher information of the middle link is then calculated and plotted as a function of this loss probability in Figure 4.14a. The plot shows that configuration 1 is consistently superior. Note that the curves for cases 2 and 4 coincide (by virtue of their equivalence from the reversibility theorem). This plot can however, by no means be taken as conclusive, since it is just a 1-dimensional parameter sweep.



(a) (b)

Figure 4.14: (a) All links have the same loss probability. Cramer rao bound for the middle link plotted as a function of this. (b)All edge links have the same loss probability (0.5). Cramer rao bound for the middle link plotted as a function of its loss probability.

ii setting the 4 edge links to have a loss probability of 0.5 and then plot the Cramer-Rao bound as a function of the loss probability of the middle link in Figure 4.14b. Again, configuration 1(coding) seems to be superior over most of the region, with configuration 3(coding) being the best at a region of low loss probability of the middle link.

We now run 2 parameter sweeps

i Set all the 4 edge links in the basic topology to α_e and the middle link (to be estimated) to α_m. We sweep these 2 parameters over the $(0,1)^2$ range and then section the 2-D space into regions that are coloured according to the particular configuration that has the highest Fisher information or equivalently, the lowest lower-bound to the variance in Figure 4.15.

ii The links 1 and 2 are set to α_s , while the links 4 and 5 are set to α_r. The middle link is fixed at 4 different values and the 2-D space is sectioned as done above. See Figures 4.16 and 4.17.

From Figure 4.15, we see that when no apriori information is available as regards the quality of links from (to) the sources (receivers)incident to the link

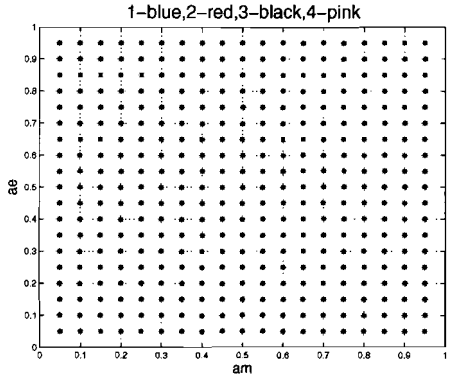
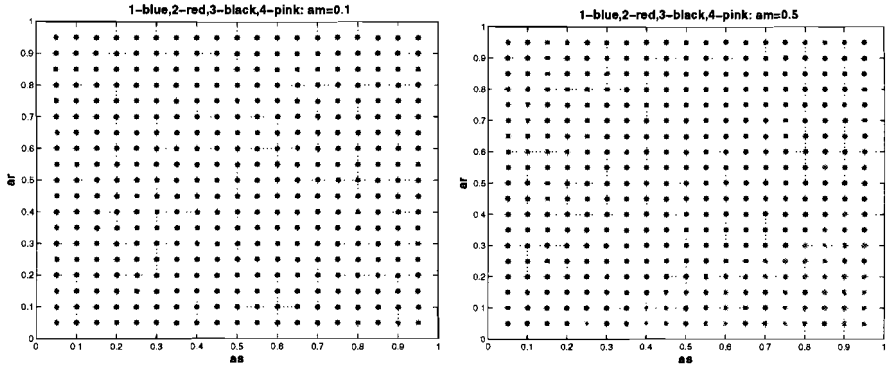


Figure 4.15: The 4 edge links are set to α_e and middle link is α_m. Configuration 1 has the lowest variance in most of the region, with configuration 3 proving to be superior for in a region of high a_m



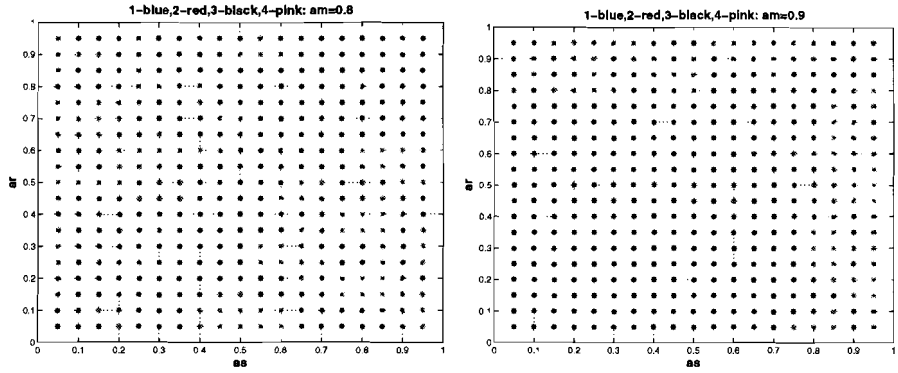
(a) (b)

Figure 4.16: Graph sectioned based on configuration with lowest Cramer Rao bound of the middle link. The links 1 and 2 are set to α_s , while the links 4 and 5 are set to α_r. The middle link is fixed at $\alpha_m = 0.1$ in (a)and $\alpha_m = 0.5$ in (b). α_s and α_r are swept over the $(0,1)^2$ range.

that is to be estimated, i.e. we set all edge links (paths) to have the same probability, then it is the two coding schemes (configurations 1 and 3) that out perform the others.

Following this, in Figure 4.16(a), configuration 1 is consistently the best estimator. This is a case with a middle link that has very poor link probability (0.1). As the middle link improves in Figure 4.16(b), the multicast and inverse multicast estimators start performing well. *Note the symmetry wrt. each other with which these configurations appear in the graph.* Further improvement of the middle link in 4.17(a), shows all 4 estimators occupying regions of the graph. Configuration 3, which did poorly up until now, appears in this plot and finally, out performs the others with a more improved middle link in Figure 4.17(b).

So far, we have studied estimator performance as regards only a single link. We now study some other networks, all of whose links we are interested in

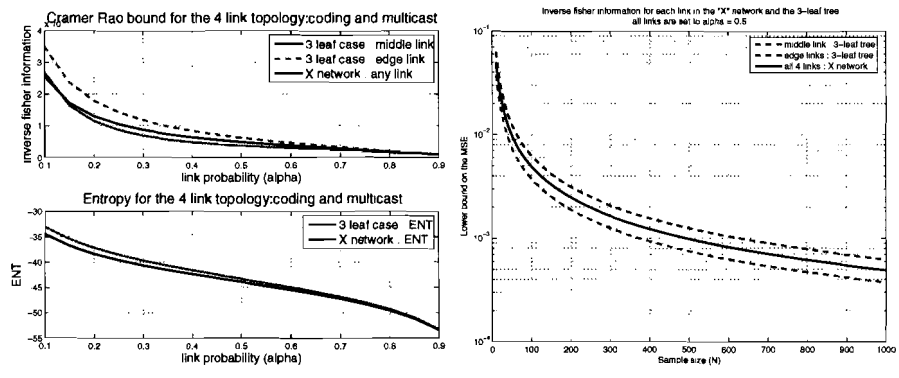


(a) (b)

Figure 4.17: Graph sectioned based on configuration with lowest Cramer Rao bound of the middle link. The links 1 and 2 are set to α_s , while the links 4 and 5 are set to α_r. The middle link is fixed at $\alpha_m = 0.8$ in (a)and $\alpha_m = 0.9$ in (b). α_s and α_r are swept over the $(0,1)^2$ range.

estimating.

Consider the 4-link configurations of Figure 4.6 : the 'X network' with a coding cum multicast node and the 3-leaf multicast tree. The X-network is symmetric in all its 4 links when each is set to the same α. We plot the Cramer Rao bound as a function of α for the middle link and edge links in the multicast tree, and for one of the 4 links in the X network in Figure 4.18(a-i). In Figure 4.18(a-ii), we plot the ENT as a function of α for these 2 configurations.



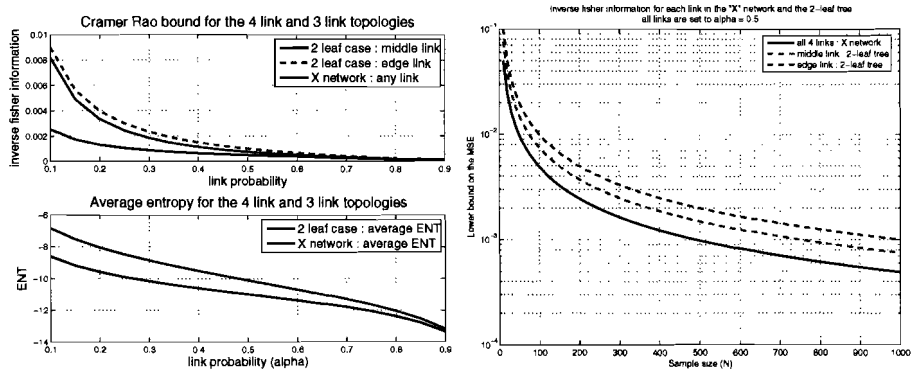
(a) (b)

Figure 4.18: (a) Cramer Rao bound for links in the X network and 3 leaf multicast tree (b) ENT for the X-network and 3-leaf multicast tree.
range.

We see that the middle link of the multicast tree is estimated with a lower variance as compared to any link in the X-network, but the edge links are estimated poorly, hence, the overall entropy (ENT) of the X-network is lower. Part (b) of the figure plots the cramer ra bound as a function of the sample size

(N).

We explained in Section 4.2 that we do not have an ML estimation procedure (heuristic) for networks that contain nodes of the coding cum multicasting type. One suboptimal method of estimation is demonstrated by means of the simple X-network. Consider Figure 4.19.



(a) (b)

Figure 4.19: (a) Cramer Rao bound for links in the X network and 2 leaf multicast tree (b) Average (per link) ENT for the X-network and 3-leaf multicast tree.

range.

We can for instance, make an estimate of α_1 in the X-network by disregarding the second source(S_2)and working with the a 3 link (2-leaf)subtree with node set $V = (1, 3, 4, 5)$. α_2 could in turn be estimated by working with the subtree with node set $V = (2, 3, 4, 5)$, α_3 by looking at $V = (1, 2, 3, 4)$ etc. We can make a measure of the sub-optimality of this approach by comparing the variance of the middle link of such a 2-leaf tree against that of a link in the X-network. This is done in the plot in (a-i) above. The solid blue line would be the variance obtained by the suboptimal method just described. The variance resulting from ML estimation is shown by the solid black line (we also plot the variance of the 2 edge links of the subtree considered in the suboptimal method, just for the sake of comparison). The plot in (a-ii) compares the per-link (average) ENT function of the X-network against that of the 3-link subtree. Part (b) of the figure plots the Cramer Rao bounds as a function of the sample size (N).

We see that even for this 4 link network, the degradation in performance is considerable when some information is disregarded and a suboptimal approach is adopted. We now study the 9-link topology shown in Figure 4.20.

We do the following

i place a single source at node 1

ii place a single source at node 3

iii place sources at nodes 1 and 3 (with the coding node at 9)

iv place sources at nodes 1 and 4 (with the coding node at 9)

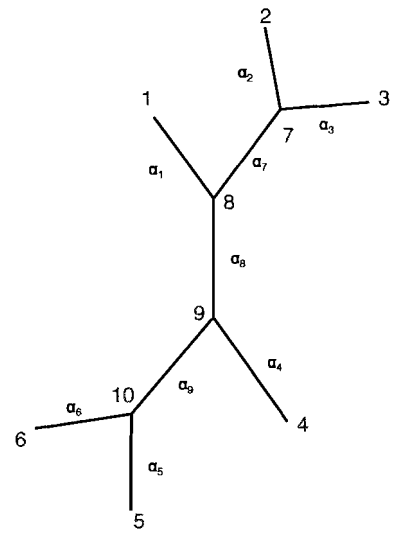


Figure 4.20: The 4 edge links are set to α_e and middle link is α_m. Configuration 1 has the lowest variance in most of the region, with configuration 3 proving to be superior for in a region of high a_m

We set all 9 links to the same link probability and then plot the entropy as a function of α in Figure 4.21.

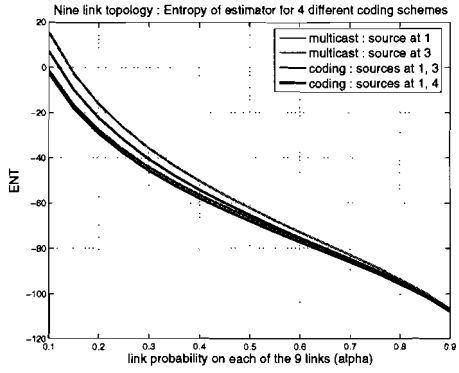
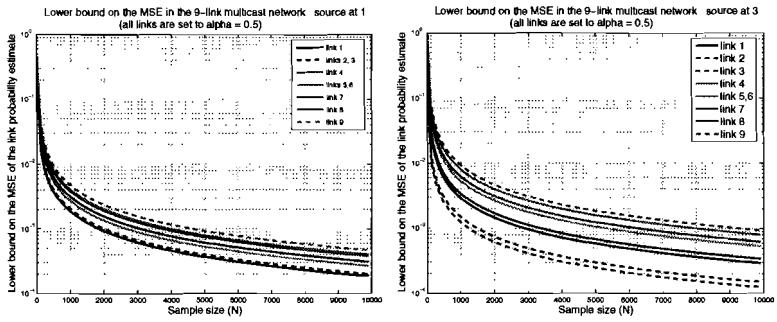


Figure 4.21: ENT for 4 different configurations of the 9 link topology.

We see that when the topology assumes the configuration of a multicast tree hung from node 3 (pink line), the far end (case ii), the performance is poorest. When multicasting is done from node 1 (red line), centrally placed (case i), performance improves very much. When two sources are placed at nodes 1 and 3 (case iii), we see that the performance (blue line) is an average of (i) and (ii). This is intuitively satisfying. Note finally that when 2 sources are placed at the symmetric nodes 1 and 4 (black line), the performance is consistently the best.

In all the networks studied so far, we see that when no apriori information about link states is available, a coding scheme that is as symmetrically applied as possible, proves to be the best estimator.

Finally, we plot the mean squared error of estimates of all 9 links as a function of N for the 2 multicast trees (cases i and ii) in Figure 4.22.



(a) (b)

Figure 4.22: MSE of each of the nine links in the 9-link topology for (a) multicast from node 1 (b) multicast from node 3

We see from these plots that *i general, links closer to the source are better estimated, and variance of the estimates increases with distance from the source.*

4.4 Convergence properties

If a certain set of conditions [10] are satisfied, it has been proven that the MLE exists and converges almost surely to the true value. Further, $I(\alpha)^{\frac{1}{2}}(\hat{\alpha} - \alpha)$ would then converge in distribution towards a standard normal distribution, as n goes to ∞. It follows that asymptotically, the distribution of $(\hat{\alpha} - \alpha)$ can be approximated by $N(0, I(\hat{\alpha})^{-1})$ or $N(0, J(\hat{\alpha})^{-1})$. This set of conditions is

1. The set α of values is compact (closed and bounded) and the true value α_0 is not on the boundary.
2. Regularity and derivatives : There exists a neighborhood B of α_0 and a constant K such that for $\alpha \in B$, and for all i, j, k, n, $\frac{1}{n}\mathbb{E}[\frac{\partial L^3(\alpha)}{\partial\alpha_i \partial\alpha_j \partial\alpha_k}] \leq K$
3. For $\alpha \in B$, the Fisher information matrix is finite in $(0,1)^{|E|}$ and has full rank (will be proven in this section)
4. For $\alpha \in B$, the interchanges of integration and differentiation in $\int \frac{\partial p(x;\alpha)}{\partial\alpha_i} dx = \frac{\partial}{\partial\alpha_i} \int p(x;\alpha) dx$ and $\int \frac{\partial^2 p(x;\alpha)}{\partial\alpha_i \partial\alpha_j} dx = \frac{\partial}{\partial\alpha_i} \int \frac{\partial p(x;\alpha)}{\partial\alpha_j} dx$ are valid.
5. Identifiability : For different values of α, the densities $p(x;\alpha)$ are different (to be proven)

We must prove the third and fifth items above: (i) non-singularity of the Fisher information matrix and (ii) identifiability, for any tree with arbitrary coding scheme applied to it.

We prove non-singularity of the Fisher matrix in the latter part of this section.

Now identifiability has been proven in [1] for a multicast tree. The proof follows from the algorithmic technique they developed there for optimal estimation. The same holds true for an inverse multicast tree (and inverse multicast-multicast tree). We are however, unable to prove identifiability for a tree with any arbitrary coding scheme applied to it.

We have verified that the mean squared error (MSE) converges to the Cramer Rao bound for (a) the 4 basic configurations (b) the X-network and (c) the 9 link network with 2 coding points in Figure 4.11.

In Figure 4.23, we set all 5 links to have a link probability = 0.7 and then look at convergence of the MLE as a function of the number of probes, for basic configurations 1 and 2. The figure shows the estimated value (for one loss realization) of α for the middle link. Both estimators converge to the true value, but from the graph, it appears that when a network coding estimator is used, less probes are required.

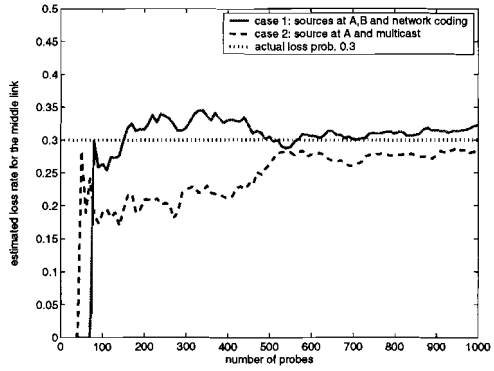
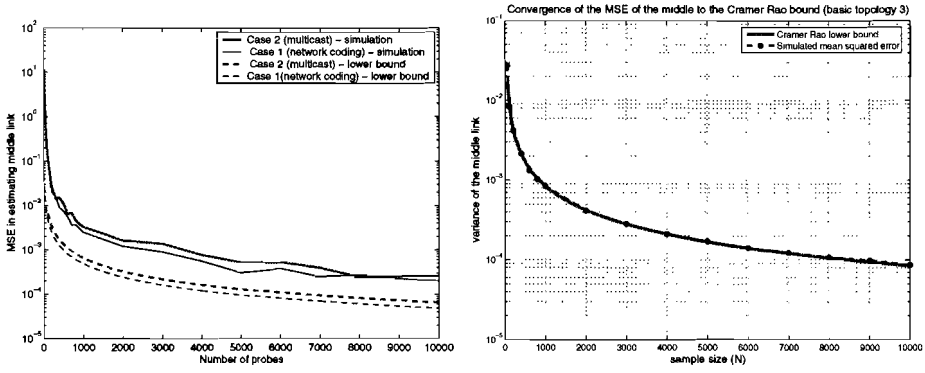


Figure 4.23: Basic configurations 1 and 2, all links are at $\alpha = 0.7$. All links are set to α. A one shot estimate of the middle link is plotted as a function of N.

Now, we simulate for the variance of the estimate of the middle link in basic configurations 1,2 and 3. All links are set at $\alpha = 0.7$. The curves for cases 2 and 4 coincide, hence the latter is omitted here. See Figure 4.24.

