Efficient Operation of Coded Packet Networks

by

Desmond S. Lun

B.Sc., University of Melbourne (2001)
B.E. (Hons.), University of Melbourne (2001)
S.M., Massachusetts Institute of Technology (2002)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author .....
Department of Electrical Engineering and Computer Science
May 15, 2006

Certified by .....
Muriel Médard
Esther and Harold Edgerton Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Efficient Operation of Coded Packet Networks

by

Desmond S. Lun

Submitted to the Department of Electrical Engineering and Computer Science
on May 15, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

A fundamental problem faced in the design of almost all packet networks is that of efficient operation—of reliably communicating given messages among nodes at minimum cost in resource usage. We present a solution to the efficient operation problem for coded packet networks, i.e., packet networks where the contents of outgoing packets are arbitrary, causal functions of the contents of received packets. Such networks are in contrast to conventional, routed packet networks, where outgoing packets are restricted to being copies of received packets and where reliability is provided by the use of retransmissions.

This thesis introduces four considerations to coded packet networks:

1. efficiency,
2. the lack of synchronization in packet networks,
3. the possibility of broadcast links, and
4. packet loss.

We take these considerations and give a prescription for operation that is novel and general, yet simple, useful, and extensible.

We separate the efficient operation problem into two smaller problems, which we call network coding—the problem of deciding what coding operation each node should perform given the rates at which packets are injected on each link—and subgraph selection—the problem of deciding those rates. Our main contribution for the network coding problem is to give a scheme that achieves the maximum rate of a multicast connection under the given injection rates. As a consequence, the separation of network coding and subgraph selection results in no loss of optimality provided that we are constrained to only coding packets within a single connection. Our main contribution for the subgraph selection problem is to give distributed algorithms that optimally solve the single-connection problem under certain assumptions. Since the scheme we propose for network coding can easily be implemented in a distributed manner, we obtain, by combining the solutions for each of the smaller problems, a distributed approach to the efficient operation problem.

We assess the performance of our solution for three problems: minimum-transmission wireless unicast, minimum-weight wireline multicast, and minimum-energy wireless multicast. We find that our solution has the potential to offer significant efficiency improvements over existing techniques in routed packet networks, particularly for multi-hop wireless networks.

Thesis Supervisor: Muriel Médard
Title: Esther and Harold Edgerton Associate Professor of Electrical Engineering

Preface

Vladimir Nabokov once opined, "My loathings are simple: stupidity, oppression, crime, cruelty, soft music. My pleasures are the most intense known to man: writing and butterfly hunting." I share all of Nabokov's loathings, but only one of his pleasures—and that began only recently. Of course, the lepidoptera I've been involved with are none that Nabokov would recognize or, I imagine, much revere. Nevertheless, the butterflies to which I refer—from the butterfly network of Ahlswede et al. (see Figure 7 of [2]) to its wireless counterpart (see Figure 1 of [73]) to further generalizations—have certainly given me a great deal of pleasure since I began investigating network coding in the spring of 2003.

This thesis represents the culmination of my work over the last three years, which began with the simple question, how would all this actually work? I was intrigued by network coding. But I couldn't quite reconcile it with the way that I understood data networks to operate. So I thought to take the basic premise of network coding and put it in a model that, at least to me, was more satisfying. The following pages lay out a view of coded packet networks that, while certainly not the only one possible, is one that I believe is simple, relevant, and extensible—I can only hope that it is sufficiently so to be truly useful.

Various parts of the work in this thesis appear in various published papers [29, 70, 71, 72, 73, 74, 76, 77, 78] and various as yet unpublished papers [75, 79]. A brief glance at the author lists of the these papers, and it is evident that I cannot claim

sole credit for this work—many others are involved.

My adviser, Professor Muriel Médard, is foremost among them. I would like to thank her for all that she has taught me and all that she has done to aid my development—both professional and personal. The way that she manages the multitude of demands on her time continues to amaze and inspire me. I would like to thank also my thesis readers, Professors Michelle Effros, Ralf Koetter, and John Tsitsiklis. All have contributed helpful discussions and advice. I would like to thank Ralf in particular, as he has served almost as a second adviser to me. His insight and enthusiasm have been invaluable.

Various others have contributed to various parts of the work, and I wish to acknowledge them for it: Niranjan Ratnakar (Section 3.2), Dr. Payam Pakzad (Section 2.4), Dr. Christina Fragouli (Section 2.4), Professor David Karger (Section 3.3), Professor Tracey Ho (Section 3.1), Ebad Ahmed (Sections 4.2 and 4.3), Fang Zhao (Sections 4.2 and 4.3), and Hyunjoo Lee (Section 4.2). All have been a delight to work with. I am grateful also to Guy Weichenberg and Ed Schofield for their helpful comments on early drafts of the manuscript.

On a personal level, there are many to thank, but I will keep it brief. I am aware of Friedrich Nietzsche's maxim, "Ein Mensch mit Genie ist unausstehlich, wenn er nicht mindestens noch zweierlei dazu besitzt: Dankbarkeit und Reinlichkeit." [A man with spirit is unbearable if he does not have at least two other things: gratitude and cleanliness.] And, while I shan't discuss my cleanliness, I certainly don't wish any of my friends or family to feel that I am not grateful for the favor they have shown me. I am. But I want to keep this to those to whom I am really indebted the most: Mum, Dad, Guy, and Katie. I love you all.

Desmond S. Lun

Cambridge, Mass.

April 2006

This research was supported by the National Science Foundation under grant nos.

CCR-0093349 and CCR-0325496; by the Army Research Office through University of California subaward no. S0176938; and by the Office of Naval Research under grant no. N00014-05-1-0197.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A FUNDAMENTAL problem faced in the design of almost all packet networks is that of efficient operation—of reliably communicating given messages among nodes at minimum cost in resource usage. At present, the problem is generally addressed in the following way: messages admitted into the network are put into packets that are routed hop-by-hop toward their destinations according to paths chosen to meet the goal of efficiency, e.g., to achieve low energy consumption, to achieve low latency, or, more generally, to incur low cost of any sort. As packets travel along these paths, they are occasionally lost because of various reasons, which include buffer overflow, link outage, and collision; so, to ensure reliability, retransmissions of unacknowledged packets are sent either on a link-by-link basis, an end-to-end basis, or both. This mode of operation crudely characterizes the operation of the internet and has held sway since at least its advent.

But much has changed about packet networks since the advent of the internet. The underlying communications technologies have changed, as have the types of services demanded, and, under these changes, the mode of operation described above has met with difficulties. We give two examples. First, while wireline communications were once dominant in packet networks, wireless communications involving nodes on the ground, in the air, in space, and even underwater are now increasingly

prevalent. In networks where such wireless links are present, this mode of operation can certainly be made to work, but we encounter problems—most notably with the use of retransmissions. Wireless links are highly unreliable compared to wireline ones and are sometimes associated with large propagation delays, which means that, not only are more retransmissions required, but packet acknowledgments are themselves sometimes lost or subject to large delay, leading to substantial inefficiencies from the retransmission of unacknowledged packets. Moreover, hop-by-hop routing fails to exploit the inherent broadcast nature often present in wireless links, leading to further inefficiencies.

Second, while unicast services were once the norm, multicast services are now required for applications such as file distribution and video-conferencing. For multicast services, hop-by-hop routing means routing over a tree, which is difficult to do efficiently—finding the minimum-cost tree that spans a multicast group equates to solving the Steiner tree problem, which is a well-known NP-complete problem [16, 105]. Moreover, if there are many receivers, many retransmitted packets may be needed, placing an unnecessary load on the network and possibly overwhelming the source. Even if the source manages, packets that are retransmitted are often useful only to a subset of the receivers and redundant to the remainder.

The problems we mentioned can and generally have been resolved to some degree by ad hoc methods and heuristics. But that is hardly satisfactory—not only from an intellectual standpoint, since ad hoc solutions do little for our understanding of the fundamental problem, but also from a practical standpoint, since they tend to lead to complex, inefficient designs that are more art than science. Indeed, as Robert G. Gallager has commented, "much of the network field is an art [rather than a science]" [41]. And while it is evident that engineering real-world systems is an activity that will always lie between an art and a science, it is also evident that the more we base our designs on scientific principles, the better they will generally be.

In this thesis, therefore, we eschew such "routed" packet networks altogether in fa-

vor of a new approach: we consider coded packet networks—generalizations of routed packet networks where the contents of outgoing packets are arbitrary, causal functions of the contents of received packets. In this context, we consider the same fundamental problem, i.e., we ask, how do we operate coded packet networks efficiently?

We present a prescription for the operation of coded packet networks that, in certain scenarios (e.g., in multi-hop wireless networks), yields significant efficiency improvements over what is achievable in routed packet networks. We begin, in Section 1.1, by discussing coded packet networks in more detail and by clarifying the position of our work, then, in Section 1.2, we describe our network model. We outline the body of the thesis in Section 1.3.

1.1 Coded packet networks

The basic notion of network coding, of performing coding operations on the contents of packets throughout a network, is generally attributed to Ahlswede et al. [2]. Ahlswede et al. never explicitly mentioned the term "packet" in [2], but their network model, which consists of nodes interconnected by error-free point-to-point links, implies that the coding they consider occurs above channel coding and, in a data network, is presumably applied to the contents of packets.

Still, their work is not the first to consider coding in such a network model. Earlier instances of work with such a network model include those by Han [45] and Tsitsiklis [103]. But the work of Ahlswede et al. is distinct in two ways: First, Ahlswede et al. consider a new problem—multicast. (The earlier work considers the problem of transmitting multiple, correlated sources from a number of nodes to a single node.) Second, and more importantly, the work of Ahlswede et al. was quickly followed by other work, by Li et al. [64] and by Koetter and Médard [62], that showed that codes with a simple, linear structure were sufficient to achieve capacity in the multicast problem. This result put structure on the codes and gave hope that practicable capacity-achieving codes could be found.

The subsequent growth in network coding was explosive. Practicable capacity-achieving codes were quickly proposed by Jaggi et al. [54], Ho et al. [50], and Fragouli and Soljanin [40]. Applications to network management [49], network tomography [38, 47], overlay networks [43, 55, 116], and wireless networks [44, 60, 94, 110, 111] were studied; capacity in random networks [88], undirected networks [66, 67], and Aref networks [91] was studied; security aspects were studied [17, 20, 36, 48, 53]; the extension to non-multicast problems was studied [32, 58, 82, 90, 92, 93]; and further code constructions based on convolutional codes and other notions were proposed [25, 34, 35, 39, 46, 63]. Most notoriously, network coding has been adopted as a core technology of Microsoft's Avalanche project [43]—a research project that aims to develop a peer-to-peer file distribution system, which may be in competition with existing systems such as BitTorrent.