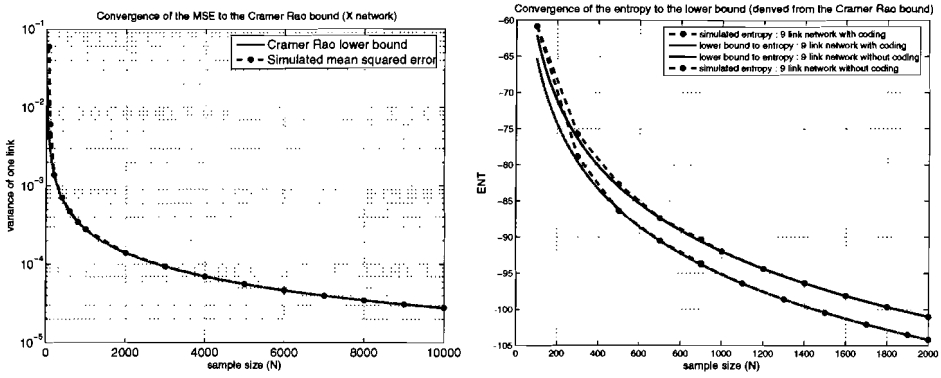
The CRB is approached in all the cases. Convergence occurs for a smaller number of probes in the case 3. We now follow the same procedure for the X-network and the 9-link topology. See Figure 4.25. In X-network, we see that the bound is achieved for N less than 100 probes. In the 9-link network, the bound is achieved between 300 and 500 probes. Note again here that the configuration wherein coding is done (red lines) is superior to when multicasting is done (blue lines).

We prove non-singularity of the Fisher information matrix corresponding to any tree with arbitrary coding scheme applied to it in Appendix A, where we first reproduce the proof for the specific cases of a multicast tree (done in [1]) and an inverse multicast tree (proof is similar to that of a multicast tree). Then, the proof for the general case i.e. for a network with an arbitrary coding scheme is done simultaneously with working out an example.



(a) (b)

Figure 4.24: All links are at $\alpha = 0.7$. Variance of middle link and cramer rao bound as a function of N.(a) Basic configurations 1 and 2 (b) Basic configuration 3



(a) (b)

Figure 4.25: All links are at $\alpha = 0.7$. Variance of middle link and cramer rao bound as a function of N.(a) X-network (b) 9-link topology with 2 coding points (see figure 4.11.)

5

Estimation algorithms

The general ML estimator is computationally challenging for large tree topologies, as explained in Chapter 2. Therefore, there is a need for low complexity, yet efficient heuristics that estimate link probabilities. A heuristic that computes maximum likelihood estimates of link probabilities of tree topologies that assume a multicast configuration is developed in [1]. We reproduce this in Section 5.1. In Section 5.2, we develop a heuristic for ML estimation of link probabilities of tree topologies that assume an inverse multicast configuration. We then also derive an ML estimator for multicast-inverse multicast trees. These estimators do not result in closed form expressions of the likelihood function. In Sections 5.4.1, 5.4.2 and 5.5, we develop heuristics that compute (sub-optimal) estimates of link probabilities for networks to which arbitrary coding schemes are applied.

5.1 ML estimator for a multicast tree

Given a network that can be represented as an acyclic graph $G = (V, E)$, estimating the link probabilities of the network can be done by letting the graph assume the configuration of a multicast tree [1] and sending probe traffic from the (single) source and making end measurements at the leaves/receivers of the network. A generic heuristic that can be applied to compute the maximum likelihood estimates of link probabilities was developed in [1] and will be reproduced here briefly.

The tree must be a logical multicast tree, i.e. all nodes excepting the leaf nodes (of degree 1) must have degree 3 or more as required by the identifiability theorem (see Chapter 1). Every node has one parent and (unless it is a receiver node), has two or more children. Some definitions follow.

There is a single source S and the set of receivers is denoted by R.

α_k is the link probability of the link ending in node k.

The parameter space to be estimated is the set of 'link probabilities' $\alpha = (\alpha_k)_{k \in V \setminus S}$ which denotes the success probabilities of all links in the graph.

The j children of k are represented by the set $d(k) = \{d_1, d_2, \ldots, d_m, \ldots, d_j\}$.

The (single) parent of k is the node $f(k)$ and the m^{th} ancestor of k is $f^m(k)$.

x is the 1xN observation vector, where N is the number of receivers. That is $x = (x_n)_{n=1\ to\ N}$ where x_n is the observation made at the n^{th} receiver. The observation made in the m^{th} trial of the experiment is x^m

Ω is the observation space. $\Omega = \{0,1\}^N$

n is the sample size and $n(x)$ is the number of outcomes wherein the observed event is x.

The probability of n independent $x^1, \ldots x^n$ is

$$p(x^1, \ldots x^n; \alpha) = \prod_{m=1}^{n} p(x^m; \alpha) = \prod_{x \in \Omega} p(x; \alpha)^{n(x)} \tag{5.1}$$

Our task is to estimate the value of α from a set of experimental data $(n(x))_{x\in\Omega}$ We will maximise the log likelihood function which is defined as

$$L(\alpha) = log\ p(x^1, \ldots x^n; \alpha) = \sum_{x \in \Omega} n(x) log\ p(x; \alpha) \tag{5.2}$$

$\Omega(k) = \{x \in \Omega : \bigcup_{n \in R(k)} x_n = 1\}$ is the set of events wherein the receiver set of k, $R(k)$ receives a packet from the source.

γ_k is the probability of occurrence of an event in $\Omega(k)$ i.e. $\gamma_k = P_\alpha[\Omega(k)]$.

$\beta_k = P[\Omega(k)|\text{a packet reaches} f(k)]$ is the probability of $R(k)$ receiving a packet given that the parent of node k, which is $f(k)$ receives a packet.

$\mu_k = P_\alpha[\text{a packet reackes node } k]$.
An estimate of γ_k is

$$\hat{\gamma}_k = \sum_{x \in \Omega(k)} \hat{p}(x) \tag{5.3}$$

where $\hat{p}(x) = \frac{n(x)}{n}$ is the observed proportion of trials with outcome x.
Now α can be calculated from $\gamma = (\gamma_k)_{k \in V}$ and the MLE $\breve{\alpha} = arg\ max_\alpha L(\alpha)$ can be calculated in the same manner from the estimates $\hat{\gamma}$.

The measured quantities γ's can be described as

$$\gamma_k = \beta_k \prod_{i=1}^{l(k)} \alpha_{f^i(k)} \tag{5.4}$$

where $l(k)$ is the distance of node k from the source.

The β's satisfy the relation

$$\bar{\beta}_k = \bar{\alpha}_k + \alpha_k \prod_{j \in (d(k))} \bar{\beta}_j, \quad k \in V \setminus R \text{ and} \tag{5.5}$$

$$\beta_k = \alpha_k, \quad k \in R \tag{5.6}$$

Now from the relations 5.4 and 5.6, we have the following equation

$$\left(1 - \frac{\gamma_k}{\mu_k}\right) = \prod_{j \in (d(k))} \left(1 - \frac{\gamma_{d_j}}{\mu_k}\right) \tag{5.7}$$

which has a unique valid solution for μ_k [1], i.e. only 1 root which lies in $(0, 1)$. Finally, the link probability α_k is calculated as

$$\alpha_k = \frac{\mu_k}{\mu_{f(k)}} \quad k \in V \setminus S \tag{5.8}$$

5.2 ML estimator for an inverse multicast tree

Now consider the topology in Figure 5.1 to which an inverse multicast configuration is applied.

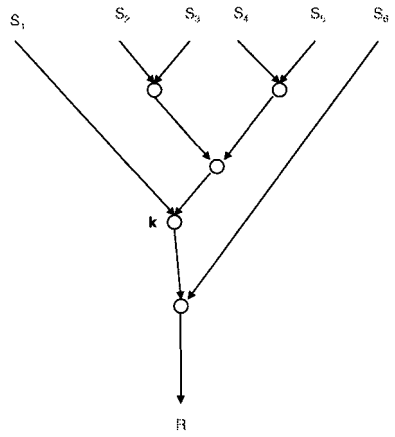


Figure 5.1: An inverse multicast tree.

The notation to be used in this section is largely the same as in the previous section, and will not be repeated, only notational differences will be given here. We will develop an algorithm for estimating some set $L \subseteq E$ of links in the network.

There is a single receiver R, all other leaf nodes serving as sources.

The set of sources is S, with M being the number of sources.

α_k is the link probability of the link *starting* in node k.

The i parents of k are represented by the set $f(k) = \{p_1, p_2, \ldots, p_m, \ldots, p_i\}$.

The source subset of node k is $S(k)$.

$(S(k))$ is the 1xM binary vector which is an indexing of the sources in $S(k)$. That is, each of the M places in the vector corresponds to one of the M sources and is a 1 whenever the corresponding source is in the source subset of k, $S(k)$.

$x = (x_n)_{n=1\ to\ M}$ where x_n is the the n^{th} bit in the packet observed at the receiver. The observation made in the m^{th} trial of the experiment is x^m

$\Omega(k) = \{x \in \Omega : \bigcup_{n \in S(k)} x_n = 1\}$ is the set of events wherein a source in $S(k)$ reaches the receiver.

γ_k is the probability of occurrence of an event in $\Omega(k)$ i.e. $\gamma_k = P_\alpha[\Omega(k)]$.

$\beta_k = P[\Omega(k)|\text{a packet reaches} f(k)]$ is the probability of the receiver receiving a packet from node k given that the parent of node k, which is $f(k)$ receives a packet.

$\mu_k^{p_m} = P[\text{any source in } S(p_m) \text{ reaches node } k]$

$\mu_k = P[\text{any source in } S(k) \text{ reaches node } k]$

$$\begin{aligned} \gamma_{p_1} &= \mu_k^{p_1} \beta_{d(k)} \\ &\cdots \\ \gamma_{p_m} &= \mu_k^{p_m} \beta_{d(k)} \\ &\cdots \\ \gamma_{p_i} &= \mu_k^{p_i} \beta_{d(k)} \\ \gamma_k &= \mu_k \beta_{d(k)} \end{aligned} \tag{5.9}$$

Note that $\beta_{d(k)}$ denotes the probability that the path (k, R) is functional. Let us call $\beta_{d(k)}$ as β for convenience.

We now use the relation

$$\mu_k = 1 - \bar{\mu}_k^{p_1} \ldots \bar{\mu}_k^{p_m} \ldots \bar{\mu}_k^{p_i} \tag{5.10}$$

to derive the following equation relating the observations (γ's) with the conditional probabilities (β's).

$$\left(1-\frac{\gamma_k}{\beta}\right) = \left(1-\frac{\gamma_{p_1}}{\beta}\right) * \ldots * \left(1-\frac{\gamma_{p_m}}{\beta}\right) * \ldots * \left(1-\frac{\gamma_{p_i}}{\beta}\right) \tag{5.11}$$

Then, Equation 5.11 is solved for β, which is the probability that the path from node k to the receiver R is functional. We can show easily [1] that there is a unique valid solution to β, i.e. only 1 root which lies in $(0,1)$. We can then compute the node probabilities as

$$\begin{aligned} \mu_k^{p_1} &= \frac{\gamma_{p_1}}{\beta} \\ &\ldots \\ \mu_k^{p_m} &= \frac{\gamma_{p_m}}{\beta} \\ &\ldots \\ \mu_k^{p_i} &= \frac{\gamma_{p_i}}{\beta} \\ \mu_k &= \frac{\gamma_k}{\beta} \end{aligned}$$

Finally, the link probability α_k is calculated as

$$\alpha_k = \frac{\mu_{d(k)}}{\mu_k} \quad k \in V \setminus R \tag{5.12}$$

Now that we have seen how to derive maximum likelihood estimates of the link probabilities of some desired set of links in multicast and inverse multicast trees by means of generic heuristics (without explicitly writing out the formula for the likelihood function), we show that the same procedures can be applied to a network as shown in Figure 5.2 for deriving ML estimates of link probabilities.

5.3 ML estimator for an inverse multicast-multicast type tree

We represent the network by the graph $G = (V, E)$, the set of coding nodes is denoted by V_c and the set of multicast nodes by V_m.

For estimation of all nodes in V_c, we treat the tree simply as an inverse multicast tree, with the full set of receivers (R_1 to R_8 in the example shown in the figure) playing the role of the single receiver R in the inverse multicast tree we studied previously.

For estimation of all nodes in V_m, we treat the tree simply as a multicast tree, with the full set of sources (S_1 to S_5 in the example shown in the figure) playing the role of the single source S in the multicast tree which we studied in Section 5.1. Hence, let us denote the full source set by (S) and the full receiver set by (R).

Therefore, the probabilities of a coding node such as j receiving a packet from either parent and all parents collectively, would be estimated according to Equation 5.11, which is reproduced here

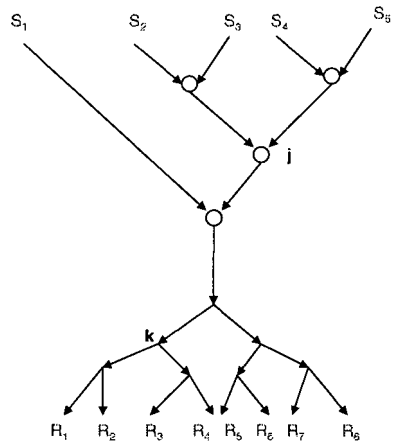


Figure 5.2: An inverse multicast-multicast tree. j is a coding node and k is a multicast node.

$$\left(1-\frac{\gamma_j}{\beta}\right) = \left(1-\frac{\gamma_{p_1}}{\beta}\right) * \ldots * \left(1-\frac{\gamma_{p_m}}{\beta}\right) * \ldots * \left(1-\frac{\gamma_{p_i}}{\beta}\right) \tag{5.13}$$

where $\gamma_j = P$[any source in the source subset of j i.e. $S(j)$ reaches the full receiver set (R)] and β is the probability that the channel from node j to (R) is functional. We solve this equation for β and obtain a unique valid solution, which we can then substitute in the equation set 5.9 to compute the estimates of probabilities of node j receiving a packet from each (or any) of its parents. These can then be used to obtain estimates of links incident to node j.

A multicast node such as k on the other hand, would be estimated according to Equation 5.7, which is also reproduced here

$$\left(1-\frac{\gamma_k}{\mu_k}\right) = \left(1-\frac{\gamma_{d_1}}{\mu_k}\right) * \ldots * \left(1-\frac{\gamma_{d_m}}{\mu_k}\right) * \ldots * \left(1-\frac{\gamma_{d_j}}{\mu_k}\right) \tag{5.14}$$

where $\gamma_k = P$[any source in the full source set (S) reaches the receiver subset of k i.e. $R(k)$] and μ_k is the probability of node k receiving some packet from (S). We solve this equation for μ_k and obtain a unique valid solution, which we can then use to obtain estimates of links incident to node k.

5.4 Estimation of a tree with an arbitrary coding configuration

In the previous two sections, we saw that estimating a link in a multicast network, inverse-multicast network or a concatenation of the two, requires only the measurement of those information flows that traverse the link. However, in Chapter 4, we proved that when a link (j, k) in a network with an arbitrary

coding scheme applied to it has to be estimated, the above does not hold true. That is, we must make measurements at receivers that occur 'above' the link, i.e. receivers that 'hang off' the path(s) $(S(j), j)$ and of sources that occur 'below' the link, i.e. sources that feed into the path(s) $(k, R(k))$. See Figure 4.12 . We also showed in Sections 4.2 in Chapter 4 why it is not yet clear how to go about developing a generic heuristic for ML estimation for such an arbitrary case. Following from this, we derive three suboptimal estimation procedures that can be used to estimate some desired set $L \subseteq E$ of links in a network with an arbitrary coding scheme. Now for any of the sub-optimal algorithms, the degree of sub-optimality would depend on a number of factors, including the number of coding nodes; the placement of these nodes in the network, i.e. their distance from source and receiver leaves; the shape of the topology; the apriori link probabilities of the network.

The following three sub-sections present the 'sub-optimal algorithm 1', the 'subtree decomposition' algorithm and the 'message passing' algorithm.

5.4.1 Suboptimal estimator 1

We consider trees to which an arbitrary coding scheme is applied. Basic definitions are illustrated using Figure 5.3.

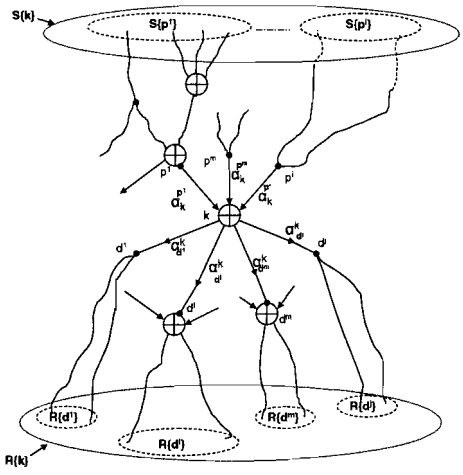


Figure 5.3: Arbitrary coding scheme.

Definitions

1. Consider a tree $G(V, E)$ representing a network as shown in the figure above. The nodes within the tree are classified as 'coding nodes' (belonging to the subset of V denoted by 'V_c') or 'multicast nodes' (belonging to the subset of V denoted by 'V_m') depending upon whether they have more than one input (the incoming packets of which are then linearly combined) or a single input, which is simply forwarded to all outgoing nodes.
   The nodes which are of the coding cum multicast type are denoted by V_{cm}.

2. The parameter to be estimated is the set of 'link probabilities' α which denotes the success probabilities of all links in the graph.
3. The total number of sources is M and the total number of receivers is N.
4. The set of receivers is $R = \{R_1, R_2, \ldots, R_N\}$.
5. The set of sources is $S = \{S_1, S_2, \ldots, S_M\}$.
6. Every node in V_m has at least 2 children.
7. A node in V_c has one child, and has more than 2 parents.
8. A node in V is denoted by $k = k(i,j)$ where i is the number of parents of k and j is the number of children of k.
9. The i parents of k are represented by the set $f(k) = \{p_1, p_2, \ldots, p_m, \ldots, p_i\}$.
10. The j children of k are represented by the set $d(k) = \{d_1, d_2, \ldots, d_m, \ldots, d_j\}$.
11. The link probability of the link ending in node $k : k \in V_m$ is α_k, whereas for $k : k \in V_c$, $\alpha_k^{p_m}$ characterises the link from the m^{th} parent of k to k.
12. Let $\alpha_{(k,R\{k\})}$ denote the 'channel probability' that at least one path from node k to its receiver set $R\{k\}$ is functional.
13. $R(k)$ is the receiver subset of node k.
14. $S(k)$ is the source subset of node k.
15. $(S(k))$ is the 1xM binary vector which is an indexing of the sources in $S(k)$. That is, each of the M places in the vector corresponds to one of the M sources and it is a 1 whenever the corresponding source is in the source subset of k, $S(k)$.
16. Ω is the observation space.
17. The 1xN observation vector

$$\begin{aligned} x &= \{x_{R1}, x_{R2} \ldots, x_{Rm}, \ldots, x_{RN}\} \\ &= (x)^N \end{aligned}$$

where each of the x_{Rm} is an 'M-dimensional packet' that traverses the network, and which is received at receiver R_m, that is

$$x_{Rm} = \{x_{Rm}(1), x_{Rm}(2), \ldots, x_{Rm}(M)\}$$

A packet arriving at some node $k \in V$ is denoted by x_k. If $k \in V_c$, then $x_k^{p_m}$ refers to the packet arriving at node k from its m^{th} parent.
The source packets are designed such that the M sources transmit the binary packets [1000...], [0100...],...,[0000...1].

18. $\Omega(k)$ will be used to denote the set of events wherein the receiver subset $R(k)$ of node k receives a packet from some source in $S(k)$.

19. γ_k will be used to denote the probability of $S(k)$ reaching $R(k)$.

20. β_k denotes the probability of $R(k)$ receiving a packet from $S(k)$ given that the parent $f(k)$ of k receives a packet from $S(k)$. In a similar fashion, $\beta_{(d(k))}$ denotes the probability of $R(k)$ receiving a packet from $S(k)$ given that node k receives a packet from $S(k)$.
Note that $\beta_{(d(k))} = \alpha_{(k,R(k))}$ is the probability that the channel from k to $R(k)$ is functional.

Now, consider as example, a node k in the network as shown in Figure 5.4.

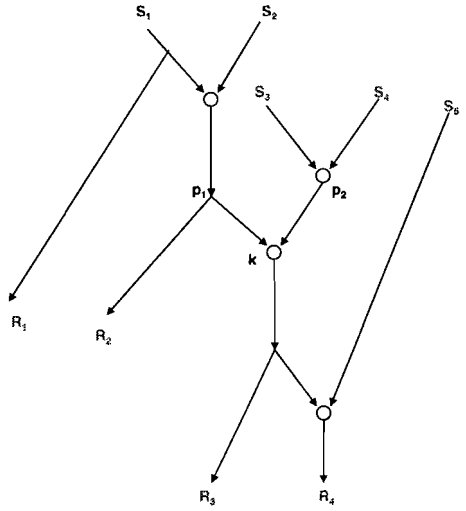


Figure 5.4: Example of a network with an arbitrary coding scheme

It has 2 parents p_1 and p_2. In general, the set of outcomes wherein at least one source in the source subset $S(p_m)$ (of some parent p_m of node k) reaches the receiver subset $R(k)$ is denoted by

$$\Omega^{p_m}(k) = \{x \in \Omega : \bigcup_{i:R_i \in R(k)} (S(p_m)) \bigcap x_{R_i} \neq \{\phi\}\}$$

The probability of occurrence of an event in $\Omega^{p_m}(k)$ is given by

$$\gamma_k^{p_m} = P_\alpha\{\Omega^{p_m}(k)\}$$

In the example shown in the figure, the probability of source subset $S(p_1)$ reaching $R(k)$ is the probability of at least one of the sources (S_1, S_2) reaching at least 1 receiver in (R_3, R_4) is denoted as $P_\alpha[\Omega^{p_1}(k)]$.

Now in general, the probability an event in $\Omega^{p_m}(k)$ occurring conditioned on the reception of at least one source in $S(p_m)$ by the parent node p_m

$$\beta_k^{p_m} = P_\alpha\{\Omega^{p_m}(k) | \text{a source in } S(p_m) \text{ reaches } p_m\}$$

The probability of at least 1 source in $S(p_m)$ reaching node k is

$$\mu_k^{p_m} = P\{\text{a source in } S(p_m) \text{ reaches k}\}$$

Now the log likelihood function is to be maximised. If the sample size is n and the observation at the m^{th} trial is denoted by x^m, then the log likelihood function is given as

$$L(\alpha) = log(x^1, x^2 \ldots x^n; \alpha) = \sum_{x \in \Omega} n(x) log\ p(x; \alpha) \tag{5.15}$$

where $n(x)$ is the number of times event x occurs.

The estimates of γ_k are obtained as

$$\hat{\gamma}_k^{p_m} = \sum_{x \in \Omega^{p_m}(k)} \hat{p}(x), \quad \text{where} \quad \hat{p}(x) = \frac{n(x)}{n} \tag{5.16}$$

Our analysis is done for a tree, i.e. an acyclic graph. The absence of cycles results in the following for any node $k(i,j) \in V$,

1.

$$S(p_k^m) \bigcap S(p_k^n) = \{\phi\} \quad \text{for all parent nodes } p_k^m \text{ and } p_k^n \textit{ and}$$
$$S(k) = S(p_k^1) + S\{p_k^2) \ldots + S(p_k^i) \tag{5.17}$$

2.

$$R(p_k^m) \bigcap R(p_k^n) = \{\phi\} \quad \text{for all parent nodes } p_k^m \text{ and } p_k^n \textit{ and}$$
$$R(k) = R(p_k^1) + R(p_k^2) \ldots + R(p_k^j) \tag{5.18}$$

Algorithm

Now we demonstrate the application of this algorithm on the network in Figure 5.5. The configuration has 1 coding node C, with all the other nodes being multicast nodes. Now we will follow the procedure outlined below.

for a coding node

We will estimate the coding node C (the probability of C receiving a packet from S_1, S_2 and collectively from either S_1 or S_2) by making the approximation shown in Figure 5.6. We do this by making measurements of S_1 and S_2 in the receiver set of node C, i.e. in subtree 3. Note that we ignore the receivers in subtrees 1 and 2.

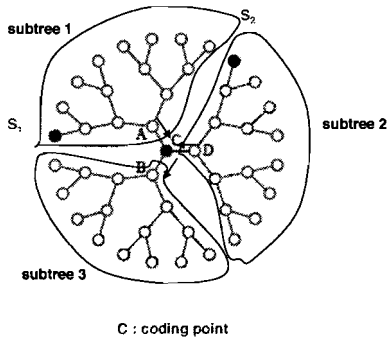


Figure 5.5: A network with an arbitrary coding scheme applied to it.

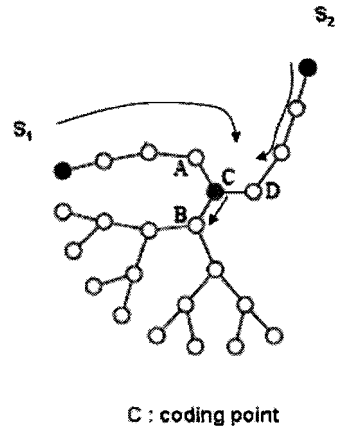


Figure 5.6: The coding node in the network shown in figure 5.5 is estimated by the approximation shown here.

The measurements are described as

$$\begin{aligned} \gamma_k^{p_1} &= \mu_k^{p_1}\beta_{d(k)} \\ &\cdots \\ \gamma_k^{p_m} &= \mu_k^{p_m}\beta_{d(k)} \\ &\cdots \\ \gamma_k^{p_t} &= \mu_k^{p_i}\beta_{d(k)} \\ \gamma_k &= \mu_k\beta_{d(k)} \end{aligned} \tag{5.19}$$

Note that $\beta_{d(k)}$ denotes the probability that the channel $(k, R(k))$ is functional.

We now use the relation

$$\mu_k = 1 - \bar{\mu}_k^{p_1} \ldots \bar{\mu}_k^{p_m} \ldots \bar{\mu}_k^{p_t} \tag{5.20}$$

to derive the following equation relating the observations (γ's) with the conditional probabilities (β's).

$$\left(1-\frac{\gamma_k}{\beta_{d(k)}}\right) = \left(1-\frac{\gamma_k^{p_1}}{\beta_{d(k)}}\right) * \ldots * \left(1-\frac{\gamma_k^{p_m}}{\beta_{d(k)}}\right) * \ldots * \left(1-\frac{\gamma_k^{p_i}}{\beta_{d(k)}}\right) \tag{5.21}$$

This equation is now solved to obtain $\beta_d(k)$. From this, the probability of the node k receiving packets from $S\{p_1\}$, $S\{p_m\}$,...,$S\{p_i\}$ are respectively

$$\begin{aligned} \mu_k^{p_1} &= \frac{\gamma_k^{p_1}}{\beta_{d(k)}} \\ &\ldots \\ &\ldots \\ \mu_k^{p_m} &= \frac{\gamma_k^{p_m}}{\beta_{d(k)}} \\ \mu_k^{p_i} &= \frac{\gamma_k^{p_i}}{\beta_{d(k)}} \end{aligned} \tag{5.22}$$

(5.23)

We thus see that when estimating the probability of a coding node,this algorithm is a better approximation algorithm when the coding node is placed closer to the periphery of the network than when it is deep within the network, because in the former case, the number of receivers that occur 'above' the coding node is likely to be fewer than that which would have been the case had the coding node been placed far away from the periphery. As we saw, this is because we ignore the measurements made at these receivers that occur above the coding node, thus making the algorithm better suited when the number of such receivers (which we ignore) is small.

for a multicast node

Again, we take as an example the configuration in figure 5.5. All nodes except C are multicast nodes. Now, the approximations made are as follows.

1. When we estimate node probabilities of a node (k) in subtree 1, we ignore measurements of S_2 in the receiver set of C, $R(C)$. We also ignore receivers outside $R(k)$.

2. When we estimate node probabilities of a node (k) in subtree 2, we ignore measurements of S_1 in the receiver set of C, $R(C)$. We also ignore receivers outside $R(k)$.

3. When we estimate node probabilities of of a node (k) in subtree 3, we ignore measurements of S_1 and S_2 in the receivers outside of the receiver set of C, $R(C)$.

The measurements are described as

$$\begin{aligned}
\gamma_k &= \mu_{f(k)}\beta_k \\
\gamma_{d_1} &= \mu_k\beta_{d_1} \\
&\dots \\
\gamma_{d_m} &= \mu_k\beta_{d_m} \\
&\dots \\
\gamma_{d_j} &= \mu_k\beta_{d_j}
\end{aligned}$$

Now, β_k is related to $\beta_{d_1}, \dots$ and β_{d_j} as

$$\bar{\beta}_k = (\bar{\alpha}_k^{p_1} * \dots * \bar{\alpha}_k^{p_i}) + (1 - \bar{\alpha}_k^{p_1} * \dots * \bar{\alpha}_k^{p_i})(\bar{\beta}_{d_1} * \dots * \bar{\beta}_{d_j}) \tag{5.25}$$

Also, $\mu_{(f(k))}$ is related to μ_k as

$$\mu_k = \alpha_k\mu_{f(k)} \tag{5.26}$$

We substitute from Equations 5.24 and 5.26 in Equation 5.24 to obtain

$$\left(1 - \frac{\gamma_k}{\mu_k}\right) = \left(1 - \frac{\gamma_{d_1}}{\mu_k}\right) * \dots * \left(1 - \frac{\gamma_{d_m}}{\mu_k}\right) * \dots * \left(1 - \frac{\gamma_{d_j}}{\mu_k}\right) \tag{5.27}$$

This is now solved for μ_k.

In the case of estimating the probability of a multicast node k, this algorithm is a better approximation when the number of sources that feed into the $(k, R(k))$ path is small. This is because, when estimating the multicast node, we ignore measurements of sources that occur 'below' the multicast node, thus making the algorithm better suited when the number of such sources is small.

Finally, when $k \in V_c$, the link probabilities of the links incoming to k are obtained as

$$\alpha_k^{p_m} = \frac{\mu_k^{p_m}}{\mu_{p_m}} \quad m = 1 \dots i \tag{5.28}$$

and the link probabilities of the links below the node k are obtained as

$$\alpha_{d_m}^k = \frac{\mu_{d_m}^k}{\mu_k} \quad m = 1 \dots j \tag{5.29}$$

In practice, we apply this algorithm to configurations wherein the multicast nodes outnumber the coding nodes. This is in general the type of coding scheme that would be applied in a practical setting as well.

Finally, we apply the suboptimal algorithm 1 to the basic configuration 3 and plot the ENT obtained in figure 5.7.

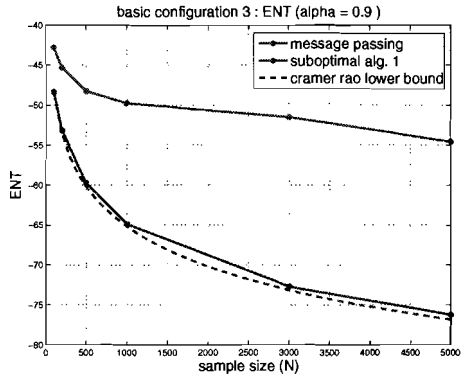


Figure 5.7: The ENT for the basic configuration 3 obtained upon application of the suboptimal algorithm 1, the message passing algorithm and the cramer rao lower bound to the ENT.

We compare the algorithm's performance against the cramer rao bound. We see that the ENT does come close to the lower bound, but does not converge to it asymptotically (even after 5000 probes). This is because the estimates resulting from this sub-optimal algorithm are biased, and remain so even asymptotically, hence, the MSE does not approach the CRB. Figure 4.24 (b) in Chapter 4 shows that the ML estimator does reach the cramer rao bound for as small a number of probes as 200.

5.4.2 Suboptimal estimator 2 : subtree decomposition

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 packers 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.

Algorithm 1:

Subtree Decomposition Algorithm

– Consider a graph G. Given a choice of sources S and receivers R, the coding points are determined and G is partitioned into $|T|$ subtrees.
– Each source sends one probe packet.Each receiver receives at most one probe packet.
– For each of the $|T|$ 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, 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 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.
  - else, deduce that $\mathcal{C}$ did not receive x_i, w.p. $p_{\mathcal{C}}$.

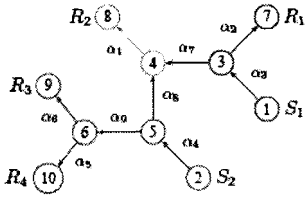


Figure 5.8: A network topology with 9 links. The link orientations depicted correspond to nodes 1 and 2 acting as sources of probes.

For example, in Fig. 5.8, 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 p, we assume that 4 did not receive a probe packet. In general, the parameters p and $p_{\mathcal{C}}$ should depend on the graph structure and prior information we may have about the link-loss rates.

The subtree method is a better approximation algorithm when the coding node(s) is located deep within the network, at a large distance from its source subset.

Now we find that the subtree method does reasonably well (compared to an optimal estimator) when the coding node(s) k is placed deep within the network, i.e. at a large distance from its source set $S(k)$. This is evident because when no measurement is made at the receiver set of k, attributing the loss to the part of the network that precedes the coding node is a better approximation when the coding node is deep within the network than when it is closer to the periphery (its source subset) since in the former scenario, the length of the path $(S(k), k)$ is larger, hence making the probability of a loss having occurred along it, more likely.

5.5 Suboptimal estimator 3 : Message passing algorithm

This approach was first developed in [13] for multicast based inference. We apply the same concepts to inference based on arbitrary network coding schemes. Consider a network represented as a graph $G = (V, E)$. E is the set of links in the network, with every link $e \in E$ being associated with a random variable X_e that represents the state of the link. The state X_e takes values in $\{0, 1\}$ and is defined as

$$X_e = \begin{cases} 1\ , & \text{link e is functional} \\ 0\ , & \text{link e is not functional} \end{cases}$$

Denote by W, the set of all paths in the graph, with every path $w \in W$ being associated with a random variable X_w that represents the state of the path. The state X_w takes values in $\{0, 1\}$ and is defined as

$$X_w = \begin{cases} 1\ , & \text{path w is functional} \\ 0\ , & \text{path w is not functional} \end{cases}$$

Every path $w \in W$ is described by the links $e \in w$ contained in that path as

$$w = (e)_{e \in w} \tag{5.30}$$

The state X_w of a path w is related to the states X_e, $\forall\ e \in w$ of the links contained in it as

$$X_w = \oplus_{e \in w} X_e \tag{5.31}$$

Also, we represent the set of states of each of the links e contained in a path w as

$$(X_e)_{e \in w}$$

Collectively, the set of random variables representing the states of all the links in the network is represented as

$$X_E = \{X_e\ :\ e \in E\} \tag{5.32}$$

and the set of random variables representing the states of all the paths in the network is represented as

$$X_W = \{X_w\ :\ w \in W\} \tag{5.33}$$

The parameter set, which is to be estimated is

$$\alpha_E = \{\alpha_e\ :\ e \in E\} \tag{5.34}$$

The problem of link loss inference in the network is then the problem of estimating α_E based on the observation of $X_W^{(1,N)}$. We will denote the sets of link and path states at time instant 'i' as $X_E^{(i)}$ and $X_W^{(i)}$. The problem is set up as, for each $e \in E$, finding the $\hat{\alpha}_e$ that maximises the posterior probability $P[\alpha_e|X_W^{(1,N)}]$ of α_e conditioned on the observation (of path states)$X_W^{(1,N)}$ (at N consecutive instants of time).

This can be done by representing the joint probability mass function (PMF) of the three random variables X_E, X_W and α_E as a factor graph. A factor graph is a bipartite graph that represents the factorisation of a multivariate probability mass function by specifying a set of conditional independence relationships between the variables. There are two types of vertices in the graph : variable vertices representing the variables of the global multivariate function and function vertices representing the factors in the factorisation. A variable vertex is connected to a function vertex if the variable is a function of the factor. The sum product algorithm [12] is a generic message passing algorithm that can be applied to the factor graph in order to compute desired marginals of the PMF. We will now show this can be used to find the required posterior mentioned above.

Let the joint PMF of $X_E^{(i)}$ parameterised by α_E be denoted by $B_E(X_E, \alpha_E)$. Upon the assumption that each link suffers from independent link loss rates, we have

$$B_E(X_E, \alpha_E) = \prod_{e \in E} B(X_e, \alpha_e) \tag{5.35}$$

where

$$B(X_e, \alpha_e) = X_e\alpha_e + (1 - X_e)(1 - \alpha_e) \tag{5.36}$$

Then the joint PMF of $\alpha_E, X_E^{(1,n)}$ and $X_W^{(1,n)}$ factors as

$$P(\alpha_E, X_E^{(1,n)}, X_W^{(1,n)}) \propto \prod_{i=1}^{n} B_E(X_E^{(i)}, \alpha_E) P_{W|E}(X_W^{(i)}, X_E^{(i)}) \tag{5.37}$$

where $P_{W|E}(X_W^{(i)}, X_E^{(i)})$ is the conditional PMF of $X_W^{(i)}$ given $X_E^{(i)}$, and in fact, if we define a constraint indicator function $\delta[.]$ as

$$\delta[C(x)] = \begin{cases} 1\,, & \text{if } C(x) \\ 0\,, & \text{otherwise} \end{cases}$$

the conditional density can be written as

$$P_{W|E}(X_W^{(i)}, X_E^{(i)}) = \prod_{w \in W} \delta[X_w^{(i)} = \oplus_{e \in w} X_e^{(i)}] \tag{5.38}$$

Then our objective becomes finding $\hat{\alpha}_e$ for each $e \in E$ that maximises

$$P(\alpha_E|X_W^{(1,n)}) \propto \sum_{\sim\alpha_e} P(\alpha_E, X_E^{(1,n)}, X_W^{(1,n)})$$
$$\propto \sum_{\sim\alpha_e} \prod_{i=1}^{n} B_E(X_E^{(i)}, \alpha_E) P_{W|E}(X_W^{(i)}, X_E^{(i)}) \quad (5.39)$$

The objective of finding for each $e \in E$ the maximising $\hat{\alpha}_e$ for the function

$$\sum_{\sim\alpha_e} \prod_{i=1}^{n} B_E(X_E^{(i)}, \alpha_E) P_{W|E}(X_W^{(i)}, X_E^{(i)}) \quad (5.40)$$

precisely coincides with the objective of the sum-product algorithm. Now, the function

$$\prod_{i=1}^{n} B_E(X_E^{(i)}, \alpha_E) P_{W|E}(X_W^{(i)}, X_E^{(i)}) \quad (5.41)$$

can be represented by a factor graph as shown in Figure 5.9. Hence, the sum-product algorithm can be used to simultaneously find $\hat{\alpha}_e$ for all $e \in E$ in parallel.