Of the various work on network coding, we draw particular attention to the code construction by Ho et al. [50]. Their construction is very simple: they proposed that every node construct its linear code randomly and independently of all other nodes, and, while random linear codes were not new (the study of random linear codes dates as early as the work of Elias [33] in the 1950s), the application of such codes to the network multicast problem was. Some years earlier, Luby [69] searched for codes for the transmission of packets over a lossy link and discovered random linear codes, constructed according to a particular distribution, with remarkable complexity properties. This work, combined with that of Ho et al., led to a resurgence of interest in random linear codes (see, e.g., [1, 25, 30, 81, 85, 96]) and to the recognition of a powerful technique that we shall exploit extensively: random linear coding on packets.

The work we have described has generally focused on coding and capacity—growing, as it has, from coding theory and information theory—and has been removed from networking theory, which generally focuses on notions such as efficiency and quality of service. While it is adequate, and indeed appropriate, to start in this way, it is clear that, with network coding being concerned with communication net-

works, topics under the purview of networking theory must eventually be broached.

This thesis makes an attempt. It introduces four considerations absent from the original work of Ahlswede et al.: First, we consider efficiency by defining a cost for inefficiency. This is a standard framework in networking theory, which is used, e.g., in the optimal routing problem (see, e.g., [13, Sections 5.4–5.7]). Second, we consider the lack of synchronization in packet networks, i.e., we allow packet injections and receptions on separate links to occur at completely different rates with arbitrary degrees of correlation. Third, we consider the possibility of broadcast links, i.e., we allow links in the network to reach more than one node, capturing one of the key characteristics of wireless networks. Fourth, we consider packet loss, i.e., we allow for the possibility that packets are not received at the end or ends of the link into which they are injected.

Some of these considerations are present in other, concurrent work. For example, efficiency is also considered in [28, 111]; and the possibility of broadcast links and packet loss are also considered in [44, 60]. These papers offer alternative solutions to special cases of the problem that we tackle. We take all four considerations and give a prescription for operation that is novel and general, yet simple, useful, and extensible.

1.2 Network model

We set out, in this section, to present our network model. The intent of the model is to capture heterogeneous networks composed of wireline and wireless links that may or may not be subject to packet losses. Thus, the model captures a wide variety of networks, affording us a great degree of generality.

But that is not to say that we believe that coding should be applied to all networks. There is a common concern about the wisdom of doing coding in packet networks since coding, being a more complicated operation than routing, increases the computational load on nodes, which are often already overtaxed in this regard. Indeed, in high-speed

optical networks, bottlenecks are caused almost exclusively by processing at nodes rather than by transmission along links [21, 86]. But high-speed optical networks are certainly not the only type of network of interest, and there are others where coding seems more immediately applicable. Two such types of networks are application-level overlay networks and multi-hop wireless networks—in both cases, having coding capability at nodes is feasible, and we expect bottlenecks to originate from links rather than nodes.

We represent the topology of the network with a directed hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of hyperarcs. A *hypergraph* is a generalization of a graph, where, rather than arcs, we have hyperarcs. A *hyperarc* is a pair (i, J), where i, the start node, is an element of $\mathcal{N}$ and J, the set of end nodes, is a non-empty subset of $\mathcal{N}$.

Each hyperarc (i, J) represents a broadcast link from node i to nodes in the non-empty set J. In the special case where J consists of a single element j, we have a point-to-point link. The hyperarc is now a simple arc and we sometimes write (i, j) instead of $(i, \{j\})$. The link represented by hyperarc (i, J) may be lossless or lossy, i.e., it may or may not be subject to packet erasures.

To establish the desired connection or connections, packets are injected on hyperarcs. Let A_{iJ} be the counting process describing the arrival of packets that are injected on hyperarc (i, J), and let A_{iJK} be the counting process describing the arrival of packets that are injected on hyperarc (i, J) and received by exactly the set of nodes $K \subset J$; i.e., for $\tau \geq 0$, $A_{iJ}(\tau)$ is the total number of packets that are injected on hyperarc (i, J) between time 0 and time τ, and $A_{iJK}(\tau)$ is the total number of packets that are injected on hyperarc (i, J) and received by all nodes in K (and no nodes in $\mathcal{N} \setminus K$) between time 0 and time τ. For example, suppose that three packets are injected on hyperarc $(1, \{2, 3\})$ between time 0 and time τ_0 and that, of these three packets, one is received by node 2 only, one is lost entirely, and one is received by both nodes 2 and 3; then we have $A_{1(23)}(\tau_0) = 3$, $A_{1(23)\emptyset}(\tau_0) = 1$, $A_{1(23)2}(\tau_0) = 1$,

$A_{1(23)3}(\tau_0) = 0$, and $A_{1(23)(23)}(\tau_0) = 1$. We have $A_{1(23)2}(\tau_0) = 1$ not $A_{1(23)2}(\tau_0) = 2$ because, while two packets are received by node 2, only one is received by exactly node 2 and no other nodes. Similarly, we have $A_{1(23)3}(\tau_0) = 0$ not $A_{1(23)3}(\tau_0) = 1$ because, while one packet is received by node 3, none are received by exactly node 3 and no other nodes.

We assume that A_{iJ} has an average rate z_{iJ} and that A_{iJK} has an average rate z_{iJK}; more precisely, we assume that

$$\lim_{\tau\to\infty} \frac{A_{iJ}(\tau)}{\tau} = z_{iJ}$$

and that

$$\lim_{\tau\to\infty} \frac{A_{iJK}(\tau)}{\tau} = z_{iJK}$$

almost surely. Hence, we have $z_{iJ} = \sum_{K\subset J} z_{iJK}$ and, if the link is lossless, we have $z_{iJK} = 0$ for all $K \subsetneq J$.

The vector z, consisting of z_{iJ}, $(i, J) \in \mathcal{A}$, defines the rate at which packets are injected on all hyperarcs in the network, and we assume that it must lie within some constraint set Z. Thus, the pair $(\mathcal{H}, Z)$ defines a capacitated graph that represents the network at our disposal, which may be a full, physical network or a subnetwork of a physical network. The vector z, then, can be thought of as a subset of this capacitated graph—it is the portion actually under use—and we call it the *coding subgraph* for the desired connection or connections. For the time being, we make no assumptions about Z except that it is a convex subset of the positive orthant containing the origin. This assumption leaves room for Z to take complicated forms; and indeed it does, particularly when the underlying physical network is a wireless network, where transmissions on one link generally interfere with those on others. For examples of forms that Z may take in wireless networks, see [27, 56, 57, 61, 111, 114].

We associate with the network a convex cost function f that maps feasible coding subgraphs to real numbers and that we seek to minimize. This cost function

might represent, e.g., energy consumption, average latency, monetary cost, or a combination of these considerations. We assume convexity primarily for simplicity and tractability. Certainly, cases where f is non-convex may still be tractable, but proving general results is difficult. We expect, at any rate, that most cost functions of interest will indeed be convex, and this is generally true of cost functions representing the considerations that we have mentioned.

With this set-up, the objective of the *efficient operation problem* is to establish a set of desired connections at specified rates at minimum cost. This is the problem we address.

As the following example will illustrate, the problem we have defined is certainly non-trivial. Nevertheless, its scope is limited: we consider rate, or throughput, to be the sole factor that is explicitly important in determining the quality of a connection, and we consider the rates of packet injections on hyperarcs (i.e., the coding subgraph) to be the sole factor that contributes to its cost. Rate is frequently the most important factor under consideration, but there are others. For example, memory usage, computational load, and delay are often also important factors. At present, we unfortunately do not have a clean way to consider such factors. We discuss the issue further in Section 2.4 and Chapter 5.

1.2.1 An example

We refer to this example as the *slotted Aloha relay channel*, and we shall return to it throughout the thesis. This example serves to illustrate some of the capabilities of our approach, especially as they relate to the issues of broadcast and interference in multi-hop wireless networks.

One of most important issues in multi-hop wireless networks is medium access, i.e., determining how radio nodes share the wireless medium. A simple, yet popular, method for medium access control is slotted Aloha (see, e.g., [13, Section 4.2]), where nodes with packets to send follow simple random rules to determine when they trans-

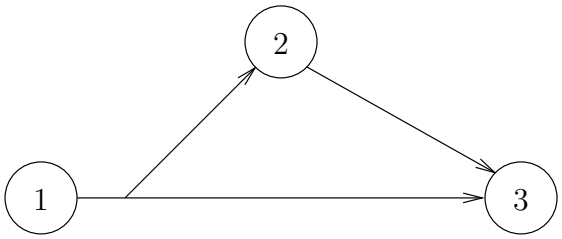

Figure 1.1: The slotted Aloha relay channel.

mit. In this example, we consider a multi-hop wireless network using slotted Aloha for medium access control.

We suppose that the network has the simple topology shown in Figure 1.1 and that, in this network, we wish to establish a single unicast connection of rate R from node 1 to node 3. The random rule we take for transmission is that the two transmitting nodes, node 1 and node 2, each transmit packets independently in a given time slot with some fixed probability. In coded packet networks, nodes are never "unbacklogged" as they are in regular, routed slotted Aloha networks—nodes can transmit coded packets whenever they are given the opportunity. Hence $z_{1(23)}$, the rate of packet injection on hyperarc $(1, \{2, 3\})$, is the probability that node 1 transmits a packet in a given time slot, and likewise z_{23}, the rate of packet injection on hyperarc $(2, 3)$, is the probability that node 2 transmits a packet in a given time slot. Therefore, $Z = [0, 1]^2$, i.e., $0 \leq z_{1(23)} \leq 1$ and $0 \leq z_{23} \leq 1$.

If node 1 transmits a packet and node 2 does not, then the packet is received at node 2 with probability $p_{1(23)2}$, at node 3 with probability $p_{1(23)3}$, and at both nodes 2 and 3 with probability $p_{1(23)(23)}$ (it is lost entirely with probability $1 - p_{1(23)2} - p_{1(23)3} - p_{1(23)(23)}$). If node 2 transmits a packet and node 1 does not, then the packet is received at node 3 with probability p_{233} (it is lost entirely with probability $1 - p_{233}$). If both nodes 1 and 2 each transmit a packet, then the packets collide and neither of the packets is received successfully anywhere.

It is possible that simultaneous transmission does not necessarily result in collision, with one or more packets being received. This phenomenon is referred to as

multipacket reception capability [42] and is decided by lower-layer implementation details. In this example, however, we simply assume that simultaneous transmission results in collision.

Hence, we have

$$z_{1(23)2} = z_{1(23)}(1 - z_{23})p_{1(23)2}, \tag{1.1}$$

$$z_{1(23)3} = z_{1(23)}(1 - z_{23})p_{1(23)3}, \tag{1.2}$$

$$z_{1(23)(23)} = z_{1(23)}(1 - z_{23})p_{1(23)(23)}, \tag{1.3}$$

and

$$z_{233} = (1 - z_{1(23)})z_{23}p_{233}. \tag{1.4}$$

We suppose that our objective is to set up the desired connection while minimizing the total number of packet transmissions for each message packet, perhaps for the sake of energy conservation or conservation of the wireless medium (to allow it to be used for other purposes, such as other connections). Therefore

$$f(z_{1(23)}, z_{23}) = z_{1(23)} + z_{23}.$$

The slotted Aloha relay channel is very similar to the relay channel introduced by van der Meulen [104], and determining the capacity of the latter is one of the famous, long-standing, open problems of information theory. The slotted Aloha relay channel is related to the relay channel (hence its name), but different. While the relay channel relates to the physical layer, we are concerned with higher layers, and our problem is ultimately soluble. Whether our solution has any bearing on the relay channel is an interesting issue that remains to be explored.

We return to the slotted Aloha relay channel in Sections 2.2.3 and 3.1.1.

1.3 Thesis outline

The main contribution of this thesis is to lay out, for coded packet networks conforming to our model, a solution to the efficient operation problem that we have posed, namely, the problem of establishing a set of desired connections at specified rates at minimum cost. This solution is contained in Chapters 2 and 3.

Chapter 2 looks at the problem of determining what coding operation each node should perform given the coding subgraph. We propose using a particular random linear coding scheme that we show can establish a single multicast connection at rates arbitrarily close to its capacity in a given coding subgraph. This means that, at least for establishing a single multicast connection, there is no loss of optimality in using this coding scheme and determining the coding subgraph independently. The optimality to which we refer is with respect to the efficient operation problem that we have defined, which, as we have mentioned, does not explicitly consider factors such as memory usage, computational load, and delay. In Section 2.4, we include memory usage as a factor under explicit consideration. We modify the coding scheme to reduce the memory usage of intermediate nodes and assess, by analysis and computer simulation, the effect of this modification on various performance factors.

Chapter 3, on the other hand, looks at the problem of determining the coding subgraph. We argue that, even when we wish to establish multiple connections, it suffices, in many instances, simply to use the coding scheme described in Chapter 2 and to determine the coding subgraph independently. Thus, this problem, of determining the coding subgraph, can be written as a mathematical programming problem, and, under particular assumptions, we find distributed algorithms for performing the optimization. We believe that these algorithms may eventually form the basis for protocols used in practice.

In Chapter 4, we evaluate, by computer simulation, the performance of the solution we laid out and compare it to the performance of existing techniques for routed packet networks. We find that our solution has the potential to offer significant efficiency

improvements, particularly for multi-hop wireless networks. For some readers, this chapter may be the one to read first. It can be understood more or less independently of Chapters 2 and 3 and is, in a sense, "the bottom line"—at least in so far as we have managed to elucidate it. The interested reader may then proceed to Chapters 2 and 3 to understand the solution we propose.

Our conclusion, in Chapter 5, gives a final perspective on our work and discusses the road ahead.

Chapter 2

Network Coding

This chapter deals with what we call the network coding part of the efficient operation problem. We assume that the coding subgraph z is given, and we set out to determine what coding operation each node should perform. We propose using a particular random linear coding scheme that we show can establish a single multicast connection at rates arbitrarily close to its capacity in z. More precisely, for a given coding subgraph z, which gives rise to a particular set of rates $\{z_{iJK}\}$ at which packets are received, the coding scheme we study achieves (within an arbitrarily small factor) the maximum possible throughput when run for a sufficiently long period of time. Exactly how the injection rates defined by z relates to the reception rates $\{z_{iJK}\}$ and how the losses, which establishes this relationship, are caused is immaterial for our result—thus, losses may be due to collisions, link outage, buffer overflow, or any other process that gives rise to losses. The only condition that we require the losses to satisfy is that they give rise to packet receptions where the average rates $\{z_{iJK}\}$ exist, as our network model specifies (see Section 1.2).

As a consequence of the result, in establishing a single multicast connection in a network, there is no loss of optimality in the efficient operation problem from separating subgraph selection and network coding. We deal with subgraph selection in Chapter 3.

We begin, in Section 2.1, by precisely specifying the coding scheme we consider then, in Section 2.2, we give our main result: that this scheme can establish a single multicast connection at rates arbitrarily close to its capacity in z. In Section 2.3, we strengthen these results in the special case of Poisson traffic with i.i.d. losses by giving error exponents. These error exponents allow us to quantify the rate of decay of the probability of error with coding delay and to determine the parameters of importance in this decay.

In both these sections, we consider rate, or throughput, of the desired connection to be the sole factor of explicit importance. In Section 2.4, we include memory usage as a factor of explicit importance. We modify the coding scheme to reduce the memory usage of intermediate nodes, and we study the effect of this modification.

2.1 Coding scheme

The specific coding scheme we consider is as follows. We suppose that, at the source node, we have K message packets $w_1, w_2, \ldots, w_K$, which are vectors of length λ over some finite field $\mathbb{F}_q$. (If the packet length is b bits, then we take $\lambda = \lceil b/\log_2 q \rceil$.) The message packets are initially present in the memory of the source node.

The coding operation performed by each node is simple to describe and is the same for every node: received packets are stored into the node's memory, and packets are formed for injection with random linear combinations of its memory contents whenever a packet injection occurs on an outgoing link. The coefficients of the combination are drawn uniformly from $\mathbb{F}_q$.

Since all coding is linear, we can write any packet u in the network as a linear combination of $w_1, w_2, \ldots, w_K$, namely, $u = \sum_{k=1}^{K} \gamma_k w_k$. We call γ the *global encoding vector* of u, and we assume that it is sent along with u, as side information in its header. The overhead this incurs (namely, $K \log_2 q$ bits) is negligible if packets are sufficiently large.

Nodes are assumed to have unlimited memory. The scheme can be modified so

that received packets are stored into memory only if their global encoding vectors are linearly-independent of those already stored. This modification keeps our results unchanged while ensuring that nodes never need to store more than K packets. The case where nodes can only store fewer than K packets is discussed in Section 2.4.

A sink node collects packets and, if it has K packets with linearly-independent global encoding vectors, it is able to recover the message packets. Decoding can be done by Gaussian elimination. The scheme can be run either for a predetermined duration or, in the case of rateless operation, until successful decoding at the sink nodes. We summarize the scheme in Figure 2.1.

The scheme is carried out for a single block of K message packets at the source. If the source has more packets to send, then the scheme is repeated with all nodes flushed of their memory contents.

Related random linear coding schemes are described in [25, 50] for the application of multicast over lossless wireline packet networks, in [30] for data dissemination, and in [1] for data storage. Other coding schemes for lossy packet networks are described in [44] and [60]; the scheme described in the former requires placing in the packet headers side information that grows with the size of the network, while that described in the latter requires no side information at all, but achieves lower rates in general. Both of these coding schemes, moreover, operate in a block-by-block manner, where coded packets are sent by intermediate nodes only after decoding a block of received packets—a strategy that generally incurs more delay than the scheme we describe, where intermediate nodes perform additional coding yet do not decode [85].

2.2 Coding theorems

In this section, we specify achievable rate intervals for the coding scheme in various scenarios. The fact that the intervals we specify are the largest possible (i.e., that the scheme is capacity-achieving) can be seen by simply noting that the rate of a connection must be limited by the rate at which distinct packets are being received

Initialization:

- The source node stores the message packets $w_1, w_2, \ldots, w_K$ in its memory.

Operation:

- When a packet is received by a node,
  - the node stores the packet in its memory.
- When a packet injection occurs on an outgoing link of a node,
  - the node forms the packet from a random linear combination of the packets in its memory. Suppose the node has L packets $u_1, u_2, \ldots, u_L$ in its memory. Then the packet formed is

$$u_0 := \sum_{l=1}^{L} \alpha_l u_l,$$

    where α_l is chosen according to a uniform distribution over the elements of $\mathbb{F}_q$. The packet's global encoding vector γ, which satisfies $u_0 = \sum_{k=1}^{K} \gamma_k w_k$, is placed in its header.

Decoding:

- Each sink node performs Gaussian elimination on the set of global encoding vectors from the packets in its memory. If it is able to find an inverse, it applies the inverse to the packets to obtain $w_1, w_2, \ldots, w_K$; otherwise, a decoding error occurs.

Figure 2.1: Summary of the random linear coding scheme we consider.

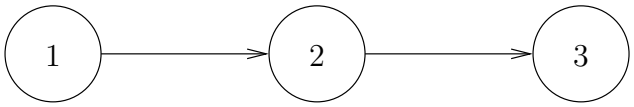

Figure 2.2: A network consisting of two point-to-point links in tandem.

over any cut between the source and the sink. A formal converse can be obtained using the cut-set bound for multi-terminal networks (see [26, Section 14.10]).

2.2.1 Unicast connections

We develop our general result for unicast connections by extending from some special cases. We begin with the simplest non-trivial case: that of two point-to-point links in tandem (see Figure 2.2).

Suppose we wish to establish a connection of rate arbitrarily close to R packets per unit time from node 1 to node 3. Suppose further that the coding scheme is run for a total time Δ, from time 0 until time Δ, and that, in this time, a total of N packets is received by node 2. We call these packets $v_1, v_2, \ldots, v_N$.

Any packet u received by a node is a linear combination of $v_1, v_2, \ldots, v_N$, so we can write

$$u = \sum_{n=1}^{N} \beta_n v_n.$$

Now, since v_n is formed by a random linear combination of the message packets $w_1, w_2, \ldots, w_K$, we have

$$v_n = \sum_{k=1}^{K} \alpha_{nk} w_k$$

for $n = 1, 2, \ldots, N$. Hence

$$u = \sum_{k=1}^{K} \left(\sum_{n=1}^{N} \beta_n \alpha_{nk} \right) w_k,$$

and it follows that the kth component of the global encoding vector of u is given by

$$\gamma_k = \sum_{n=1}^{N} \beta_n \alpha_{nk}.$$

We call the vector β associated with u the *auxiliary encoding vector* of u, and we see that any node that receives $\lfloor K(1+\varepsilon) \rfloor$ or more packets with linearly-independent auxiliary encoding vectors has $\lfloor K(1+\varepsilon) \rfloor$ packets whose global encoding vectors collectively form a random $\lfloor K(1+\varepsilon) \rfloor \times K$ matrix over $\mathbb{F}_q$, with all entries chosen uniformly. If this matrix has rank K, then node 3 is able to recover the message packets. The probability that a random $\lfloor K(1+\varepsilon) \rfloor \times K$ matrix has rank K is, by a simple counting argument, $\prod_{k=1+\lfloor K(1+\varepsilon) \rfloor - K}^{\lfloor K(1+\varepsilon) \rfloor} (1 - 1/q^k)$, which can be made arbitrarily close to 1 by taking K arbitrarily large. Therefore, to determine whether node 3 can recover the message packets, we essentially need only to determine whether it receives $\lfloor K(1+\varepsilon) \rfloor$ or more packets with linearly-independent auxiliary encoding vectors.