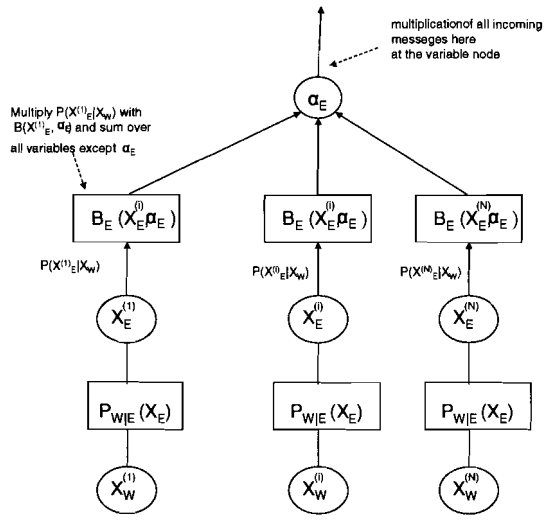


Figure 5.9: The factor graph representing the the function in Equation 5.41.

Let us further express the factors $B_E(.)$ and $P_{W|E}(.)$ according to Equations 5.35 and 5.38. Then, as an example, the factor graph of the 3-link network shown in Figure 5.10 can be expanded to a form similar to that shown in Figure 5.11.

This diagram is a representation of the factor graph in the form of layers of subgraphs. Each layer corresponds to a single trial of the experiment (which has a sample size of N) The message passing algorithm when applied to the factor

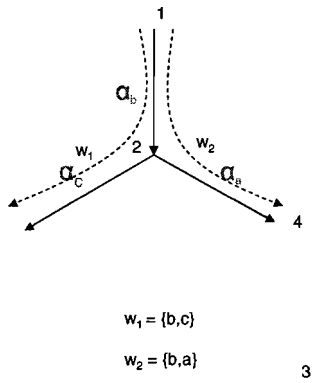


Figure 5.10: A 3 link network, the factor graph of which is shown in Figure 5.11. There 3 links a, b and c and 2 paths : $w_1 = (b, c)$ and $w_2 = (b, a)$

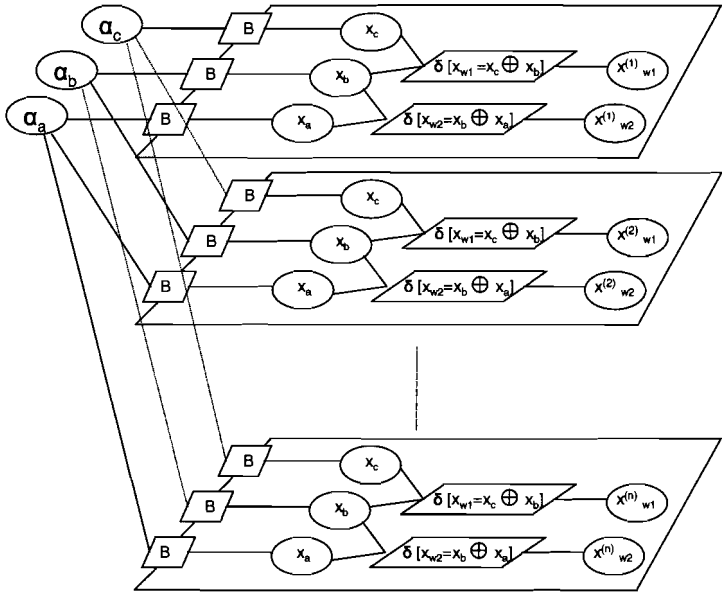


Figure 5.11: The factor graph representing the function in Equation 5.41 in its expanded form. The sample size is n.

graph therefore requires messages to be passed within each layer, called 'intra-layer' message passing, and after convergence, messages are passed between the layers, called 'inter-layer' message passing. With such a schedule, the processing of the path state at each time instant can be carried out independently.

The variable nodes corresponding to each path - $X_{\{c,b\}}$, and $X_{\{a,b\}}$ - in each layer are leaf nodes, and hence can be discarded, with the messages simply being passed between the $\delta[.]$ check nodes, which henceforth, we will refer to as X_w and the X_e variable nodes.

Each layer defines the relation between the function $P(X_W|X_E)$ and its arguments X_W and X_E.

$$P(X_W|X_E) = \prod_{w \in W} [x_w = \oplus x_e] \tag{5.42}$$

We now derive the messages computed at the check nodes and the variable nodes within each layer. Each message is defined to be the posterior probability of a link taking state 1 (the probability of the link state being zero is one minus this quantity). In the initialisation phase, each variable node X_e passes a message $\mu_{e \to w}$ representing the uniform distribution over the link state X_e to every adjacent X_w check node, i.e. $\mu_{e \to w} = \frac{1}{2}$.

Notation:

- The edge directed from variable node X_e to check node X_w is denoted as $e \to w$.
- The edge directed from check node X_w to variable node X_e is denoted as $w \to e$.
- $\mu_{e \to w}$ is the linear message passed along $e \to w$, which is the posterior probability of $X_e = 1$.
- $\mu_{w \to e}$ is the linear message passed along $w \to e$, which is the posterior probability of $X_e = 1$.

The entire algorithm is implemented in the logarithmic domain, hence the messages are converted into log-likelihood ratios (LLR's). We have

$$LLR(\mu_{w \to e}) = \gamma_{w \to e} = log\left(\frac{\mu_{w \to e}}{1 - \mu_{w \to e}}\right) \tag{5.43}$$

Similarly,

$$LLR(\mu_{e \to w}) = \gamma_{e \to w} = log\left(\frac{\mu_{e \to w}}{1 - \mu_{e \to w}}\right) \tag{5.44}$$

Computation at a check node:

Consider a path $w \in W$ that contains a link $e \in E$. We have

$$\mu_{w \to e} = P[X_e = 1 | X_w = x_w] \tag{5.45}$$

From the theory of the sum product (SP) algorithm applied on a factor graph, we know that the quantity computed by the check node X_w which is passed along edge $w \to e$ of the factor graph is

$$\begin{aligned} &\text{quantity computed by the SP} \\ &\text{algorithm along edge } w \to e = P[X_w = x_w | X_e = x_e] \end{aligned} \tag{5.46}$$

We will proceed to compute the quantity in 5.46 which will then be used to derive the required message defined in Equation 5.45. Toward this end, we can derive an expression for $P[X_w = 0|X_e = x_e]$, and then $P[X_w = 1|X_e = x_e]$ follows simply as $1 - P[X_w = 0|X_e = x_e]$. Therefore, consider

$$\begin{aligned} P[X_w = 0|X_e = x_e] &= \sum_{\{x_{e'}\}\forall\ e' \in w\setminus e} P[X_w = 0|(X_{e'} = x_{e'})_{e'\in w}]\ P[(X_{e'} = x_{e'})_{e'\in w\setminus e}] \\ &= \sum_{\sim x_e} P[X_w = 0|(X_{e'} = x_{e'})_{e'\in w}]\ P[(X_{e'} = x_{e'})_{e'\in w\setminus e}] \\ &= \sum_{\sim x_e} \delta[0 = \oplus_{e'\in w} x_{e'}] \prod_{e'\in w\setminus e} P[X_{e'} = x_{e'}] \\ &= 1 - x_e \prod_{e'\in w\setminus e} \mu_{e'} \end{aligned} \tag{5.47}$$

We have to compute $\mu_{w\to e} = P[X_e = 1|X_w = x_w]$. We have

$$P[X_e = 1|X_w = 1] = 1 \tag{5.48}$$

and

$$\begin{aligned} P[X_e = 1|X_w = 0] &= \frac{P[X_w = 0|X_e = 1]P[X_e = 1]}{P[X_w = 0]} \\ &= \frac{(1 - \prod_{e'\in w\setminus e} \mu_{e'})P[X_e = 1]}{P[X_w = 0]} \end{aligned} \tag{5.49}$$

From Equation 5.47, we have

$$\begin{aligned} P[X_w = 0] &= 1 - P[X_e = 1] \prod_{e'\in w\setminus e} \mu_{e'\to w} \\ \therefore \frac{P[X_w = 0]}{P[X_e = 1]} &= \frac{1}{P[X_e = 1]} - \prod_{e'\in w\setminus e} \mu_{e'\to w} \\ &= 2 - \prod_{e'\in w\setminus e} \mu_{e'\to w} \quad \because \text{we set } P[X_e = 1] \text{ to } \frac{1}{2} \text{ since} \\ &\qquad \text{we have no apriori information about the state of link e} \end{aligned} \tag{5.50}$$

Substituting this in Equation 5.49, we have

$$P[X_e = 1|X_w = 0] = \frac{1 - \prod_{e'\in w\setminus e} \mu_{e'\to w}}{2 - \prod_{e'\in w\setminus e} \mu_{e'\to w}}$$

$$\mu_{w\to e} = \begin{cases} 1\ , & \text{if } X_w = 1 \\ \frac{1-\prod_{e'\in w\setminus e} \mu_{e'\to w}}{2-\prod_{e'\in w\setminus e} \mu_{e'\to w}}\ , & \text{if } X_w = 0 \end{cases} \tag{5.52}$$

Converting to the LLR form,

$$\gamma_{w\to e} = \begin{cases} \infty\,, & \text{if } X_w = 1 \\ log\left(1 - \prod_{e'\in w\setminus e} \frac{e^{\gamma_{e'\to w}}}{1+e^{\gamma_{e'\to w}}}\right), & \text{if } X_w = 0 \end{cases} \tag{5.53}$$

Computation at a variable node:

The computation at a variable node is simply the summation of all incoming messages, therefore

$$\gamma_{e\to w} = \sum_{w'\in N(e)\setminus w} \gamma_{w'\to e} \tag{5.54}$$

where $N(e)$ is the set of all check nodes X_w adjacent to the variable node x_e in the factor graph.

In terms of the linear messages, this translates into

$$\mu_{e\to w} = \frac{\prod_{w'\in N(e)\setminus w} \mu_{w'\to e}}{\prod_{w'\in N(e)\setminus w} \mu_{w'\to e} + \prod_{w'\in N(e)\setminus w}(1-\mu_{w'\to e})} \tag{5.55}$$

At the end of each iteration, a summary message μ_e is computed for each variable vertex e using messages along all edges incident to it as

$$\mu_{e\to w} = \frac{\prod_{w'\in N(e)} \mu_{w'\to e}}{\prod_{w'\in N(e)} \mu_{w'\to e} + \prod_{w'\in N(e)}(1-\mu_{w'\to e})} \tag{5.56}$$

The iterative process is terminated when the summary messages converge or when a pre-set number of iterations have been completed. Now when the summary message has been computed for each of the N trials of the experiment, we apply the sum-product algorithm by passing messages between the layers of the graph, i.e. inter layer message passing. Messages are passed from the variable nodes x_e corresponding to the links in the network to the vertices α_e via the check nodes representing the $B(.)$ function.

The required posterior $P[\alpha_e|X_W^{(1,N)}]$ will be computed from the SP algorithm now. The rule for the sum-product algorithm at the check node is to multiply all incoming messages to the check node with the function represented by the check node and then sum this over all variables except the variable associated with the variable node to which the check node is sending the message.

The incoming message at the check node $B_e(.)$ is $P(X_e|X_W)$. The function associated with the check node is $B(x_e,\alpha_e) = x_e\alpha_e + (1-x_e)(1-\alpha_e)$. The message from each $B_e(.)$ check node to the α_e variable nodes for $e \in E$ is computed as

$$\begin{aligned} & \sum_{\sim\alpha_e}[x_e^i\alpha_e + (1-x_e^i)(1-\alpha_e)]P[X_e^i = x_e^i|X_W^i] \\ & \sum_{x_e^i}[x_e^i\alpha_e + (1-x_e^i)(1-\alpha_e)]P[X_e^i = x_e^i|X_W^i] \\ = \;& (\alpha_e + 0)P[X_e^i = 1|X_W^i] + (0 + (1-\alpha_w))P[X_e^i = 0|X_W^i] \\ = \;& \alpha_e\mu_e^i + (1-\alpha_e)(1-\mu_e^i) \quad \forall\, i \in \{1, N\} \end{aligned} \tag{5.57}$$

This is done at each time instant $i \in \{1, N\}$.

Now finally, message going out of the variable node α_e is by the sum-product algorithm, the product of all incoming messages. Therefore, the message at the output of the variable node α_e is

$$P[\alpha_e | X_W^{(1,N)}] \propto \prod_1^N [\alpha_e \mu_e^i + (1 - \alpha_e)(1 - \mu_e^i)] \tag{5.58}$$

We now evaluate this function (in the R.H.S. above) for every element of some discrete set $S = \{\frac{1}{L}, \frac{1}{L} \ldots 1\}$ for $L > 1$ and then choose $\hat{\alpha}_e$ as that element of S that maximises this function. Of course this entails an additional quantisation error, but if the discretisation is done finely enough (compared with the (intrinsic) variance of the estimator), then the effect of quantisation will not be felt keenly. Moreover, efficient use of computation power can be made by performing adaptive discretisation : using a coarse interval initially to make a rough estimate and then using a finer quantisation interval over a narrow region around the initial estimate of the maximum.

We show the posterior probability function in Figure 5.12.

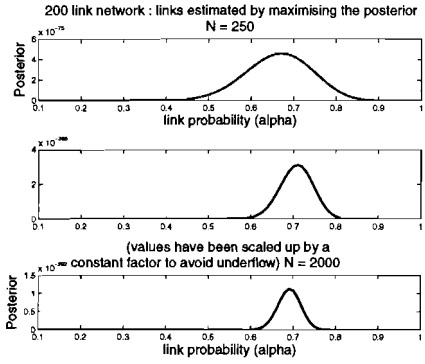


Figure 5.12: The posterior probability function in Equation 5.58 for 3 different values of the sample size in a 200 link network (which will be studied in Chapter 6). Note how the function becomes steeper as the sample size grows. Care must be taken to ensure that overflow/underflow does not occur during the maximisation process here.

Care must be taken especially in this step of the algorithm to ensure that overflow/underflow does not occur during the maximisation process here.

Now we apply the message passing algorithm on the 3 link network seen in Figure 5.10. We set all 3 links to α. We then sweep this parameter across its entire range, and for a fixed sample size (N) we find draw estimates of each of the links. The estimates of one link in the 3 link tree is shown in Figure 5.13.

We see that in a large scale sense, the mean of the estimate tends to the actual link probability (apriori) with increasing α, but that there is some local 'oscillation' of the mean about this apriori value.

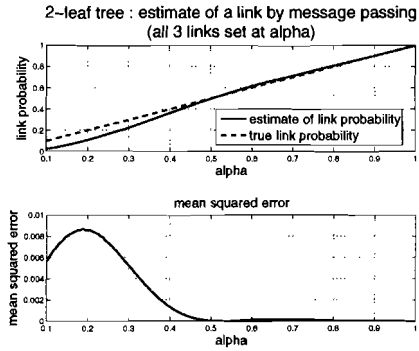


Figure 5.13: The message passing algorithm applied to the 3 link network in Figure 5.10. The estimate of one of the links as a function of the parameter α is shown here for a fixed N.

Performance of the message passing algorithm

Figure 5.7 in Section 5.4.1 shows the ENT resulting from application of the message passing algorithm on basic configuration 3. As compared to the cramer rao bound for the ENT as well as the performance of the suboptimal algorithm 1, message passing performs rather poorly. But we will see in Chapter 6 that this approach is much more suited for application to large networks.

Now we know that when the factor graph contains cycles, the sum-product algorithm will result in a sub optimal solution. The degree of sub-optimality is determined by the number and size of these cycles, i.e. the sparsity of the factor graph.

Now a cycle is created in the factor graph of a network configuration when

1. two different paths (set of links connecting some source to some receiver) have more than 1 link in common. Two paths that have two links in common give rise to $\binom{2}{2} = 1$ length 4 cycle in the factor graph. Two paths that have 3 links in common, give rise to $\binom{3}{2} = 3$ length 4 cycles. Hence in general, when two paths have 'm' paths in common, then this creates $\binom{m}{2}$ length 4 cycles in the factor graph.

   This is illustrated in Figure 5.14.

   Configuration 2 contains two paths w_2 and w_3 that share links 1 and 3, hence resulting in the length 4 cycle show in red. Note that the factor graph of configuration 3 does not contain any cycles.

2. a set say W_m of m paths cover a set E_m of m links, with each of the paths in W_m containing at least two links in E_m. This results in a length $2m$ cycle in the factor graph. See Figure 5.16.

   We see in this example that 6 paths if connected in this fashion results in a length 12 cycle, shown in red in the factor graph.

It is a well known fact that as the factor graph becomes more and more cyclic, the performance of the sum-product algorithm degrades. Hence, every additional link that two paths share adds tight (length 4) cycles to the factor graph, thus making it the more ill suited for application of the sum-product

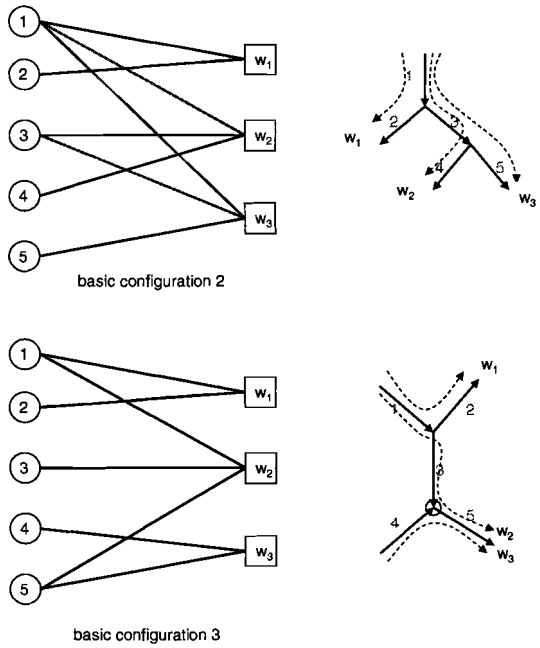


Figure 5.14: Basic configurations 2 and 3 and their factor graphs. The factor graph of configuration 3 s cycle free.

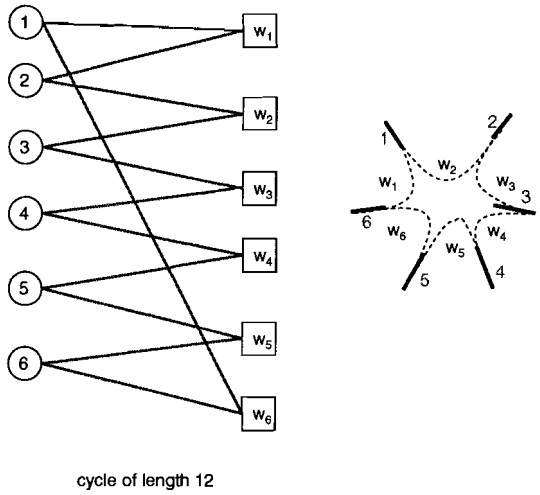


Figure 5.15: When paths in the network are connected to each other as shown, a cycle results. The dark black lines are the links which adjacent paths have in common. Here, the 6 paths result in a length 12 cycle in the factor graph.

algorithm. Also, as the number of sets of paths that are connected as described in point number 2 above, increases, the number of cycles in the factor graph increases as well.