Our proof is based on tracking the propagation of what we call *innovative* packets. Such packets are innovative in the sense that they carry new, as yet unknown, information about $v_1, v_2, \ldots, v_N$ to a node. It turns out that the propagation of innovative packets through a network follows the propagation of jobs through a queueing network, for which fluid flow models give good approximations. We present the following argument in terms of this fluid analogy and defer the formal argument to Appendix 2.A.1 at the end of this chapter.

Since the packets being received by node 2 are the packets $v_1, v_2, \ldots, v_N$ themselves, it is clear that every packet being received by node 2 is innovative. Thus, innovative packets arrive at node 2 at a rate of z_{122}, and this can be approximated by fluid flowing in at rate z_{122}. These innovative packets are stored in node 2's memory, so the fluid that flows in is stored in a reservoir.

Packets, now, are being received by node 3 at a rate of z_{233}, but whether these packets are innovative depends on the contents of node 2's memory. If node 2 has more

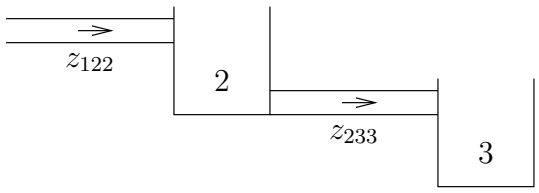

Figure 2.3: Fluid flow system corresponding to two-link tandem network.

information about $v_1, v_2, \ldots, v_N$ than node 3 does, then it is highly likely that new information will be described to node 3 in the next packet that it receives. Otherwise, if node 2 and node 3 have the same degree of information about $v_1, v_2, \ldots, v_N$, then packets received by node 3 cannot possibly be innovative. Thus, the situation is as though fluid flows into node 3's reservoir at a rate of z_{233}, but the level of node 3's reservoir is restricted from ever exceeding that of node 2's reservoir. The level of node 3's reservoir, which is ultimately what we are concerned with, can equivalently be determined by fluid flowing out of node 2's reservoir at rate z_{233}.

We therefore see that the two-link tandem network in Figure 2.2 maps to the fluid flow system shown in Figure 2.3. It is clear that, in this system, fluid flows into node 3's reservoir at rate $\min(z_{122}, z_{233})$. This rate determines the rate at which packets with new information about $v_1, v_2, \ldots, v_N$—and, therefore, linearly-independent auxiliary encoding vectors—arrive at node 3. Hence the time required for node 3 to receive $\lfloor K(1+\varepsilon) \rfloor$ packets with linearly-independent auxiliary encoding vectors is, for large K, approximately $K(1+\varepsilon)/\min(z_{122}, z_{233})$, which implies that a connection of rate arbitrarily close to R packets per unit time can be established provided that

$$R \leq \min(z_{122}, z_{233}). \tag{2.1}$$

The right-hand side of (2.1) is indeed the capacity of the two-link tandem network, and we therefore have the desired result for this case.

We extend our result to another special case before considering general unicast connections: we consider the case of a tandem network consisting of L point-to-point

Figure 2.4: A network consisting of L point-to-point links in tandem.

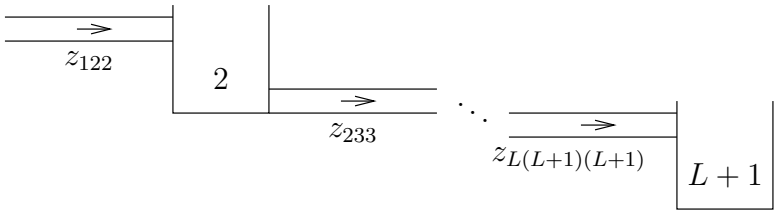

Figure 2.5: Fluid flow system corresponding to L-link tandem network.

links and $L+1$ nodes (see Figure 2.4).

This case is a straightforward extension of that of the two-link tandem network. It maps to the fluid flow system shown in Figure 2.5. In this system, it is clear that fluid flows into node $(L+1)$'s reservoir at rate $\min_{1\le i\le L}\{z_{i(i+1)(i+1)}\}$. Hence a connection of rate arbitrarily close to R packets per unit time from node 1 to node $L+1$ can be established provided that

$$R \le \min_{1\le i\le L}\{z_{i(i+1)(i+1)}\}. \tag{2.2}$$

Since the right-hand side of (2.2) is indeed the capacity of the L-link tandem network, we therefore have the desired result for this case. A formal argument is in Appendix 2.A.2.

We now extend our result to general unicast connections. The strategy here is simple: A general unicast connection can be formulated as a flow, which can be decomposed into a finite number of paths. Each of these paths is a tandem network, which is the case that we have just considered.

Suppose that we wish to establish a connection of rate arbitrarily close to R packets per unit time from source node s to sink node t. Suppose further that

$$R \le \min_{Q\in\mathcal{Q}(s,t)} \left\{ \sum_{(i,J)\in\Gamma_+(Q)} \sum_{K\not\subset Q} z_{iJK} \right\},$$

where $\mathcal{Q}(s,t)$ is the set of all cuts between s and t, and $\Gamma_+(Q)$ denotes the set of forward hyperarcs of the cut Q, i.e.,

$$\Gamma_+(Q) := \{(i,J) \in \mathcal{A} \,|\, i \in Q, J \setminus Q \neq \emptyset\}.$$

Therefore, by the max-flow/min-cut theorem (see, e.g., [4, Sections 6.5–6.7], [10, Section 3.1]), there exists a flow vector x satisfying

$$\sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi} = \begin{cases} R & \text{if } i = s, \\ -R & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i \in \mathcal{N}$,

$$\sum_{j\in K} x_{iJj} \leq \sum_{\{L\subset J | L\cap K\neq\emptyset\}} z_{iJL} \tag{2.3}$$

for all $(i,J) \in \mathcal{A}$ and $K \subset J$, and $x_{iJj} \geq 0$ for all $(i,J) \in \mathcal{A}$ and $j \in J$.

Using the conformal realization theorem (see, e.g., [10, Section 1.1]), we decompose x into a finite set of paths $\{p_1, p_2, \ldots, p_M\}$, each carrying positive flow R_m for $m = 1, 2, \ldots, M$, such that $\sum_{m=1}^{M} R_m = R$. We treat each path p_m as a tandem network and use it to deliver innovative packets at rate arbitrarily close to R_m, resulting in an overall rate for innovative packets arriving at node t that is arbitrarily close to R. Some care must be take in the interpretation of the flow and its path decomposition because the same packet may be received by more than one node. The details of the interpretation are in Appendix 2.A.3

2.2.2 Multicast connections

The result for multicast connections is, in fact, a straightforward extension of that for unicast connections. In this case, rather than a single sink t, we have a set of sinks T. As in the framework of static broadcasting (see [97, 98]), we allow sink

nodes to operate at different rates. We suppose that sink $t \in T$ wishes to achieve rate arbitrarily close to R_t, i.e., to recover the K message packets, sink t wishes to wait for a time Δ_t that is only marginally greater than K/R_t. We further suppose that

$$R_t \leq \min_{Q \in \mathcal{Q}(s,t)} \left\{ \sum_{(i,J) \in \Gamma_+(Q)} \sum_{K \not\subset Q} z_{iJK} \right\}$$

for all $t \in T$. Therefore, by the max-flow/min-cut theorem, there exists, for each $t \in T$, a flow vector $x^{(t)}$ satisfying

$$\sum_{\{j|(i,J) \in \mathcal{A}\}} \sum_{j \in J} x^{(t)}_{iJj} - \sum_{\{j|(j,I) \in \mathcal{A}, i \in I\}} x^{(t)}_{jIi} = \begin{cases} R & \text{if } i = s, \\ -R & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i \in \mathcal{N}$,

$$\sum_{j \in K} x^{(t)}_{iJj} \leq \sum_{\{L \subset J | L \cap K \neq \emptyset\}} z_{iJL}$$

for all $(i, J) \in \mathcal{A}$ and $K \subset J$, and $x^{(t)}_{iJj} \geq 0$ for all $(i, J) \in \mathcal{A}$ and $j \in J$.

For each flow vector $x^{(t)}$, we go through the same argument as that for a unicast connection, and we find that the probability of error at every sink node can be made arbitrarily small by taking K sufficiently large.

We summarize our results with the following theorem statement.

Theorem 2.1. *Consider the coding subgraph z. The random linear coding scheme described in Section 2.1 is capacity-achieving for multicast connections in z, i.e., for K sufficiently large, it can achieve, with arbitrarily small error probability, a multicast connection from source node s to sink nodes in the set T at rate arbitrarily close to*

R_t packets per unit time for each $t \in T$ if

$$R_t \leq \min_{Q \in \mathcal{Q}(s,t)} \left\{ \sum_{(i,J) \in \Gamma_+(Q)} \sum_{K \not\subset Q} z_{iJK} \right\}$$

for all $t \in T$.[1]

Remark. The capacity region is determined solely by the average rates $\{z_{iJK}\}$ at which packets are received. Thus, the packet injection and loss processes, which give rise to the packet reception processes, can in fact take any distribution, exhibiting arbitrary correlations, as long as these average rates exist.

2.2.3 An example

We return to the slotted Aloha relay channel described in Section 1.2.1. Theorem 2.1 implies that the random linear coding scheme we consider can achieve the desired unicast connection at rates arbitrarily close to R packets per unit time if

$$R \leq \min(z_{1(23)2} + z_{1(23)3} + z_{1(23)(23)}, z_{1(23)3} + z_{1(23)(23)} + z_{233}).$$

Substituting (1.1)–(1.4), we obtain

$$\begin{aligned} R \leq \min(&z_{1(23)}(1 - z_{23})(p_{1(23)2} + p_{1(23)3} + p_{1(23)(23)}), \\ &z_{1(23)}(1 - z_{23})(p_{1(23)3} + p_{1(23)(23)}) + (1 - z_{1(23)})z_{23}p_{233}). \end{aligned}$$

We see that the range of achievable rates is specified completely in terms of the parameters we control, $z_{1(23)}$ and z_{23}, and the given parameters of the problem, $p_{1(23)2}$, $p_{1(23)3}$, $p_{1(23)(23)}$, and p_{233}. It remains only to choose $z_{1(23)}$ and z_{23}. This, we deal with in the next chapter.