Therefore, given a specific network configuration which we want to estimate by means of the message passing algorithm, we can make a fair prediction of its

performance by studying the nature of the factor graph that the configuration translates into. Any non-trivial network results in a cyclic factor graph. The 'average number of links contained per path' and the 'average number of paths that a link is contained in' gives a raw estimate of the extent of cyclicity of the factor graph. The lower the values of these two quantities, the more sparsely connected the factor graph could be expected to be. All this will be corroborated in Chapter 6 when we apply the message passing algorithm to large networks (45 link and 200 link) and show that it performs better on those configurations that result in sparsely connected factor graphs.

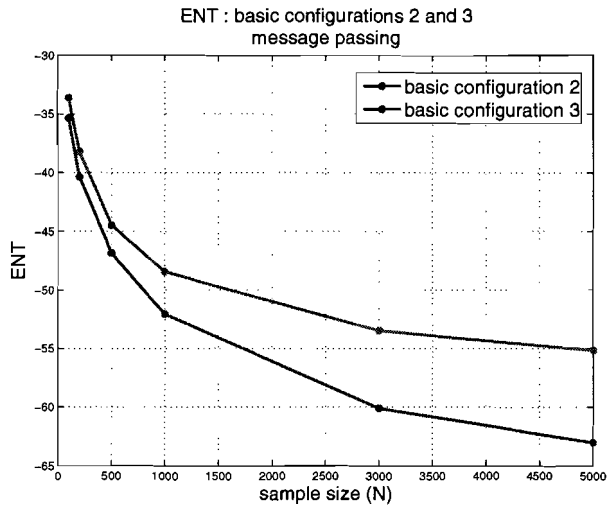


Figure 5.16: ENT of basic configurations 2 and 3 with the message passing algorithm. Note that the algorithm performs much better on configuration 3 whose factor graph is cycle free.

6

Simulation results

The estimation problem is defined by specifying the following quantities.

1. The adjacency matrix A of an undirected graph, $G = (V, E)$, representing the network whose links are to be estimated
2. Nodes in V which act as sources (these are leaf nodes)
3. Nodes in V which act as receivers (these are leaf nodes)
4. The set of nodes $V_c \in V$ where coding is performed

All other nodes are taken to be multicast nodes. All nodes in $\{V_c \bigcup V_m\}$, i.e. all coding and multicast nodes must be of degree 3 or higher. That is, we assume the network to contain only logical links since this is a necessary condition for identifiability (see Chapter 1).

Given which sources reach which receivers, there is a unique solution to the flow ρ in the network. We can thus rewrite the adjacency matrix A of the undirected graph as that of a directed graph which is uniquely characteristic of the chosen solution to the estimation problem (that is the specific choice of source, receiver and coding nodes). We thus know

- for every node $k \in V$, its parents ($f(k)$) and children ($d(k)$).
- for every edge $e \in E$, its origin node $f(e)$ and destination node $d(e)$.

Having done this, the estimation algorithms discussed in Chapter 5 can be applied to the directed graph.

Various solutions to the estimation problem are characterised by means of simulations. We apply all of the algorithms studied in Chapter 5 to each of the coding schemes chosen for study. These algorithms are, of course, suboptimal for arbitrary coding schemes (i.e. the scheme is not a multicast, inverse multicast or multicast-inverse multicast). Comparison of various solutions must therefore be made along 2 dimensions : (a) choice of coding scheme and (b) choice of estimation algorithm.

In Section 6.1, a 45 link network, is studied. We choose several coding schemes and then apply the algorithms in Chapter 5 as well as a modification/enhancement of the subtree decomposition method, tailored to this specific topology, which makes additional measurements. The idea is to get as close to

an optimal solution as possible in order to be able to make a *fair* comparison between arbitrary coding schemes and the traditional multicast approach (which is optimally estimated). In Section 6.1, we generate a random 200 link tree by means of a random branching process (to which any desired probability distribution can be applied) and apply the various coding schemes and the three suboptimal algorithms to this. Finally, our conclusions from the simulation results are listed in Section 6.3.

6.1 45 link network

We begin by considering a symmetric 45 link binary tree shown in Figure 6.1.
(6.0)
(6.1)

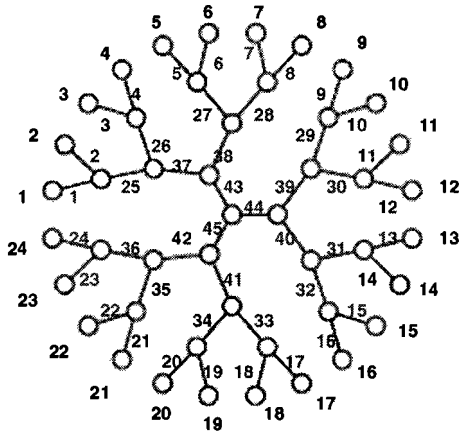


Figure 6.1: 45 link binary tree with 24 leaves

This is a depth 4 tree that has 24 leaves. We apply the following coding schemes

- multicast tree : 1 source, 23 receivers
- 2 sources, 22 receivers, 1 coding point
- 3 sources, 21 receivers, 2 coding points
- 5 sources, 19 receivers, 4 coding points
- 8 sources, 14 receivers, 7 coding points (this is an inverse multicast-multicast tree)
- 11 sources, 13 receivers, 10 coding points

In this section, we will denote the source i by S_i and the probe packet it generates as x_i.

In the case of the multicast tree, the single source can be placed at any of the 24 leaves, and these will be equivalent because of the symmetry (if all links

are assumed to have the same link probability). There is only 1 type of flow throughout the entire graph.

With 2 sources, the graph is divided into 3 subtrees as shown in Figure 6.2.

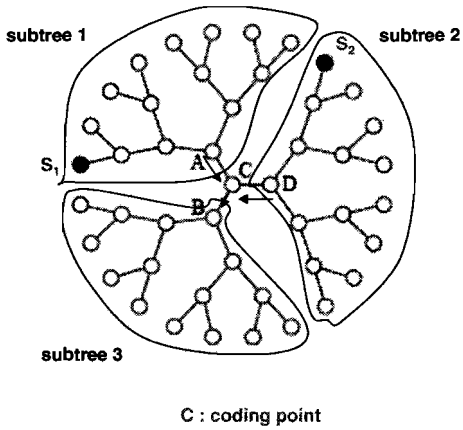


Figure 6.2: 45 link tree : 2 sources, 1 coding node, 3 subtrees.

We first apply the suboptimal algorithm 1 and the message passing algorithm to this configuration. The average of the estimates of the 45 link probabilities obtained for a sample size (N) of 250 is shown in Figure 6.3.

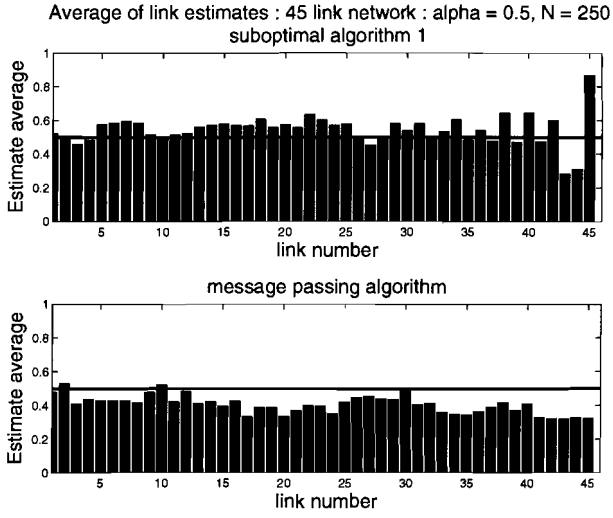


Figure 6.3: 45 link tree with 2 sources. All links are set to $\alpha = 0.5$. Suboptimal algorithm 1 and message passing.

Note how, with the suboptimal algorithm 1, the 2 incoming links into the coding node, links 43 and 44, are underestimated, while the outgoing link 45 is overestimated. The average of the estimates (with suboptimal algorithm 1 as the estimator) of these 3 links are plotted as a function of α in Figure 6.4 for $N = 5000$.

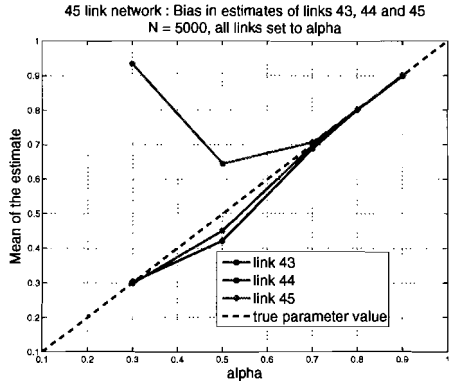


Figure 6.4: 45 link tree with 2 sources (N=5000).

Notice how, especially for lower values of α, the estimates are strongly biased away from the true parameter value. These 3 links tend to contribute significantly to he degradation of the ENT. The ENT was defined in Chapter 2 and is written here again

$$\text{ENT} = \sum_{e \in E} \log_2 \left([\hat{\alpha}_e - \alpha_e]^2 \right), \tag{6.3}$$

Estimates of other links in the network are not so strongly biased. Let us therefore look at what happens if we disregard the estimates of these 3 links (discount them in the calculation of the ENT) and then compare the multicast estimator with the 2-source estimator. The results are shown in Figure 6.5.

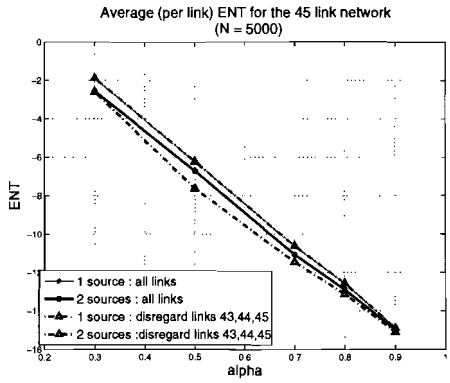


Figure 6.5: 45 link tree, $N = 5000$. ENT of multicast tree and 2-source tree when links 43, 44 and 45 are disregraded.

We see that eliminating these three links really widens the performance gap between the multicast case and the 2-source case. This results from improvement (decrease) of the ENT of the 2-source estimator. That ENT of the multicast estimator is unchanged as expected and is confirmed by the graph.

As mentioned earlier, we will now develop a heuristic tailored to the specific 45 link, 2-source topology that 'enhances' the subtree method. The procedure is described here.

Enhanced subtree method (2 sources): When no observation is made at the receiver subset of coding node,

- if a measurement is made at receiver nodes 5, 6, 7 or 8, then assume x_1 has entered C.
- if a measurement is made at receiver nodes 13, 14, 15 or 16 then assume x_2 has entered C.
- otherwise, assume neither x_1 nor x_2 entered C.

We now apply suboptimal algorithm 1, the subtree method and enhanced subtree method to the directed graph representing the 2-source network and obtain the results shown in Figure 6.6. The enhanced method is superior for small values of N (< 1500), whereas algorithm 1 does better for higher values of N. There is negligible difference in performance in the two however. It is the subtree method that in this case seems to have a relatively high (bad) value of ENT.

We set 5 of the leaves as sources, and place these uniformly around the network as shown in Figure 6.7.

The graph is subdivided symmetrically into 6 subtrees, each carrying a characteristic flow. We make enhanced measurements as in the 2 source case and will show that this will yield considerably more benefit over the general heuristics than in the 2 source case.

Enhanced subtree method (5 sources): The measurements made additional to that in the subtree method in order to estimate the probability of each of the 4 coding nodes receiving a packet from each parent is illustrated by means of the flowcharts below. We have included the flowchart for coding nodes A and D as examples in Figure 6.8.

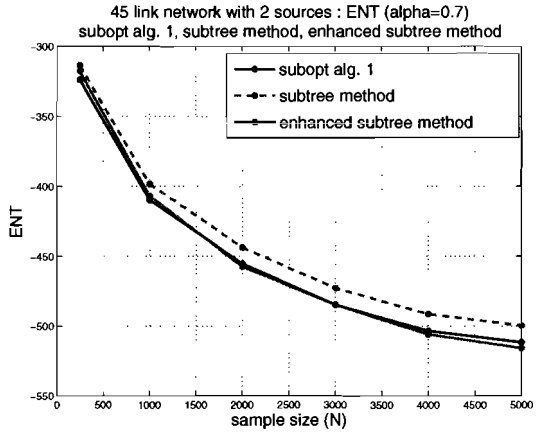
Measurements are made both at receivers 'above' coding node A and of sources 'below' coding node A in order to estimate the probability of A receiving a packet from S_1. The same procedure is followed for estimating the probability of A receiving a packet from S_2, and likewise for coding node D.

Let us call the fraction of events wherein x_1 is measured at the receiver subset (nodes 17-24) of coding node A as μ_A^1. This is the estimate of the probability of A receiving x_1 as per the subtree decomposition method. We now use the additional measurements shown in the flowchart to derive an improved estimate $\mu_A^{1'}$ of the this probability as

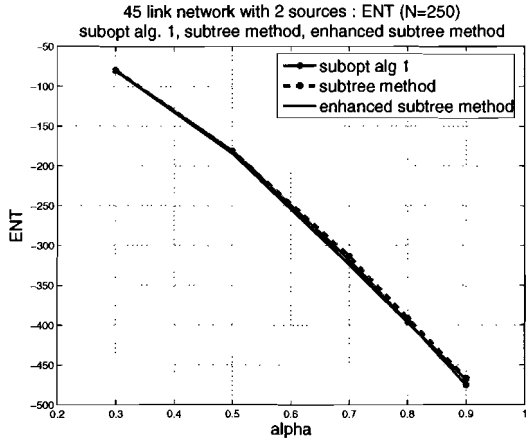
$$\mu_A^{1'} = \mu_A^1 + t_1 + \frac{t_2}{2} + \frac{t_3}{2} \tag{6.4}$$

The measurements made to improve the estimate of the probability of coding node B receiving a packet from its first parent, coding node A, i.e. of receiving a packet from S_1 or S_2 is shown in Figure 6.9. (6.4)

Let us call the fraction of events wherein x_1, x_2 or both is measured at the receiver subset (nodes 17-24) of coding node B as μ_1. This is the estimate of the probability of A receiving a packet from its first parent as per the subtree decomposition method.We now use the additional measurements shown in the flowchart to derive an improved estimate $\mu_B^{1'}$ of the this probability as



(a) ENT vs sample size for $\alpha = 0.7$



(b) ENT vs α for N=250

Figure 6.6: 45 link network, 2 sources

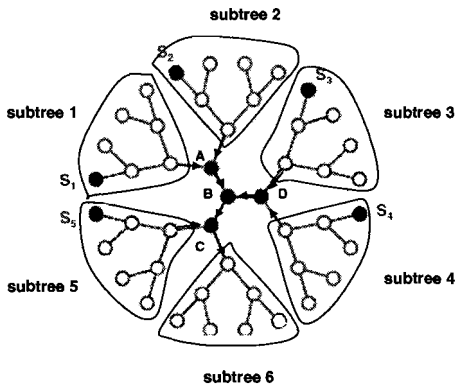


Figure 6.7: 45 link tree : 5 sources, 4 coding nodes, 6 subtrees.

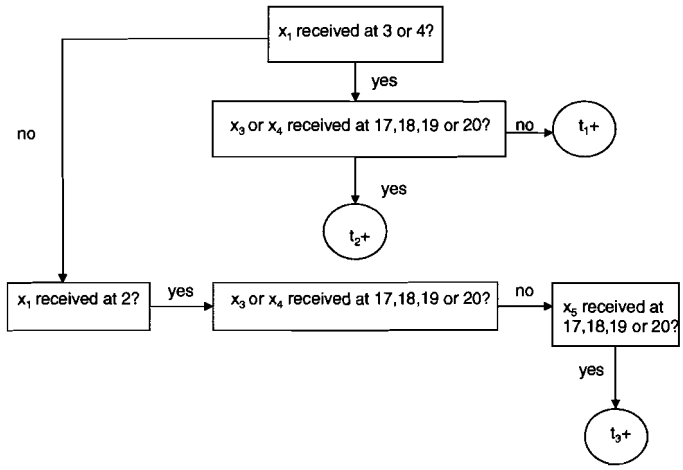


Figure 6.8: Estimating the probaility of coding node A receiving packet x_1 form S_1.

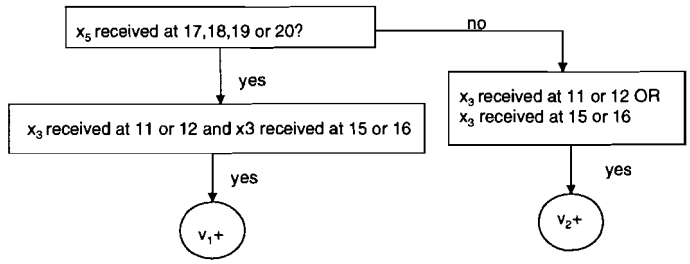


Figure 6.9: Estimating the probaility of coding node B receiving a packet from S_1 or S_2.

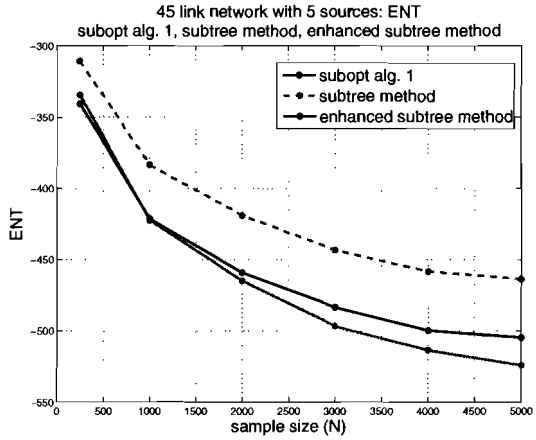
$$\mu_B^{1'} = \mu_B^1 + \frac{v_1}{2} + v_2 \tag{6.6}$$

The results are shown in Figure 6.10. The improvement rendered by the enhanced method as compared to the other 2 routines is much more evident in this 5 source case as compared with the 2 source case and as seen, the algorithm outperforms the others especially at low values of α and N.

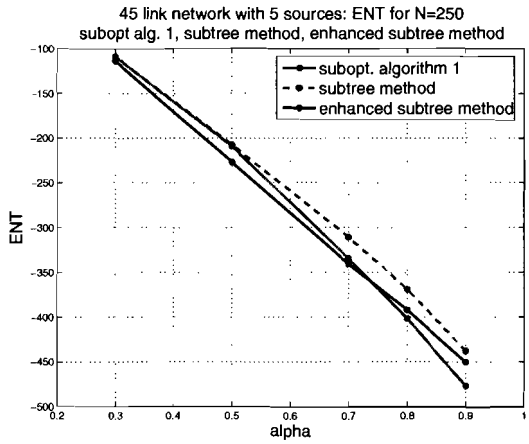
We now place 3 sources as shown in Figure 6.11.

The graph is divided into 4 subtrees. We have made no enhanced measurements in this case, and merely apply the general heuristics to the topology. The results will be depicted shortly. Finally, for the sake of comparison, we set 8 neighbouring leaves as sources as shown in Figure 6.12.

The graph here is divided into 2 subtrees - one is a multicast subtree and the other an inverse multicast subtree and as we have seen in Chapter 5, the links in such a graph can be optimally estimated. This topology is not of much practical use however, since we do not want to place all the coding points in one section of the network. It will however provide a benchmark for comparison with the other schemes.



(a) ENT vs sample size for $\alpha = 0.7$



(b) ENT vs α for N=250

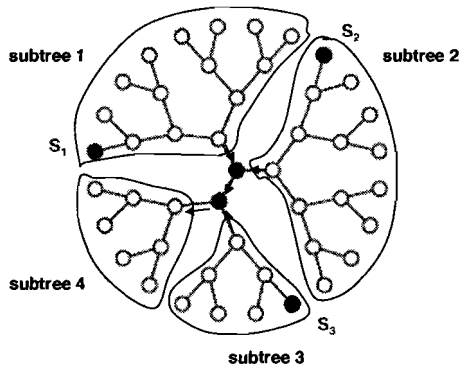


Figure 6.11: 45 link tree : 3 sources, 2 coding nodes, 4 subtrees

Finally, we placed 12 sources uniformly around the network as shown in

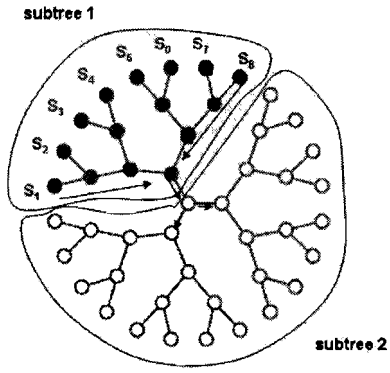


Figure 6.12: 45 link tree : 8 sources, 14 coding nodes

Figure 6.13.

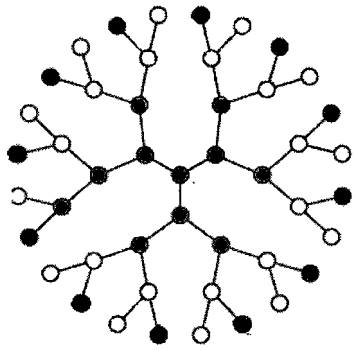


Figure 6.13: 45 link tree : 12 sources, 11 coding nodes, 12 receivers

Now we have seen that a comparison has to be made along 2 dimensions - performance of each of the heuristics and weighing up different coding schemes against each other. We have made some initial study of the first dimension. We now do the following.

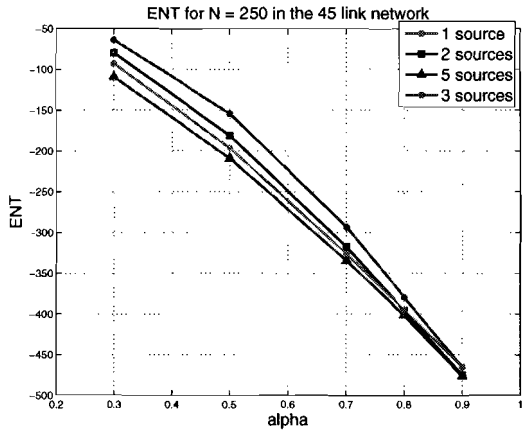
1. Set the sample size to some constant N and then plot the ENT as a function of α - the parameter which defines all 45 link probabilities. The ENT resulting only from the algorithm that is seen to perform the best are shown here.

2. Set all link probabilities to some constant α and plot the ENT as a function of sample size (N).

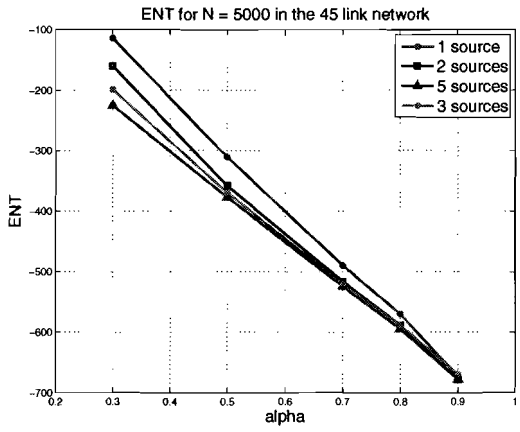
The α-sweep for $N = 250$ is shown in Figure 6.14.

The sweep with $N = 5000$ is redrawn with the results from the 8-source case included in Figure 6.15.

This scheme does rather badly for values of $\alpha < 0.5$, but improves rather swiftly for higher values. Finally, we plot the α-sweep with the results of the 12-source case included in Figure 6.16.



(a) ENT vs α for N=250



(b) ENT vs α for N=5000

Figure 6.14: 45 link network, 4 different coding schemes

The message passing algorithm delivers best results for the 12-source configuration (black line), hence this is shown in the plot. We also show the ENT resulting from the 5-source cum message passing estimator in the plot for the sake of comparison.

The message passing cum 12-source configuration estimator has the lowest ENT for values of $\alpha < 0.65$ whereafter, the performance comparatively degrades, becoming worse than even the multicast estimator for values of $\alpha > 0.75$.

The conclusion to be drawn is clear. At least for the values of N considered here, it is seen that performing coding results in much better estimators as compared to when no coding (multicast) is done, and it must be emphasised here that this is despite the fact that the multicast tree is optimally estimated whereas all the other coding schemes are sub-optimally treated.

Now, we set all links in the network to $\alpha = 0.7$ and vary the number of probes/sample size from 250 to 5000 and plot the results in Figure 6.17.

Part (a) shows that the trends observed in Figure 6.14 are consistent here for the entire sweep of N. The ENT drops steeply initially, and then flattens out

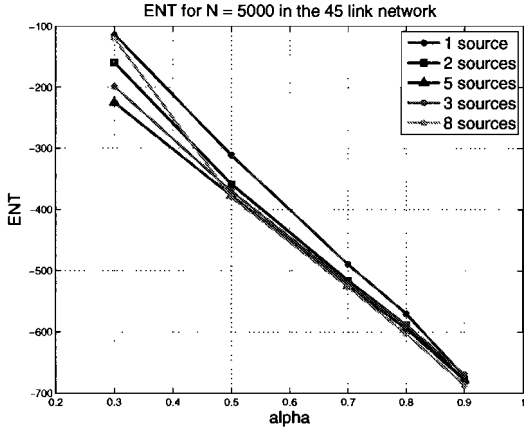


Figure 6.15: 45 link network, N=5000, α-sweep, 5 different coding schemes

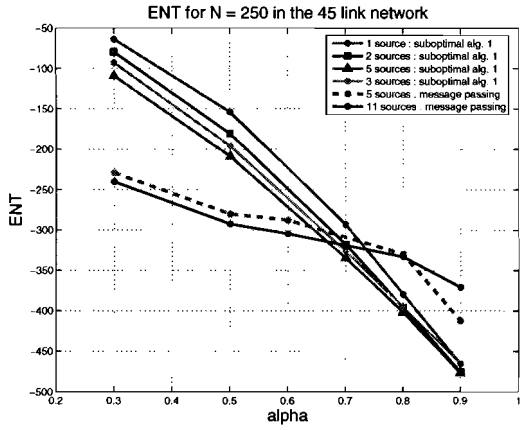


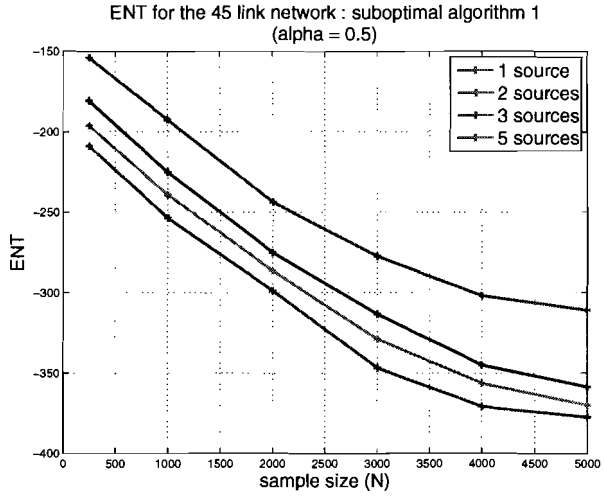
Figure 6.16: 45 link network, N=250, α-sweep, 5 different coding schemes

for $N > 4000$ samples. Part (b) plots the ENT of the message passing estimator as well. We see that this algorithm is inferior to the estimator for the multicast tree (this is an ML estimator) as well as the 2-source tree (estimated with the suboptimal algorithm 1). However, it does a great deal better than the latter for the 3-source case and even more so in the 5-source case.

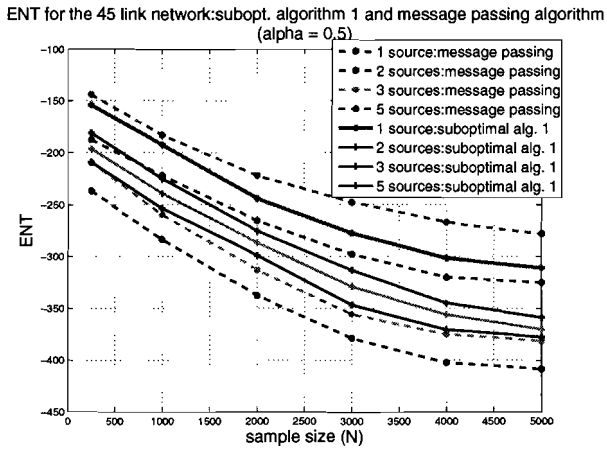
This trend can be explained by looking at the nature of the factor graphs that each of these topologies translates into. Consider Table 6.1. We see that as the network becomes more sparse, the message passing algorithm performs better.

This is because the factor graph improves, since as we saw in Chapter 5, the greater the number of links two paths have in common with each other (directly or indirectly), the more cyclic is the factor graph and the less suited it becomes for application of the belief propagation approach. The factor graphs of the 1-source and 5-source topologies are shown in Figure 6.18.

We plot the average (per link) ENT for three of the coding schemes in



(a) ENT vs N for $\alpha = 0.5$, suboptimal algorithm 1



(b) ENT vs N for $\alpha = 0.5$, suboptimal algorithm 1 and message passing

Figure 6.17: 45 link network, 4 different coding schemes, suboptimal algorithm 1 and message passing

Figure 6.19. The average ENT was defined earlier and is reproduced here.

$$\mathrm{ENT}_{av} = \log_2 \left(\frac{1}{E} \sum_{e \in E} [\hat{\alpha}_e - \alpha_e]^2 \right)$$

Note that the average ENT is a much 'harsher' metric for comparison. The average mean squared error of the 5-source estimator is about $\frac{1}{2}$ of that of the multicast estimator for $N = 5000$. This is a considerable difference.

coding scheme	links per path	paths per link
1 source	7.0435	3.6
2 sources	6.5333	4.3556
3 sources	5.7241	3.6889
5 sources	5.7714	4.4889
12 sources	4.5217	2.3111

Table 6.1: Sparsity of configurations resulting from the 45 link graph.

6.2 200 link network

Having seen the results up until this point, it seems that we can but only conclude that having no apriori information of the link probabilities, we are much better off performing coding than not. However, the 45 link network studied is a symmetric graph and in order to convince ourselves of the merits of network coding by means of an even more firmly founded conclusion, we generate a random 200 link tree (acyclic graph) and repeat the experiments we did on the 45 link network here. The tree is shown in Figure 6.20.

The tree has 102 leaves. We study 4 different estimators.

- 1 source (multicast tree)
- 2 sources
- 4 sources
- 7 sources

Note that the sources are chosen such that the source in the multicast case is one of the sources in the 2-source case, and these two sources are among the 4 sources in the following scheme and so forth. This gives us a firm ground for performance comparison. We first set all 200 links to $\alpha = 0.7$ and then perform a sample size sweep, the results of which are shown in Figure 6.21.

We have applied the suboptimal algorithm 1 and the subtree methods here. Again, the conclusion is plain to see, with the multicast estimator being the worst, followed by the 2-source, 4-source and the 7-source estimator being the best. Note that the subtree method performs nearly as well as the first algorithm for the 2-source case, but its performance degrades with increasing number of coding points (sources).

We now set the sample size $N = 1000$ and perform an α-sweep. The results are shown in Figure 6.22.

Here, we apply the suboptimal algorithm 1 and the message passing algorithms. The message passing algorithm performs considerably poorer than the maximum likelihood estimator for the multicast (1-source) case. Its performance is better than the suboptimal algorithm 1 for low values of α in the 2-source case, but this trend is reversed for higher values ($\alpha > 0.6$).

Note how the variance of the suboptimal 1 estimator decreases faster than the variance of the message passing estimator with increasing α. It has been observed that for relatively large values of α, a sizeable number of links are estimated by the message passing algorithm to have link probabilities close to 1.

Now as we move on to the 4-source and 7-source configurations, we see that the message passing algorithm greatly outperforms the other algorithm up until values of α around 0.7, the 2 algorithms performing more or less equally thereafter. Again, this trend can be explained by looking at the sparsity of the directed graphs that result from application of these coding schemes to the 200 link topology. See Table 6.2.

coding scheme	links per path	paths per link
1 source	10.0891	5.0697
2 sources	9.1496	5.7811
4 sources	7.1034	4.0995
7 sources	6.9333	4.6567

Table 6.2: Sparsity of configurations resulting from the 200 link graph.

We now look at the average (per link) entropy of the multicast estimator when applied to the 45 link and 200 link topologies. The results are shown in Figure 6.23.

We plot the ENT of both the ML estimator and the message passing estimator. We see that the average ENT is better for the 45 link network for a given number of probes. Also, note that the message passing algorithm does relatively better in the 45 link case since the tree is more sparse than in the 200 link case (see the two tables above for sparsity values).

6.3 Conclusions

We conclude with some remarks.

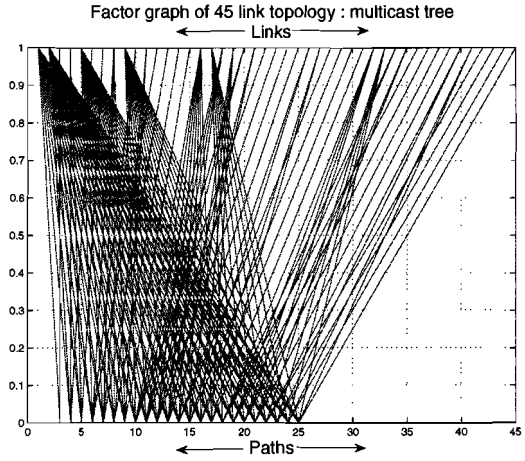
1. We note that the subtree method does well for the 2-source configuration, but that its performance becomes poorer for increasing number of sources (and hence coding nodes). Our observations show the method is suitable when the coding points are deep within the network , i.e. they must not have any sources in their proximity. The closer the coding nodes are to their receiver subsets, the better suited is the subtree method.

2. The suboptimal algorithm 1 on the other hand is more favourable to configurations wherein the coding nodes are close to their source subsets. Hence, the growing disparity between the 2 methods as the number of coding points is increased.

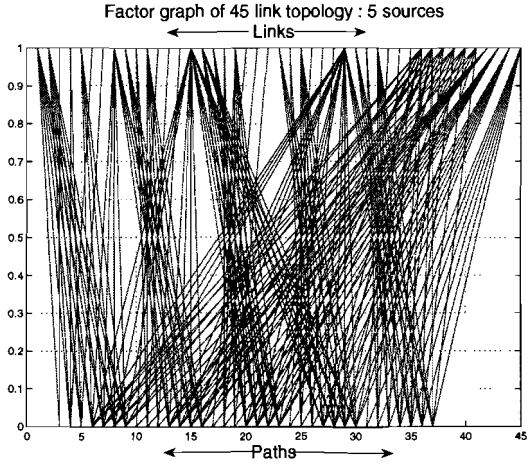
3. The message passing algorithm does well for configurations that translate into sparse factor graphs. The merits of this method over the other 2 methods are more keenly felt for lower values of α and sample size (N). The MSE of this estimator drops rather slowly with α and N as compared with MSE of the other 2 methods. Hence, as α and/or N become larger, the performance of the suboptimal method 1 and the message passing method approach one another. The summary messages were observed to definitely converge after 15 iterations. In our simulations however, we set the number of iterations to 30.

4. Having seen all this, our conclusion is that (well chosen) coding schemes result in far better estimators than the multicast estimator, and this is despite the fact that arbitrary coding schemes are sub-optimally estimated. The complexity of the suboptimal algorithm 1 is the same as that of the multicast estimator (maximum likelihood).

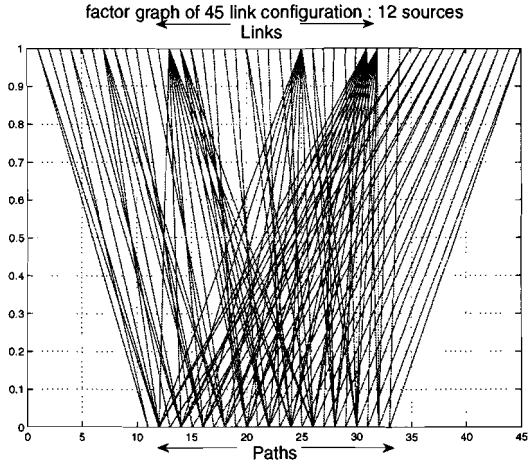
5. We have seen that certain coding schemes are much better estimated by the suboptimal algorithm 1 (schemes fewer coding nodes) as compared to the message passing approach, whereas certain other coding schemes (generally with a larger number of coding nodes, and sparse factor graphs) are better estimated by the latter. So in general, for any coding scheme, we do have some sub-optimal routine that performs well enough to justify preference for the network coding approach over the multicast approach.