[1] In earlier versions of this work [70, 76], we required the field size q of the coding scheme to approach infinity for Theorem 2.1 to hold. This requirement is in fact not necessary, and the formal arguments in Appendix 2.A do not require it.

2.3 Error exponents for Poisson traffic with i.i.d. losses

We now look at the rate of decay of the probability of error p_e in the coding delay Δ. In contrast to traditional error exponents where coding delay is measured in symbols, we measure coding delay in time units—time $\tau = \Delta$ is the time at which the sink nodes attempt to decode the message packets. The two methods of measuring delay are essentially equivalent when packets arrive in regular, deterministic intervals.

We specialize to the case of Poisson traffic with i.i.d. losses. Thus, the process A_{iJK} is a Poisson process with rate z_{iJK}. Consider the unicast case for now, and suppose we wish to establish a connection of rate R. Let C be the supremum of all asymptotically-achievable rates.

We begin by deriving an upper bound on the probability of error. To this end, we take a flow vector x from s to t of size C and, following the development in Appendix 2.A, develop a queueing network from it that describes the propagation of innovative packets for a given innovation order μ. This queueing network now becomes a Jackson network. Moreover, as a consequence of Burke's theorem (see, e.g., [59, Section 2.1]) and the fact that the queueing network is acyclic, the arrival and departure processes at all stations are Poisson in steady-state.

Let $\Psi_t(m)$ be the arrival time of the mth innovative packet at t, and let $C' := (1 - q^{-\mu})C$. When the queueing network is in steady-state, the arrival of innovative packets at t is described by a Poisson process of rate C'. Hence we have

$$\lim_{m\to\infty} \frac{1}{m} \log \mathbb{E}[\exp(\theta \Psi_t(m))] = \log \frac{C'}{C' - \theta} \tag{2.4}$$

for $\theta < C'$ [14, 87]. If an error occurs, then fewer than $\lceil R\Delta \rceil$ innovative packets are received by t by time $\tau = \Delta$, which is equivalent to saying that $\Psi_t(\lceil R\Delta \rceil) > \Delta$. Therefore,

$$p_e \leq \Pr(\Psi_t(\lceil R\Delta \rceil) > \Delta),$$

and, using the Chernoff bound, we obtain

$$p_e \leq \min_{0\leq\theta<C'} \exp\left(-\theta\Delta + \log \mathbb{E}[\exp(\theta\Psi_t(\lceil R\Delta\rceil))]\right).$$

Let ε be a positive real number. Then using equation (2.4) we obtain, for Δ sufficiently large,

$$\begin{aligned} p_e &\leq \min_{0\leq\theta<C'} \exp\left(-\theta\Delta + R\Delta\left\{\log\frac{C'}{C'-\theta} + \varepsilon\right\}\right) \\ &= \exp(-\Delta(C' - R - R\log(C'/R)) + R\Delta\varepsilon). \end{aligned}$$

Hence, we conclude that

$$\lim_{\Delta\to\infty} \frac{-\log p_e}{\Delta} \geq C' - R - R\log(C'/R). \tag{2.5}$$

For the lower bound, we examine a cut whose flow capacity is C. We take one such cut and denote it by Q^*. It is clear that, if fewer than $\lceil R\Delta\rceil$ distinct packets are received across Q^* in time $\tau = \Delta$, then an error occurs. The arrival of distinct packets across Q^* is described by a Poisson process of rate C. Thus we have

$$\begin{aligned} p_e &\geq \exp(-C\Delta) \sum_{l=0}^{\lceil R\Delta\rceil-1} \frac{(C\Delta)^l}{l!} \\ &\geq \exp(-C\Delta)\frac{(C\Delta)^{\lceil R\Delta\rceil-1}}{\Gamma(\lceil R\Delta\rceil)}, \end{aligned}$$

and, using Stirling's formula, we obtain

$$\lim_{\Delta\to\infty} \frac{-\log p_e}{\Delta} \leq C - R - R\log(C/R). \tag{2.6}$$

Since (2.5) holds for all positive integers μ, we conclude from (2.5) and (2.6) that

$$\lim_{\Delta\to\infty} \frac{-\log p_e}{\Delta} = C - R - R\log(C/R). \tag{2.7}$$

Equation (2.7) defines the asymptotic rate of decay of the probability of error in the coding delay Δ. This asymptotic rate of decay is determined entirely by R and C. Thus, for a packet network with Poisson traffic and i.i.d. losses employing the coding scheme described in Section 2.1, the flow capacity C of the minimum cut of the network is essentially the sole figure of merit of importance in determining the effectiveness of the coding scheme for large, but finite, coding delay. Hence, in deciding how to inject packets to support the desired connection, a sensible approach is to reduce our attention to this figure of merit, which is indeed the approach that we take in Chapter 3.

Extending the result from unicast connections to multicast connections is straightforward—we simply obtain (2.7) for each sink.

2.4 Finite-memory random linear coding

The results that we have thus far established about the coding scheme described in Section 2.1 show that, from the perspective of conveying the most information in each packet transmission, it does very well. But packet transmissions are not the only resource with which we are concerned. Other resources that may be scarce include memory and computation and, if these resources are as important or more important than packet transmissions, then a natural question is whether we can modify the coding scheme of Section 2.1 to reduce its memory and computation requirements, possibly in exchange for more transmissions.

In this section, we study a simple modification. We take the coding scheme of Section 2.1, and we assume that intermediate nodes (i.e., nodes that are neither source nor sink nodes) have memories capable only of storing a fixed, finite number of packets, irrespective of K. An intermediate node with a memory capable of storing M packets uses its memory in one of two ways:

1. as a shift register: arriving packets are stored in memory and, if the memory is

already full, the oldest packet in the memory is discarded; or

2. as an accumulator: arriving packets are multiplied by a random vector chosen uniformly over $\mathbb{F}_q^M$, and the product is added to the M memory slots.

We first consider, in Section 2.4.1, the case of a single intermediate node in isolation. In this case, the intermediate node encodes packets and its immediate downstream node decodes them. Such a scheme offers an attractive alternative to comparable reliability schemes for a single link, such as automatic repeat request (ARQ) or convolutional coding (see, e.g., [5, 6]). In Section 2.4.2, we consider a network, specifically, the two-link tandem network (see Figure 2.2). We see that, while limiting the memory of intermediate nodes certainly results in loss of achievable rate, the relative rate loss, at least for the two-link tandem network, can be quantified, and it decays exponentially in M.

2.4.1 Use in isolation

When used in isolation at a single intermediate node, the encoder takes an incoming stream of message packets, $u_1, u_2, \ldots$, and forms a coded stream of packets that is placed on its lossy outgoing link and decoded on reception. We assume that the decoder knows, for each received packet, the linear transformation that has been performed on the message packets to yield that packet. This information can be communicated to the decoder by a variety of means, which include placing it into the header of each packet as described in Section 2.1 (which is certainly viable when the memory is used as a shift register—the overhead is $M \log_2 q$ bits plus that of a sequence number), and initializing the random number generators at the encoder and decoder with the same seed.

The task of decoding, then, equates to matrix inversion in $\mathbb{F}_q$, which can be done straightforwardly by applying Gaussian elimination to each packet as it is received. This procedure produces an approximately-steady stream of decoded packets with an

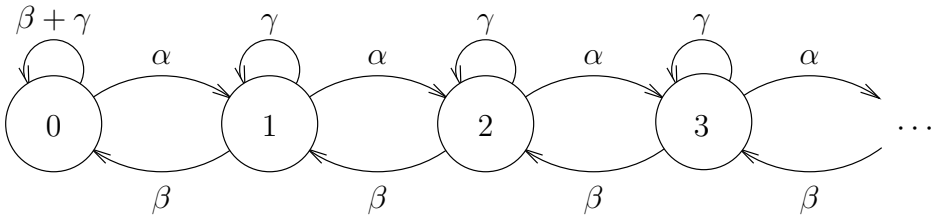

Figure 2.6: Markov chain modeling the evolution of the difference between the number of packets received by the encoder and the number of packets transmitted and not lost.

expected delay that is constant in the length of the input stream. Moreover, if the memory is used as a shift register, then the complexity of this decoding procedure is also constant with the length of the input stream and, on average, is $O(M^2)$ per packet.

We discretize the time axis into epochs that correspond to the transmission of an outgoing packet. Thus, in each epoch, an outgoing packet is transmitted, which may be lost, and one or more incoming packets are received. If transmission is to be reliable, then the average number of incoming packets received in each epoch must be at most one.

We make the following assumptions on incoming packet arrivals and outgoing packet losses, with the understanding that generalizations are certainly possible. We assume that, in an epoch, a single packet arrives independently with probability r and no packets arrive otherwise, and the transmitted outgoing packet is lost independently with probability ε and is received otherwise. This model is appropriate when losses and arrivals are steady—and not bursty.

We conduct our analysis in the limit of $q \to \infty$, i.e., the limit of infinite field size. We later discuss how the analysis may be adapted for finite q, and quantify by simulation the difference between the performance in the case of finite q and that of infinite q in some particular instances.

We begin by considering the difference between the number of packets received by the encoder and the number of packets transmitted and not lost. This quantity, we

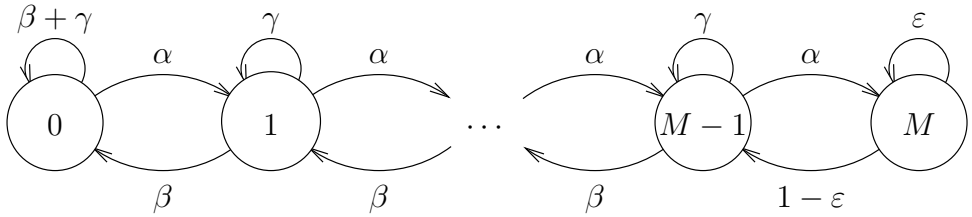

Figure 2.7: Markov chain modeling the behavior of the coding scheme in the limit of $q \to \infty$.

see, evolves according to the infinite-state Markov chain shown in Figure 2.6, where $\alpha = r\varepsilon$, $\beta = (1-r)(1-\varepsilon)$, and $\gamma = r(1-\varepsilon) + (1-r)\varepsilon$.