(a) Factor graph of the 45 link multicast tree



(b) Factor graph of the 45 link tree with 5 sources



(c) Factor graph of the 45 link tree with 12 sources

Figure 6.18: Factor graph of the 5-source, 12-source and 1-source configurations

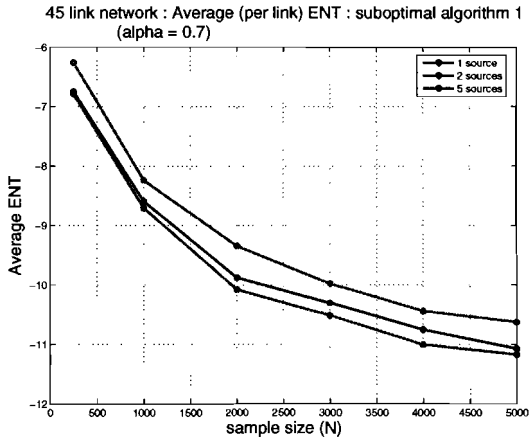


Figure 6.19: Average ENT in the 45 link network, $\alpha = 0.7$, 3 different coding schemes

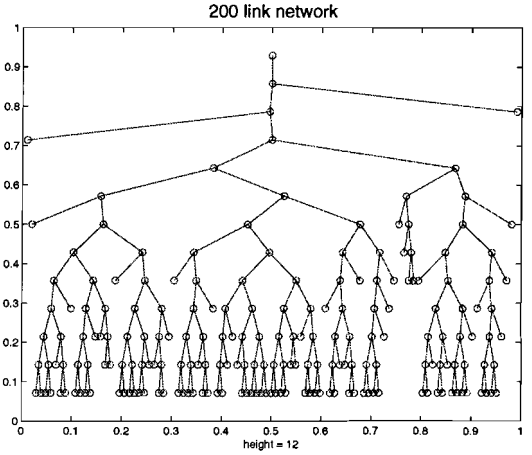
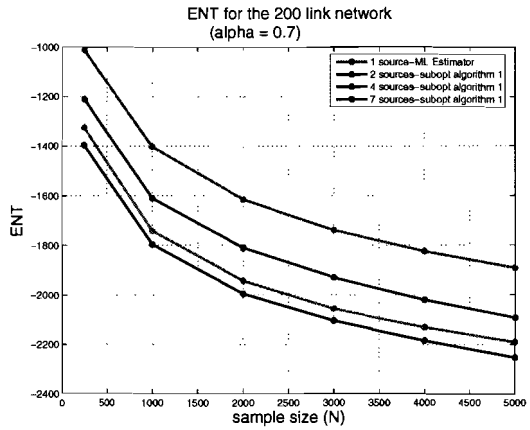
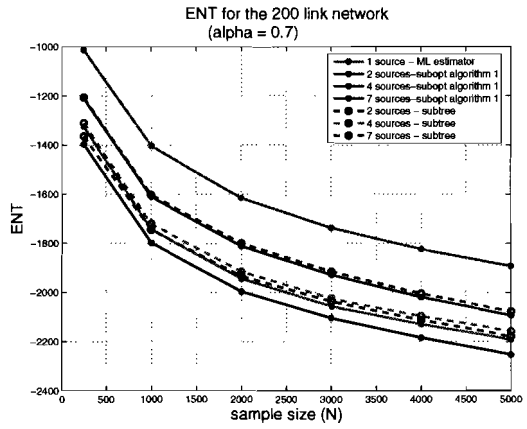


Figure 6.20: Randomly generated 200 link topology (tree)



(a) ENT vs N for $\alpha = 0.7$, suboptimal algorithm 1



(b) ENT vs N for $\alpha = 0.7$, suboptimal algorithm 1 and subtree method

Figure 6.21: 200 link network, 4 different coding schemes, suboptimal algorithm 1 and subtree method

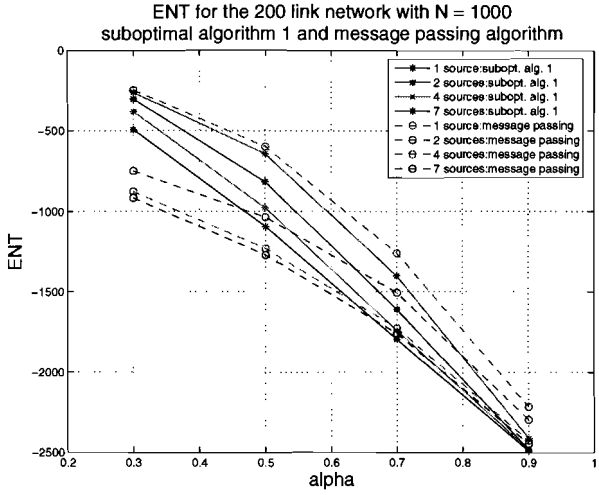


Figure 6.22: 200 link network, N=1000, α-sweep, 4 different coding schemes

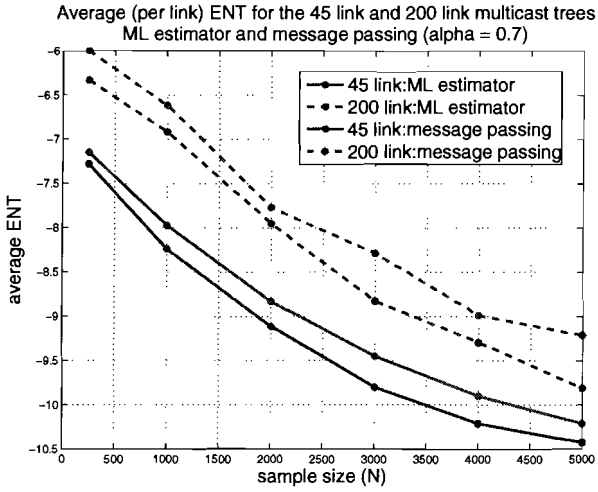


Figure 6.23: Average (per link) ENT for the multicast estimator applied to the 45 link and 200 link topologies.

7

Cyclic networks

Heretofore, we have restricted ourselves to networks that are trees or un-directed acyclic graphs. Here, we address the estimation problem when the network is a cyclic graph. The network coding principles that apply to the tomographic problem when the underlying topology is a tree can be easily extended to apply to cyclic topologies as well. Formally, a cycle of a graph , sometimes also called a circuit, is a subset of the edge set of that forms a path such that the first node of the path corresponds to the last. In other words, in our context, if a source reaches a node in the network via more than one path, this constitutes a cycle. We illustrate the estimation problem applied to a cyclic graph by means of a small example. Consider the 8 link graph shown in Figure 7.1

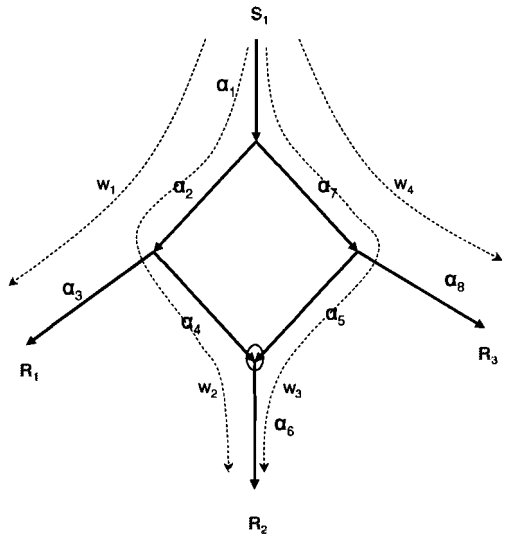


Figure 7.1: (i). The solution to the estimation problem employs a coding scheme with 1 source, 3 receivers and 1 coding point

The links e_2, e_4, e_5 and e_7 constitute the cycle. All links are required to be identified. The topology is symmetric, and if we assume no apriori information regarding link probabilities, there are two solutions to the estimation problem : (i) 1 source, 2 receivers (b) 2 sources, 1 receiver (the scheme with 3 sources and

1 receiver would be equivalent to (ii) by the reversibility theorem, hence is not considered). (i) is shown in the Figure 7.1.

Note that the information flow (flow of probe packets) in solution (i) is cyclic, with the source S_1 reaching receiver R_2 via 2 different paths w_2 and w_3. The factor graph corresponding to this configuration is shown in Figure 7.2.

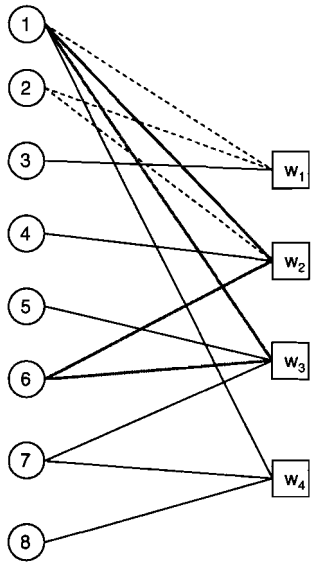


Figure 7.2: Factor graph of the configuration (i)

The factor graph is cyclic with three tight (length 4) cycles; the cycle corresponding to the paths w_2 and w_3, which constitute the cycle in the network is shown in red. One of the other two tight cycles is shown by the dashed line.

Solution (ii) is shown in Figure 7.3. Note that the information flow in case (ii) is *not* cyclic. Each of the sources S_1 and S_2 reaches each of the two receivers R_1 and R_2, but only via a single route. Hence, *a cycle in the (physical) network need not imply a cycle in the information flow, depending on the choice of coding scheme.* The factor graph of this configuration is in Figure 7.4.

There are no tight (length 4) cycles here, the single cycle of length 8, being shown in red.

Now solution (i) requires the network coding principles we have seen to be extended to accommodate the cyclic information flow.

The operation at the coding node instead of being an *xor* of its inputs as has been the case heretofore, now requires the first input to be multiplied by 2 and added to the second input. The information flow is thus not described over a binary field any more, but over a quaternary field. If the information flow were to go through more than 1 cycle, the field order would have to be increased suitably, but the basic coding principle remains the same. The principle of working is explained in Table 7.1 and is demonstrated in Figure 7.6.
(7.0)

All links are identifiable. We can work out the maximum likelihood functions for the configurations (i) and (ii) and compute the inverse Fisher information matrix. We plot the ENT function for these two cases in Figure 7.5.

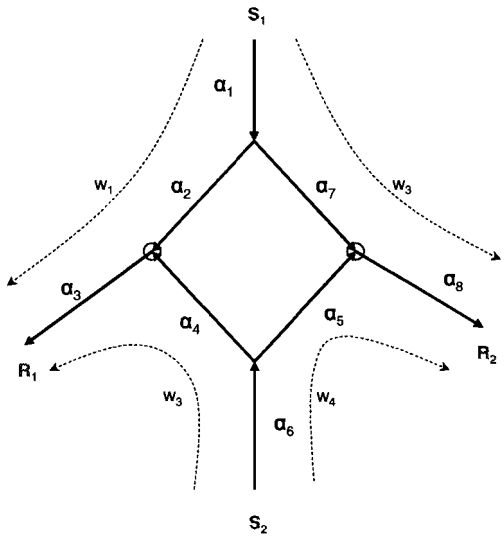


Figure 7.3: (ii). The solution to the estimation problem employs a coding scheme

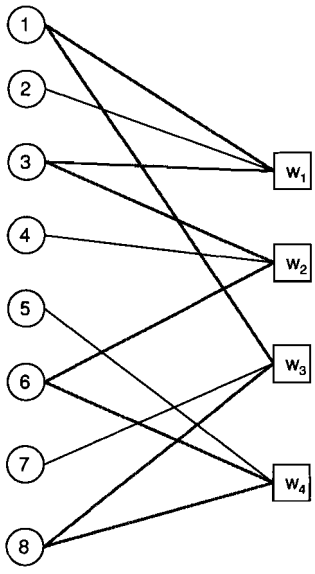


Figure 7.4: Factor graph of the configuration (ii)

The lower bound to the variance has just been plotted for illustration; our purpose is not to compare the two schemes in terms of quality of estimation, but merely to show that network coding throws up a variety of solutions when a cyclic network must be estimated.

If such a solution is chosen that involves a cyclic flow of information, the field order has to be increased; but when the solution does not cause information to flow in cycles, then the field order can be preserved as is.

Again, we note that a multicast approach would offer only the messy solution

R_1	R_2	R_3	w_1	w_2	w_3	w_4
x	$3x$	x	1	1	1	1
x	$2x$	x	1	1	0	1
x	x	x	1	0	1	1
x	ϕ	x	1	0	0	1
x	$3x$	ϕ	1	1	1	0
x	$2x$	ϕ	1	1	0	0
x	x	ϕ	1	0	0	0
x	ϕ	ϕ	1	0	0	0
ϕ	$3x$	x	0	1	1	1
ϕ	$2x$	x	0	1	0	1
ϕ	x	x	0	0	1	1
ϕ	ϕ	x	0	0	0	1
ϕ	$3x$	ϕ	0	1	1	0
ϕ	$2x$	ϕ	0	1	0	0
ϕ	x	ϕ	0	0	1	0
ϕ	ϕ	ϕ	0	0	0	0

Table 7.1: Solution (i) : observation space and functional paths. The state of a path is '1' if it is functional and '0' otherwise.

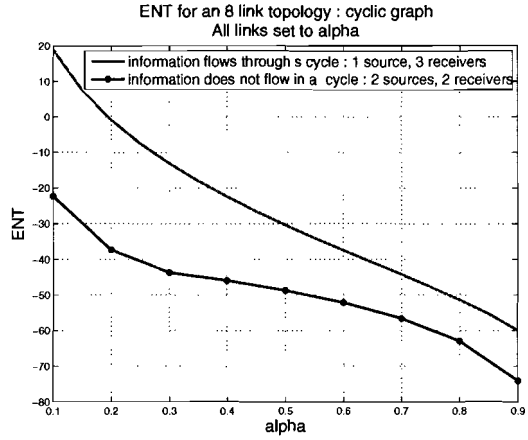


Figure 7.5: ENT of the two solutions for the cyclic 8 link topology for a fixed sample size.

of performing multiple experiments on all the different subtrees that span the graph and then combining these estimates.

The three suboptimal algorithms discussed in Chapter 5 can be applied without any modification to cyclic graphs, with the only difference being that the field might no longer be binary.

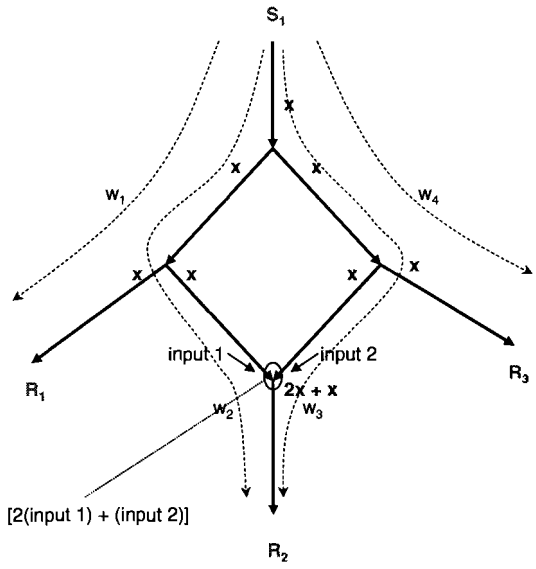


Figure 7.6: Operation at the coding node in configuration (ii)

8

Conclusions and recommendations

The subject of study in this project has been network coding based inference of link probabilities. Given a network whose links are desired to be estimated, we devised various estimators, which are a combination of both the coding scheme employed and the heuristic applied to derive link estimates.

Based on the analytical and experimental results that have been used to characterise these various estimators, we draw a few conclusions and make recommendations to further research in this area.

1. For any network, the coding scheme employed, i.e. the choice (number and placement) of the nodes that act as sources, receivers and coding points determines the quality of the estimator. This dependence again, is a function of various other factors such as the actual link probabilities of the links in the network, the size of the network and the shape of the topology.

2. The difference in the performance of the estimators is not seen to be extremely large when the network in question is relatively small, that is having below 10 links or so. The difference in performance of various estimators becomes much more evident as the size of the network grows.

3. For all the graphs we studied and coding schemes we applied, the conclusion is clear : given that we have no apriori information as regards the link probabilities in the network, we are much better off employing network coding based inference than simply multicast based inference.

4. The benefits rendered by network coding based inference are much more pronounced at low values of α and low values of sample size (N). This is nice since it means that network coding results in estimators that deliver performance far superior to multicast based estimators when a small number of probes is desired to be used.

5. We saw that distributing the sources at roughly equal distances along the periphery of the network such that the tree is partitioned into roughly equal-sized subcomponents generally yields a good estimator.

6. The suboptimal 1 algorithm performs well when the number of multicast nodes (greatly) outnumber the number of coding nodes. This holds true for the subtree method as well. The latter performs well when the coding points are located far away from the periphery (source leaves, in particular) of the network. We also saw how we can make 'enhanced' measurements to improve the performance of the subtree estimator.

   The message passing algorithm performs well when the configuration resulting from application of the coding scheme to the given topology results in a sparse factor graph. We also saw that the algorithm's benefits over the other 2 sub-optimal algorithms are more keenly felt at small values of α and sample size (N). The fall of the mean squared error of this estimator with increasing α and N was seen to be much slower than the other 2 heuristics. In general, this method performs well for large networks.

7. Given that we have observed such a large improvement in the quality of estimation when network coding is chosen over multicast and more so, despite the fact that all of the network coding based inference methods are sub-optimal whereas the multicast based estimators are optimal, makes the problem of deriving a heuristic for ML estimation of arbitrary coding schemes a very valid and important one to research further. Though network coding based estimators are superior enough to justify their use (in the sub-optimal form), the benefits delivered if optimal estimation procedures cold be developed would definitely be greater.

8. The network coding estimator should be implemented in the ns-simulator and its performance studied on networks that emulate the behaviour of the internet.

9. The performance of various coding based estimators on general graphs (networks which contain cycles) should be studied in greater extent.

A

Fisher Information Matrix is non-singular

1. If $k \in V_m$, the link probability of the incident *incoming* to k is denoted by α_k; if $k \in V_c$, that of the *outgoing* link from k is denoted by α_k (A.0)

2. $\alpha \in (0,1)^{|U|}$

3. The set of sources is S, set of receivers is R, receiver subset of node k is $R(k)$ and the source of k is called $S(k)$.

4. The subtree rooted at node k is $T(k) = (V(k), E(k))$. If $k \in V_m$, the set of nodes in the subtree, $V(k) = \{l \in V \ : \ l \preceq k\}$. If $k \in V_c$, the set of nodes in the subtree, $V(k) = \{l \in V \ : \ l \succeq k\}$.

5. In a multicast tree, the vector received at R is $x = (x_i)_{\ \forall i \in R}$ with $x_i \in \{0,1\}$ being the packet received at a certain receiver i. In an inverse-multicast tree, the vector received at the receiver R is $x = (x_i)_{\ \forall i \in S}$ where $x_i \in \{0,1\}$ is the bit received from the i^{th} source S_i. Note that the bit positions received by any node k are indexed by its source subset, $S(k)$.

6. For $k \in V_m$, X_k is the state of the path from the source (S) to node k, i.e. $X_k = 1$ if the path functions (a packet is received at node k) and otherwise, $X_k = 0$ if the path fails (no packet is received at node k). For $k \in V_c$, Y_k is the state of the path from the node k to the receiver (R), i.e. $Y_k = 1$ if the path functions (a bit from $S(k)$ is received at receiver R conditioned on it being received by node k) and otherwise, $Y_k = 0$ if the path fails (a bit from $S(k)$ is not received at receiver R conditioned on it being received by node k).

7. For a multicast tree, $U = V \setminus \{S\}$ and for an inverse-multicast tree, $U = V \setminus \{R\}$.

A.1 Multicast and inverse multicast trees

We look at a multicast tree first.

The score vector $S(\alpha) = (S_k(\alpha))_k$ with $S_k(\alpha) = \frac{\partial L}{\partial \alpha_k}(\alpha)$ (A.1)

The Fisher information matrix is

$$\begin{aligned} I_{jk}(\alpha) &= \mathrm{Cov}(S_j(\alpha), S_k(\alpha)) \\ &= E_\alpha(S_j(\alpha)S_k(\alpha)) \\ \because E_\alpha(S_\alpha) &= \sum_{x\in\Omega} p(x,\alpha)\frac{\partial}{\partial\alpha_k} log(p(x,\alpha)) \\ &= \sum_{x\in\Omega} \frac{\partial}{\partial\alpha_k}(p(x,\alpha)) \\ &= 0 \end{aligned} \tag{A.3}$$

Now suppose that $I(\alpha)$ is singular for some $\alpha = (\alpha_k)_{k\in U} \in (0,1)^{|U|}$. Then there exists some non-zero vector $c = (c_k)_k$ for which $c.I.c^T = 0$. But $c.I.c^T$ is the variance of the mean zero random variable $c.S(\alpha)$, so then we would have that $c.S(\alpha) = 0$, P_α almost surely, or equivalently

$$\sum_{k\in U} c_k \frac{\partial\ \mathrm{log}p(x,\alpha)}{\partial\alpha_k} = 0 \qquad \forall x \in \Omega \tag{A.4}$$

since $P_\alpha(\{x\}) > 0$ for all $x \in \Omega$. We show that in fact, equation A.4 implies $c_k = 0$, first for $k \in R$, then for all $k \in U$. (A.4)

1. We define the event $x^{(0)} \in \Omega$ such that $x_i^{(0)} = 1 \forall\ i \in R$. Then

$$p(x^{(0)},\alpha) = \prod_{i\in U} \alpha_i \tag{A.6}$$

2. We define the event $x^{(1)} \in \Omega$, such that for some $k \in R$

$$x_i^{(1)} = \begin{cases} 1 & i \neq k \\ 0 & i = k \end{cases} \quad \forall k \in R$$

Then

$$p(x^{(1)},\alpha) = \bar{\alpha_k} \prod_{i\in U\setminus\{k\}} \alpha_i \tag{A.7}$$

And so from equations A.4,A.6 and A.7, we have

$$\sum_{i\in U} \frac{c_i}{\alpha_i} = 0 \text{ and} \tag{A.8}$$

$$-\frac{c_k}{\bar{\alpha_k}} + \sum_{i\in U\setminus\{k\}} \frac{c_i}{\alpha_i} = 0 \tag{A.9}$$

Combining these last two equations, we find $c_k = 0$.