At the first epoch, the memory of the encoder is empty and we are in state 0. We continue to remain in state 0 in subsequent epochs until the first packet u_1 arrives. Consider the first outgoing packet after the arrival of u_1. This packet is either lost or not. Let us first suppose that it is not lost. Thus, we remain in state 0, and the decoder receives a packet that is a random linear combination of u_1, i.e., a random scalar multiple of u_1, and, since q is infinitely large by assumption, this scalar multiple is non-zero with probability 1; so the decoder can recover u_1 from the packet that it receives.

Now suppose instead that the first outgoing packet after the arrival of u_1 is lost. Thus, we move to state 1. If an outgoing packet is transmitted and not lost before the next packet arrives, the decoder again receives a random scalar multiple of u_1 and we return to state 0. So suppose we are in state 1 and u_2 arrives. Then, the next outgoing packet is a random linear combination of u_1 and u_2. Suppose further that this packet is received by the decoder, so we are again in state 1. This packet, currently, is more or less useless to the decoder; it represents a mixture between u_1 and u_2 and does not allow us to determine either. Nevertheless, it gives, with probability 1, the decoder some information that it did not previously know, namely, that u_1 and u_2 lie in a particular linear subspace of $\mathbb{F}_q^2$. As in Section 2.2, we call such an informative packet *innovative*.

Any subsequent packet received by the decoder is also innovative with probability

1. In particular, if the decoder receives a packet before the arrival of another packet u_3 at the encoder, returning us to state 0, then the decoder is able to recover both u_1 and u_2. More generally, what we see is that, provided that packets arrive only in states $0, 1, \ldots, M-1$, the decoder is able to recover, at every return to state 0, the packets that arrived between the current and the previous return. If a packet arrives in state M, however, loss occurs. Information in the encoder's memory is overwritten or corrupted, and will never be recovered. The current contents of the encoder's memory, however, can still be recovered and, from the point of view of recovering these contents, the coding system behaves as though we were in state M. Hence, to analyze the performance of the coding scheme, we modify the Markov chain shown in Figure 2.6 to that in Figure 2.7. Let x_t be the state of this Markov chain at time t. We can interpret x_t as the number of innovative packets the encoder has for sending at time t.

We now proceed to derive some quantities that are useful for designing the parameters of the coding scheme. We begin with the steady-state probabilities $\pi_i := \lim_{t\to\infty} \Pr(x_t = i)$. Since $\{x_t\}$ is a birth-death process, its steady-state probabilities are readily obtained. We obtain

$$\pi_i = \frac{\varrho^i(1-\varrho)}{1-\sigma\varrho^M} \tag{2.8}$$

for $i = 0, 1, \ldots, M-1$, and

$$\pi_M = \frac{\varepsilon\sigma\varrho^{M-1}(1-\varrho)}{1-\sigma\varrho^M}, \tag{2.9}$$

where $\varrho := \alpha/\beta = r\varepsilon/(1-r)(1-\varepsilon)$ and $\sigma := r/(1-\varepsilon)$. We assume $\varrho < 1$, which is equivalent to $r < 1-\varepsilon$, for, if not, the capacity of the outgoing link is exceeded, and we cannot hope for the coding scheme to be effective.

We now derive the probability of packet loss, p_l. Evaluating p_l is not straightforward because, since coded packets depend on each other, the loss of a packet owing

to the encoder exceeding its memory is usually accompanied by other packet losses. We derive an upper bound on the probability of loss.

A packet is successfully recovered by the decoder if the ensuing path taken in the Markov chain in Figure 2.7 returns to state 0 without a packet arrival occurring in state M. Let q_i be the probability that a path, originating in state i, reaches state 0 without a packet arrival occurring in state M. Our problem is very similar to a random walk, or ruin, problem (see, e.g., [37, Chapter XIV]). We obtain

$$q_i = \frac{1 - \sigma\varrho^{M-i}}{1 - \sigma\varrho^M}$$

for $i = 0, 1, \ldots, M$.

Now, after the coding scheme has been running for some time, a random arriving packet finds the scheme in state i with probability π_i and, with probability $1-\varepsilon$, the scheme returns to state i after the next packet transmission or, with probability ε, it moves to state $i+1$. Hence

$$\begin{aligned}
1 - p_l &\geq \sum_{i=0}^{M-1} \{(1-\varepsilon)q_i + \varepsilon q_{i+1}\}\pi_i \\
&= \sum_{i=0}^{M-1} \left\{(1-\varepsilon)\frac{1-\sigma\varrho^{M-i}}{1-\sigma\varrho^M} + \varepsilon\frac{1-\sigma\varrho^{M-i-1}}{1-\sigma\varrho^M}\right\}\frac{\varrho^i(1-\varrho)}{1-\sigma\varrho^M} \\
&= \frac{1-\varrho}{(1-\sigma\varrho^M)^2}\left\{\frac{1-\varrho^M}{1-\varrho} - (1-\varepsilon)M\sigma\varrho^M - \varepsilon M\sigma\varrho^{M-1}\right\} \\
&= \frac{1}{(1-\sigma\varrho^M)^2}\{1 - \varrho^M - (1-2\varepsilon)M\sigma\varrho^M - \varepsilon M\sigma\varrho^{M-1} + (1-\varepsilon)M\sigma\varrho^{M+1}\},
\end{aligned}$$

from which we obtain

$$p_l \leq \frac{\varrho^{M-1}}{(1-\sigma\varrho^M)^2}\{\varepsilon M\sigma + (1 - 2\sigma + M\sigma - 2\varepsilon M\sigma)\varrho - (1-\varepsilon)M\sigma\varrho^2 + \sigma^2\varrho^{M+1}\}. \quad (2.10)$$

We have thus far looked at the limit of $q \to \infty$, while, in reality, q must be finite. There are two effects of having finite q: The first is that, while the encoder may have

innovative information to send to the decoder (i.e., $x_t > 0$), it fails to do so because the linear combination it chooses is not linearly independent of the combinations already received by the decoder. For analysis, we can consider such non-innovative packets to be equivalent to erasures, and we find that the effective erasure rate is $\varepsilon(1 - q^{-x_t})$. The Markov chain in Figure 2.7 can certainly be modified to account for this effective erasure rate, but doing so makes analysis much more tedious.

The second of the effects is that, when a new packet arrives, it may not increase the level of innovation at the encoder. When the memory is used as a shift register, this event arises because a packet is overwritten before it has participated as a linear factor in any successfully received packets, i.e., all successfully received packets have had a coefficient of zero for that packet. When the memory is used as an accumulator, this event arises because the random vector chosen to multiply the new packet is such that the level of innovation remains constant. The event of the level of innovation not being increased by a new packet can be quite disastrous, because it is effectively equivalent to the encoder exceeding its memory. Fortunately, the event seems rare; in the accumulator case, we can quantify the probability of the event exactly as $1 - q^{x_t - M}$.

To examine the effect of finite q, we chose $\varepsilon = 0.1$ and simulated the performance of the coding scheme for 200,000 packets with various choices of the parameters r, q, and M (see Figures 2.8–2.11). We decoded using Gaussian elimination on packets as they were received and used the encoder's memory as a shift register to keep decoding complexity constant with the length of the packet stream. Delay was evaluated as the number of epochs between a packet's arrival at the encoder and it being decoded, neglecting transmission delay. As expected, we see that average loss rate decreases and average delay increases with increasing M; a larger memory results, in a sense, in more coding, which gives robustness at the expense of delay. Moreover, we see that a field size $q \geq 2^8$ (perhaps even $q \geq 2^4$) is adequate for attaining loss rates close to the upper bound for infinite field size.

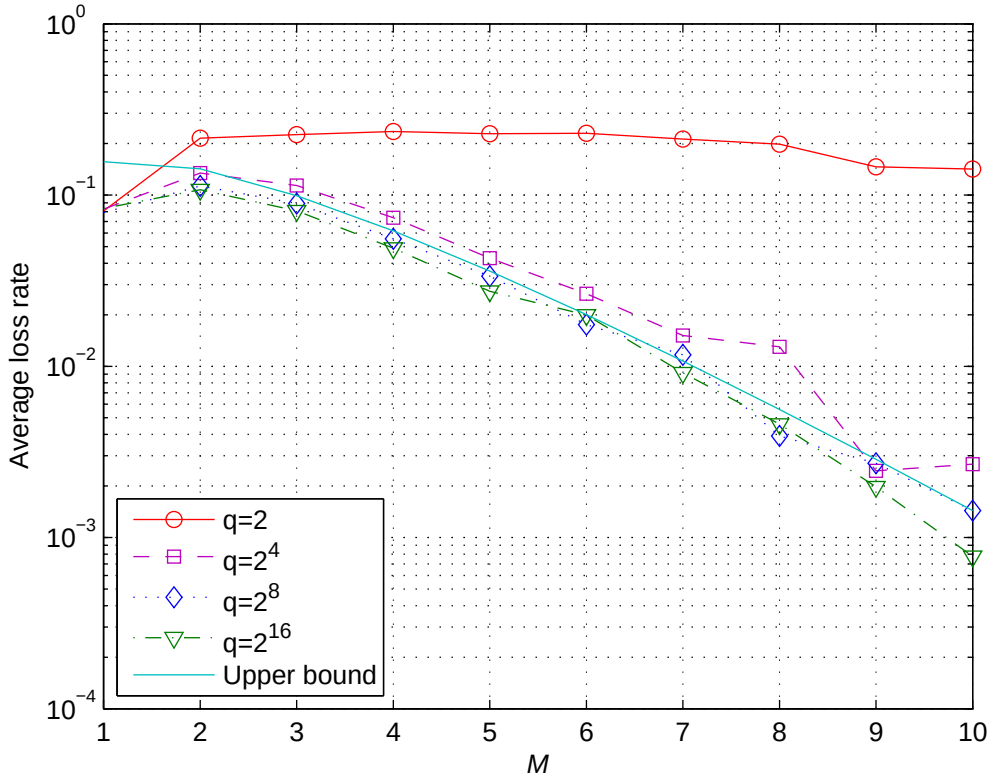

Figure 2.8: Average loss rate for 200,000 packets as a function of memory size M with $r = 0.8$, $\varepsilon = 0.1$, and various coding field sizes q. The upper bound on the probability of loss for $q \to \infty$ is also drawn.