We now proceed by induction. For some $k \in U$, assume that $c_i = 0\ \forall\ i \prec k$. We now prove that $c_k = 0$.

3 We define the event $x^{(2)} \in \Omega$, such that for some $k \in U$

$$x_i^{(2)} = \begin{cases} 1 & i \in R \setminus R(k) \\ 0 & i \in R(k) \end{cases}$$

Then

$$p(x^{(2)}, \alpha) = \left(\bar{\alpha_k} + \alpha_k \phi_k\right) \prod_{i \in U \setminus \{V(k)\}} \alpha_i \tag{A.10}$$

where $\phi_k = \prod_{l \in d(k)} \bar{\beta_l}$ with $\bar{\beta_l} = P_\alpha[X_i = 0 \; \forall \; i \in R(k) | X_k = 1]$.
Following from our assumption that we have already proved $c_j = 0 \;\; \forall \;\; j \prec k$ and from equations A.4,vA.6 and A.10, we have

$$-\frac{c_k(\phi_k - 1)}{\bar{\alpha_k} + \alpha_k \phi_k} + \sum_{i \in U \setminus V(k)} \frac{c_i}{\alpha_i} = 0 \text{ and} \tag{A.11}$$

$$\frac{c_k}{\alpha_k} + \sum_{i \in U \setminus V(k)} \frac{c_i}{\alpha_i} = 0 \tag{A.12}$$

Combining these last two equations, we find $c_k = 0$.

The proof for an inverse multicast tree is similar on account of the reversibility property.

A.2 General coding scheme on a tree

The proof is written simultaneous with an example network with a 'generalised coding scheme'. The approach is to enumerate the events wherein for every link $e = (j_0, k_0)$, $S(j_0)$ does not reach $S(k_0)$. Consider the configuration in Figure A.1.

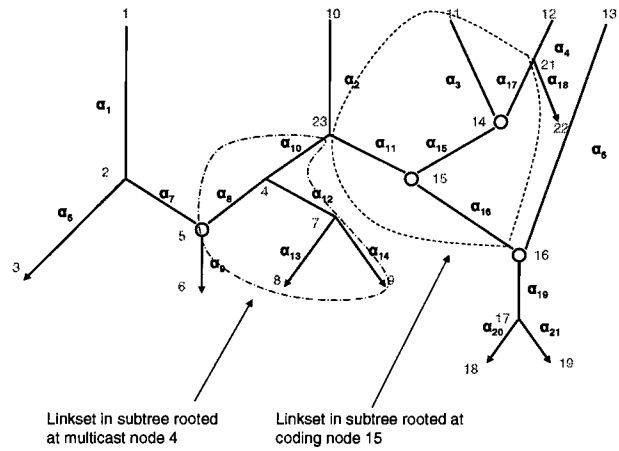


Figure A.1: A 21 link tree with a generalised coding scheme applied to it. We prove that the corresponding 21x21 Fisher is non-singular.

(A.12)

We set out to prove that the corresponding 21x21 Fisher matrix of this network is non-singular.

1. The network is denoted by $G = (V, E)$.
2. V_m is the set of multicast nodes, V_c is the set of coding nodes.
3. A link from node j_0 to k_0 is denoted by $e = (j_0, k_0)$ and has link probability α_e.
4. $\alpha \in (0,1)^{|E|}$.
5. Note that for any node k, the set of receivers $R(k)$ are connected to the sources in the source subset of k, $S(k)$.
6. The subtree rooted at node 'k' is $T(k) = (V(k), E(k))$ defined such that: (i) When k is a multicast node, $V(k)$ is the set of *multicast* nodes which are descendants of k, $V(k) = \{l \in \{V \cap V_m\} : l \preceq k\}k\}$ and $E(k)$ is the set of all links incident to $V(k)$ (ii) When k is a coding node, $V(k)$ is the set of *coding* nodes which are ancestors of k, $V(k) = \{l \in \{V \cap V_c\} : l \succeq k\}$ and $E(k)$ is the set of all links incident to $V(k)$. This is illustrated in Figure A.1.

Again, as before, a supposition that $I(\alpha)$ is singular for some $\alpha = (\alpha_e)_{e \in E} \in (0,1)^{|E|}$ would imply A.4, reproduced here with slightly different notation in accordance with the definitions just above.

$$\sum_{e \in E} c_e \frac{\partial \log p(x, \alpha)}{\partial \alpha_e} = 0 \tag{A.14}$$

We now have to show that equation A.14 implies $c_e = 0$ first for the e's which are leaves, and then for all $e \in E$.

We have $p(x^{(0)}, \alpha) = \prod_{e' \in E} \alpha_i$, which from equation A.14, results in

$$\sum_{e=1}^{21} \frac{c_e}{\alpha_e} = 0 \tag{A.15}$$

We now systematically show that $c_e = 0 \ \forall \ e \in E$. **Remember, the link under consideration will be denoted as $e = (j_0, k_0)$.**

1. *For leaf links from a source to a coding node* : $j_0 \in S, k_0 \in V_c$

   links : 3 and 5 : These are leaves. The event wherein only this link fails and all other links function is called $x^{(1)}$. In the case of link 3, this means that S_3 does not reach any receiver. The probability of this event is

$$p(x^{(1)}; \alpha) = \bar{\alpha_3} \prod_{e' \in E \setminus e_3} \alpha_{e'} \tag{A.16}$$

$$-\frac{c_3}{\bar{\alpha_3}} + \sum_{i \in E \setminus \{3\}} \frac{c'_e}{\alpha_{e'}} = 0 \quad \{\text{from equation A.14}\} \tag{A.17}$$

   Combining this with equation A.15, we have $c_3 = 0$. Similarly, it can be proven that $c_5 = 0$.

2. *For leaf links from a multicast node to a receiver* : $j_0 \in V_m$ $k_0 \in R$

   links : 6, 13, 14, 18, 20, 21 : These are all leaves as well. The event wherein only this link fails and all others function is called $x^{(2)}$. In the case of link 6, this means that the receiver k_0 (node 3) does not receive a packet from $S(k_0)$ (source node 1), while all other receivers in the receiver set of source 1 do. In this case, the only other receiver is node 6. The probability of this event is

$$p(x^{(2)};\alpha) = \bar{\alpha_6} \prod_{e' \in E \setminus \{6\}} \alpha_{e'} \tag{A.18}$$

$$-\frac{c_6}{\bar{\alpha_6}} + \sum_{i \in E \setminus \{6\}} \frac{c'_e}{\alpha_{e'}} = 0 \quad \{\text{from equation A.14}\} \tag{A.19}$$

   Combining this with equation A.15, we have $c_6 = 0$. Similarly, it can be proven that $c_{13}, c_{14}, c_{18}, c_{20} and c_{21} = 0$.

3. *For links from a source or a multicast node to a multicast node* : $j_0 \in V_m$ or $j_0 \in S, k_0 \in V_m$

   link 12 : here, $j_0 \in V_m, k_0 \in V_m$

   Proving $c_{12} = 0$ requires c_{13} and c_{14} to be 0. We have proved this above.

   Consider the event $x^{(3)}$ wherein sources in $S(j_0)$ (namely, source 10) do not reach $R(k_0)$ (namely, receivers 8 and 9), but reaches all other receivers in the receiver set of j_0, $R(j_0)$ (namely, receiver 6). The probability of this event is

$$p(x^{(3)};\alpha) = \left(\bar{\alpha_e} + \alpha_e \phi_{k_0}\right) \prod_{e' \in E \setminus E(k_0)} \alpha_{e'} \tag{A.20}$$

$$\text{where } \phi_{k_0} = \prod_{l \in d(k_0)} \bar{\beta_l} \tag{A.21}$$

   In this case,

$$p(x^{(3)};\alpha) = \left(\bar{\alpha_{12}} + \alpha_{12} \phi_{k_0}\right) \prod_{e' \in E \setminus \{12,13,14\}} \alpha_{e'} \tag{A.22}$$

$$\text{and } \phi_{k_0} = \bar{\alpha_{13}} \bar{\alpha_{14}} \tag{A.23}$$

   This and equation A.15 alongwith $c_{13} = c_{14} = 0$ give

$$-\frac{c_e(\phi_{k_0} - 1)}{\bar{\alpha_e} + \alpha_e \phi_{k_0}} + \sum_{e' \in E \setminus E(k_0)} \frac{c'_e}{\alpha'_e} = 0 \text{ which in this case is}$$

$$-\frac{c_{12}(\phi_{k_0} - 1)}{\bar{\alpha_{12}} + \alpha_{12} \phi_{k_0}} + \sum_{e' \in E \setminus \{12,13,14\}} \frac{c'_e}{\alpha'_e} = 0 \text{ and} \tag{A.24}$$

$$\frac{c_e}{\alpha_e} + \sum_{e' \in E \setminus E(k_0)} \frac{c'_e}{\alpha'_e} = 0 \text{ which in this case is}$$

$$\frac{c_{12}}{\alpha_{12}} + \sum_{e' \in E \setminus \{12,13,14\}} \frac{c'_e}{\alpha'_e} = 0 \tag{A.25}$$

Combining these two equations gives $c_{12} = 0$.

4. *For links from a multicast node to a coding node* : $j_0 \in V_m, k_0 \in V_c$

   links : 7, 8, 11, 17 : $j_0 \in V_m$, $k_0 \in V_C$

   Consider the event $x^{(4)}$ wherein only one of these links fails and all others function. Here, the sources in $S(j_0)$ do not reach the receivers in $R(k_0)$. The probability of this event in the case of link 7 is

$$p(x^{(4)};\alpha) = \bar{\alpha_7} \prod_{e' \in E \setminus \{7\}} \alpha'_e \tag{A.26}$$

   from equation A.14, we have

$$-\frac{c_7}{\bar{\alpha_7}} + \sum_{i \in E \setminus \{7\}} \frac{c'_e}{\alpha_{e'}} = 0 \tag{A.27}$$

   Combining the above with equation A.15, we get $c_7 = 0$. Similarly, c_8, c_{11} and c_{17} are 0.

5. *For links from a source or multicast node to a multicast node* : $j_0 \in V_m$ or $j_0 \in S, k_0 \in V_m$

   links : 10, 1, 2 and 4 : First consider link 10. Now $x^{(3)}$ was defined as the event wherein sources in $S(j_0)$ (namely, source node 10) do not reach $R(k_0)$ (namely, receiver nodes 6, 8 and 9), but reaches all other receivers in the receiver set of j_0, $R(j_0)$ (namely, receiver nodes 18 and 19). The probability of this event is

$$p(x^{(3)};\alpha) = \left(\bar{\alpha_e} + \alpha_e \phi_{k_0}\right) \prod_{e' \in E \setminus E(k_0)} \alpha_{e'} \tag{A.28}$$

$$\text{where } \phi_{k_0} = \prod_{l \in d(k_0)} \bar{\beta_l} \tag{A.29}$$

   and $E(k_0)$ is the linkset in the subtree rooted at k_0 (node 4 here) having only *multicasting* nodes as shown in the figure. In this case,

$$p(x^{(3)};\alpha) = \left(\bar{\alpha_{10}} + \alpha_{10} \phi_{k_0}\right) \prod_{e' \in \{E \setminus \{10,8,12,13,14\}\}} \alpha_{e'} \tag{A.30}$$

$$\text{and } \phi_{k_0} = \bar{\alpha_8}(\bar{\alpha_{12}} + \alpha_{12}\bar{\alpha_{13}}\bar{\alpha_{14}}) \tag{A.31}$$

   Note: $\left(\bar{\alpha_{10}} + \alpha_{10}\phi_{k_0}\right)$ is the probability that a loss occurs within the linkset $E(k_0)$ contained in the subtree rooted at j_0 (node 4 here), where this subtree is defined such that it contains only multicasting nodes.

   The above equation and equation A.14, along with $c_8 = c_{12} = c_{13} = c_{14} = 0$ give

$$-\frac{c_{10}(\phi_{k_0} - 1)}{\bar{\alpha_{10}} + \alpha_{10}\phi_{k_0}} + \sum_{e' \in E \setminus \{10,8,12,13,14\}} \frac{c'_e}{\alpha'_e} = 0 \text{ and} \tag{A.32}$$

$$\frac{c_{10}}{\alpha_{10}} + \sum_{e' \in E \setminus \{10,8,12,13,14\}} \frac{c'_e}{\alpha'_e} = 0 \tag{A.33}$$

   which results in $c_{10} = 0$.

   Similarly, it can be proved that $c_1 = c_2 = c_4 = 0$.

6. *For leaf links from a source to a coding node* : $j_0 \in V_c, k_0 \in V_c$ or $k_0 \in R$

   links 15, 16 and 9 :

   Link 16 is of particular interest, but to prove $c_{16} = 0$, we first need to prove $c_{15} = 0$, hence, consider link 15 and look at an event $x^{(5)}$ such that the sources in $S(j_0)$ (namely, source nodes 11 and 12) do not reach the receivers in $R(k_0)$ (namely receiver nodes 18 and 19). The event probability is

$$p(x^{(5)};\alpha) = (\bar{\alpha_{15}} + \alpha_{15}\eta_{j_0}) \prod_{e' \in \{E \setminus \{15,3,17\}\}} \alpha_{e'} \quad \text{(A.34)}$$

$$\text{and } \eta_{j_0} = \bar{\alpha_3}\bar{\alpha_{17}} \quad \text{(A.35)}$$

   Using the procedure followed so far, $c_{15} = 0$.

   We now consider link 16 and look at an event $x^{(5)}$ such that the sources in $S(j_0)$ (namely, source nodes 10, 11 and 12) do not reach the receivers in $R(k_0)$ (namely receiver nodes 18 and 19). The event probability is

$$p(x^{(5)};\alpha) = (\bar{\alpha_e} + \alpha_e\eta_{j_0}) \prod_{e' \in \{E \setminus E(j_0)} \alpha_{e'} \quad \text{(A.36)}$$

$$\text{where } \eta_{j_0} = \prod_{l \in f(j_0)} \bar{\mu_l} \quad \text{(A.37)}$$

   and $E(j_0)$ is the linkset in the subtree rooted at j_0 (node 15 here) having only *coding* nodes as shown in the figure.

   In this case,

$$p(x^{(5)};\alpha) = (\bar{\alpha_{16}} + \alpha_{16}\eta_{j_0}) \prod_{e' \in \{E \setminus \{16,11,15,3,17\}\}} \alpha_{e'} \quad \text{(A.38)}$$

$$\text{and } \eta_{j_0} = \bar{\alpha_{11}}(\bar{\alpha_{15}} + \alpha_{15}(\bar{\alpha_3}\bar{\alpha_{17}})) \quad \text{(A.39)}$$

   Note: $(\bar{\alpha_{16}} + \alpha_{16}\eta_{j_0})$ is the probability that a loss occurs within the linkset $E(j_0)$ contained in the subtree rooted at j_0 (node 15 here), where this subtree is defined such that it contains only coding nodes. The above equation and equation A.14, alongwith $c_{11} = c_{15} = c_3 = c_{17} = 0$ give

$$-\frac{c_{16}(\eta_{j_0} - 1)}{\bar{\alpha_{16}} + \alpha_{16}\eta_{j_0}} + \sum_{e' \in E \setminus \{16,11,15,3,17\}} \frac{c'_e}{\alpha'_e} = 0 \text{ and} \quad \text{(A.40)}$$

$$\frac{c_{16}}{\alpha_{16}} + \sum_{e' \in E \setminus \{16,11,15,3,17\}} \frac{c'_e}{\alpha'_e} = 0 \quad \text{(A.41)}$$

   which results in $c_{16} = 0$. Similarly, $c_9 = 0$.

7. *For links from a coding node to a multicast node* : $j_0 \in V_c, k_0 \in V_m$

   link 19 :

Finally, we consider an event $x^{(6)}$ wherein sources in $S(j_0)$ (namely, source nodes 10, 11, 12 1nd 13) do not reach receivers in $R(k_0)$ (namely receiver nodes 18 and 19). The event probability is

$$p(x^{(6)};\alpha) = \left[\bar{\alpha_e} + \alpha_e[1-(1-\prod_{l\in f(j_0)}\bar{\mu_l})(1-\prod_{m\in d(k_0)}\bar{\beta_m})]\right]\prod_{e'\setminus\{E(k_0)\bigcup E(j_0)\}}\alpha'_e \quad \text{(A.43)}$$

which in this case is

$$\begin{aligned} p(x^{(6)};\alpha) &= (\bar{\alpha_{19}} + \alpha_{19}[1-[1-(\bar{\alpha_{16}} + \alpha_{16}(\bar{\alpha_{11}}(\bar{\alpha_{15}} + \alpha_{15}\bar{\alpha_3}\bar{\alpha_{17}})))][1-\bar{\alpha_5}]] \\ &* [1-\bar{\alpha_{20}}\bar{\alpha_{21}}]\prod_{e'\in E\setminus\{E(k_0)\bigcup E(j_0)\}}\alpha'_e \end{aligned} \quad \text{(A.44)}$$

As demonstrated previously, when $p(x^{(6)})$ is used in this form in equation A.14, and alongwith equation A.15, this leads to $c_{19} = 0$.

And this concludes the proof of the non-singularity of I.

Bibliography

[1] R. Caceres, N. G. Duffield, J. Horowitz and D. Towsley, "Multicastbased inference of network-internal loss characteristics. *in IEEE Trans. in Inf. Theory, vol. 45, pp. 24622480, 1999.*

[2] C. Fragouli and A. Markopoulou, "A Network Coding approach to Overlay in Network Monitoring." *in Proc.of 43rd Allerton, September*, 2005.

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

[4] Jean-Yves Le Boudec, "Performance evaluation of computer and communication systems." (book to be published)

[5] Mark Coates, Robert Noak, "Network loss inference using Unicast end-to-end measurement." *in ITC Conference on IP Traffic, Modeling and Management, March 15, 2000.*

[6] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow." *in 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." *in IEEE Trans. on Inf. Theory, Vol. 49, Feb. 2003.*

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

[9] 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.*

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

[11] T.Cover,J.Thomas, "Elements of Information Theory" *in Wiley 1991.*

[12] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm." in *IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 498519, Feb 2001.*

[13] Y. Mao, F. R. Kschischang, B. Li, S. Pasupathy, "A Factor Graph Approach to Link Loss Monitoring in Wireless Sensor Networks" in *IEEE Journal on selected areas in communications, vol. 23, no. 4, april 2005.*