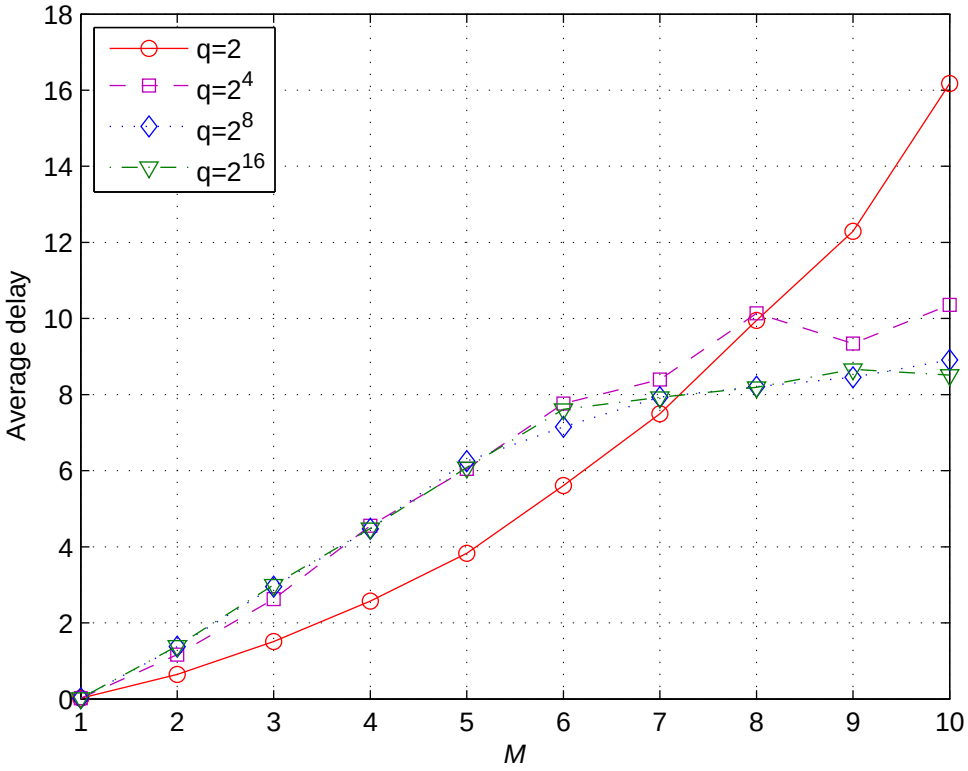

Figure 2.9: Average delay for 200,000 packets as a function of memory size M with $r = 0.8$, $\varepsilon = 0.1$, and various coding field sizes q.

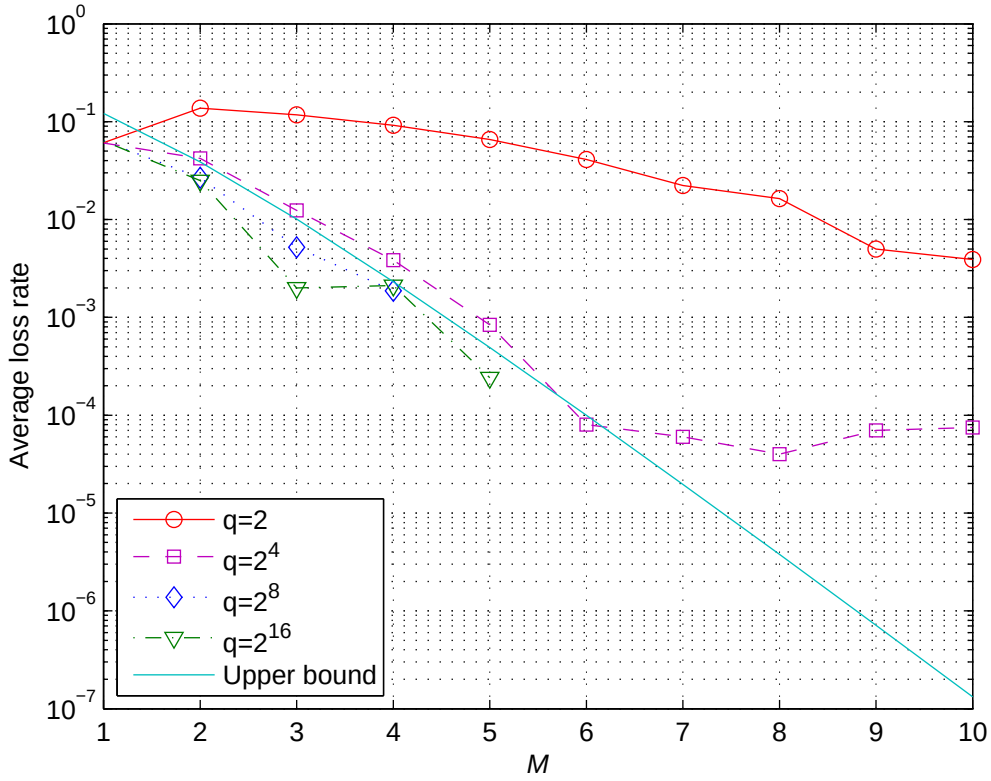

Figure 2.10: Average loss rate for 200,000 packets as a function of memory size M with $r = 0.6$, $\varepsilon = 0.1$, and various coding field sizes q. The upper bound on the probability of loss for $q \to \infty$ is also drawn.

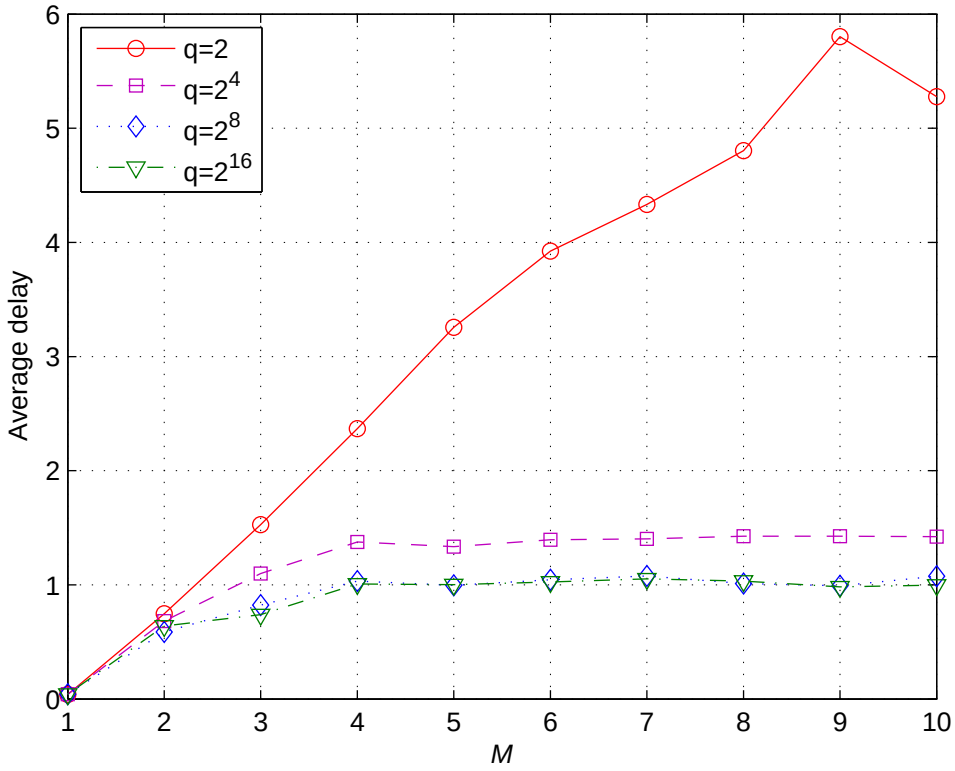

Figure 2.11: Average delay for 200,000 packets as a function of memory size M with $r = 0.6$, $\varepsilon = 0.1$, and various coding field sizes q.

2.4.2 Use in a two-link tandem network

When finite-memory random linear coding is used in isolation, packets are sometimes lost because the decoder receives linear combinations that, although innovative, are not decodable. For example, suppose the decoder receives $u_1 + u_2$, but is neither able to recover u_1 nor u_2 from other packets. This packet, $u_1 + u_2$, definitely gives the decoder some information, but, without either u_1 or u_2, the packet must be discarded. This would not be the case, however, if u_1 and u_2 were themselves coded packets—a trivial example, assuming that we are not coding over $\mathbb{F}_2$, is if $u_1 = u_2 = w_1$, where w_1 is a message packet for an outer code.

In this section, we consider finite-memory random linear coding in the context of a larger coded packet network. We consider the simplest set-up with an intermediate node: a two-link tandem network (see Figure 2.2) where we wish to establish a unicast connection from node 1 to node 3. Node 1 and node 3 use the coding scheme described in Section 2.1 without modification, while node 2 has only $M < K$ memory elements and uses the modified scheme. This simple two-link tandem network serves as a basis for longer tandem networks and more general network topologies.

We again discretize the time axis. We assume that, at each epoch, packets are injected by both nodes 1 and 2 and they are lost independently with probability δ and ε, respectively. Although situations of interest may not have transmissions that are synchronized in this way, the synchronicity assumption can be relaxed to an extent by accounting for differences in the packet injection rates using the loss rates.

We again conduct our analysis in the limit of infinite field size. The considerations for finite field size are the same as those mentioned in Section 2.4.1.

Let x_t denote the number of innovative packets (relative to $u_1, u_2, \ldots, u_N$) node 2 has for sending at time t, and let y_t denote the number of innovative packets received by node 3 at time t. By the arguments of Section 2.4.1, the following principles govern the evolution of x_t and y_t over time:

- As long as $x_t < M$, i.e., the memory does not already have M innovative

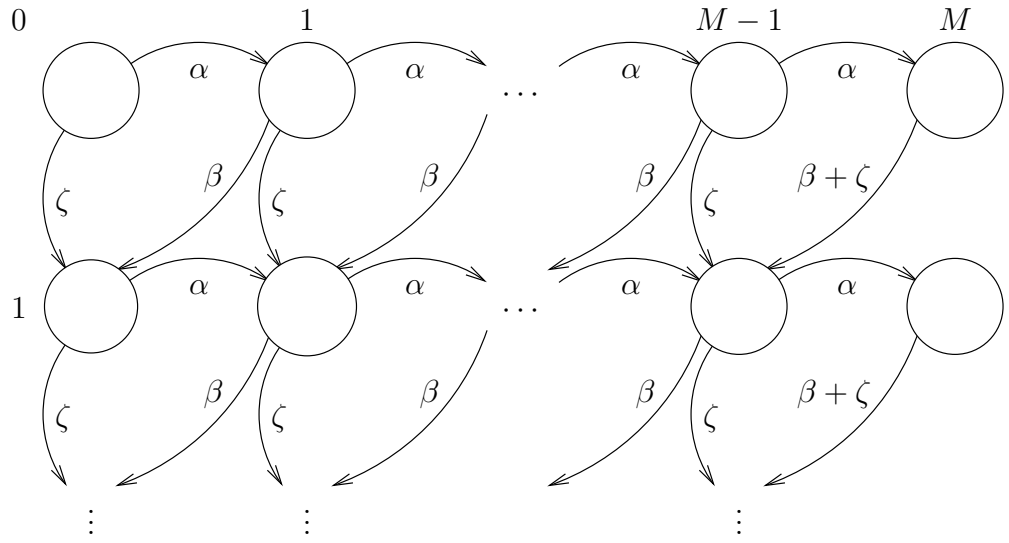

Figure 2.12: Markov chain modeling the evolution of x_t and y_t. To simplify the diagram, we do not show self-transitions.

packets, node 2 increases the innovation contents of its memory by 1 upon successful reception of a packet over arc $(1, 2)$.

- As long as $x_t > 0$, i.e., the memory is not completely redundant, the output of 2 is innovative, so y_t will increase by 1 provided that transmission over $(2, 3)$ is successful.

Let $\alpha := (1-\delta)\varepsilon$, $\beta := \delta(1-\varepsilon)$, and $\zeta := (1-\delta)(1-\varepsilon)$. Then the evolution of x_t and y_t is modeled by the Markov chain shown in Figure 2.12, where the horizontal coordinate of a state indicates x_t, and the vertical coordinate corresponds to the variable y_t.

We see that $\{x_t\}$ evolves as in Section 2.4.1, so its steady-state probabilities are given by (2.8) and (2.9) with $r = 1-\delta$. Hence, once the system is sufficiently mixed, the probability that y_t increases at time t is given by

$$\begin{aligned}\zeta\pi_0 + (1-\varepsilon)\pi_1 + \cdots + (1-\varepsilon)\pi_M &= (1-\varepsilon)(1-\delta\pi_0) \\ &= (1-\delta)(1-\pi_M).\end{aligned}$$

Therefore the system can operate at rate

$$R = (1 - \delta)(1 - \pi_M)$$

with high probability of success.

Suppose, without loss of generality, that $\delta > \varepsilon$, so $\varrho < 1$. Let R^* be the min-cut capacity, or maximum rate, of the system, which, in this case, is $1 - \delta$. Then the relative rate loss with respect to the min-cut rate is

$$1 - \frac{R}{R^*} = \pi_M. \tag{2.11}$$

As discussed before, our analysis assumes forming linear combinations over an infinitely large field, resulting in a Markov chain model with transition probabilities given in Figure 2.12. If on the other hand the field size is finite, we can still find new expressions for the transition probabilities, although the complete analysis becomes very complex. In particular, assume that the memory is used as an accumulator, so that the contents of the memory at each time are uniformly random linear combinations, over $\mathbb{F}_q$, of the received packets at node 2 by that time. Then, as we have mentioned, if the innovation content of the memory is x and a new packet arrives at node 2, the probability that node 2 can increase the innovation of its memory by 1 is $(1 - q^{x-M})$, independently from all other past events. Similarly, the probability that the output of node 2 is innovative is $(1 - q^{-x})$.

To quantify the effect of operations over a finite field, we simulated the evolution of this Markov chain for two combinations of δ and ϵ values that were also considered in Section 2.4.1 (see Figures 2.13 and 2.14). The effective rate is considered to be $R_e := y_N/N$, where N is the number of packet transmissions at A, and as before, y_N is the number of innovative packets received at node 3 by time N. We simulated this process for $N = 10^9$ packets. For different field sizes, we plot the relative rate loss with respect to the min-cut rate—i.e., $1 - R_e/R^*$—as a function of the memory size.

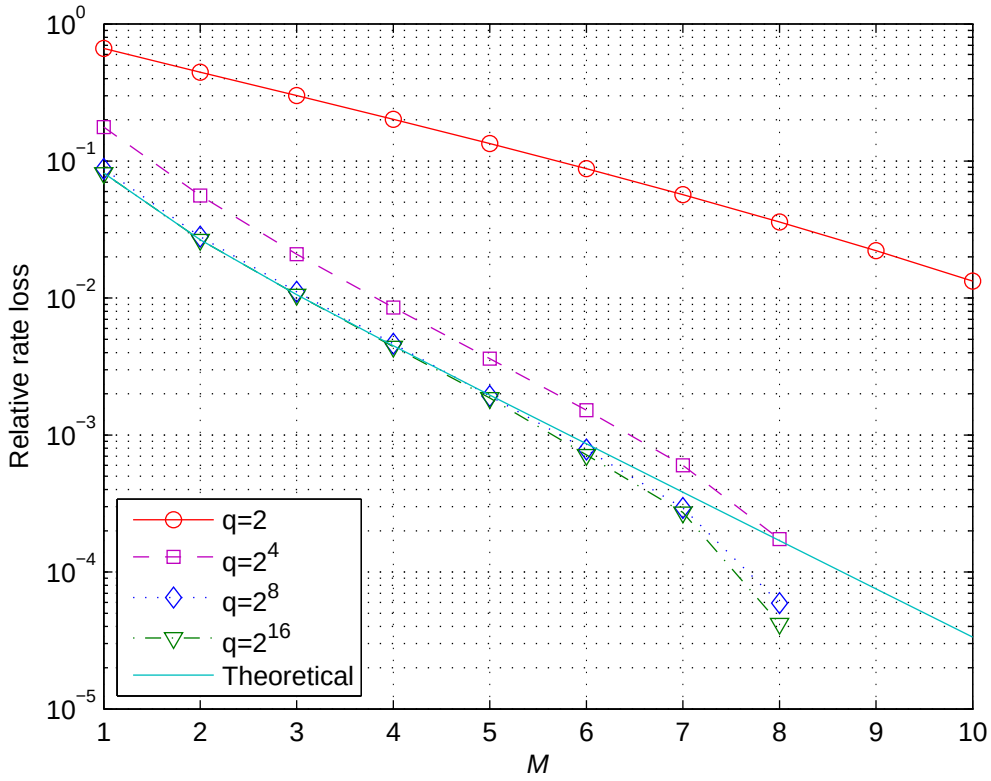

Figure 2.13: Relative rate loss with respect to min-cut rate as a function of memory size M for $\delta = 0.2$, $\varepsilon = 0.1$, and various coding field sizes q.

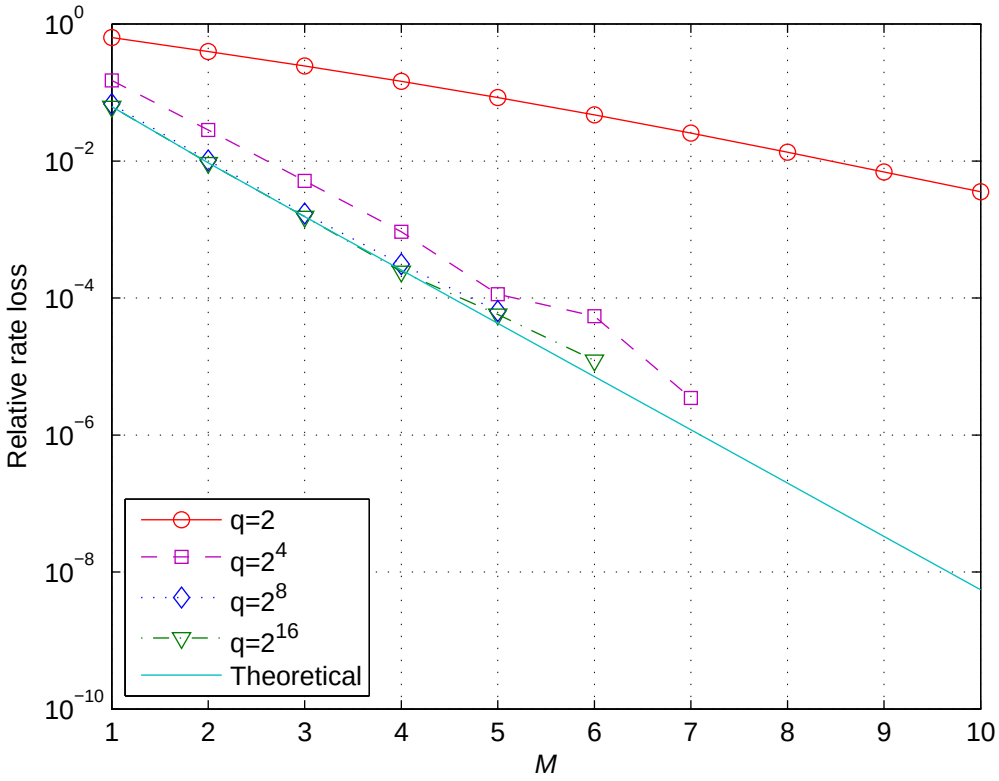

Figure 2.14: Relative rate loss with respect to min-cut rate as a function of memory size M for $\delta = 0.4$, $\varepsilon = 0.1$, and various coding field sizes q.

Also plotted is the theoretical result from (2.11).

By comparing Figures 2.13 and 2.14 with Figures 2.8 and 2.10, respectively, we see the advantage that comes from explicitly recognizing that the coding takes place in the context of a larger coded packet network. The loss rate in the latter plots essentially equates to the factor $1 - R_e/R^*$ in the former. Thus, in the limit of infinite q, we compare the probability of loss p_l upper bounded by equation (2.10) and the expression for $1 - R/R^*$ given by equation (2.11). We note that, in both cases, the decay as $M \to \infty$ is as ϱ^M. Moreover, it follows from our discussion that $1 - R/R^*$ must be a lower bound for p_l, hence p_l itself decays as ϱ^M as $M \to \infty$.

2.A Appendix: Formal arguments for main result

In this appendix, we given formal arguments for Theorem 2.1. Sections 2.A.1, 2.A.2, and 2.A.3 give formal arguments for three special cases of Theorem 2.1: the two-link tandem network, the L-link tandem network, and general unicast connections, respectively.

2.A.1 Two-link tandem network

All packets received by node 2, namely $v_1, v_2, \ldots, v_N$, are considered innovative. We associate with node 2 the set of vectors U, which varies with time and is initially empty, i.e., $U(0) := \emptyset$. If packet u is received by node 2 at time τ, then its auxiliary encoding vector β is added to U at time τ, i.e., $U(\tau^+) := \{\beta\} \cup U(\tau)$.

We associate with node 3 the set of vectors W, which again varies with time and is initially empty. Suppose packet u, with auxiliary encoding vector β, is received by node 3 at time τ. Let μ be a positive integer, which we call the *innovation order*. Then we say u is innovative if $\beta \notin \text{span}(W(\tau))$ and $|U(\tau)| > |W(\tau)| + \mu - 1$. If u is innovative, then β is added to W at time τ.[2]

[2]This definition of innovative differs from merely being informative, which is the sense in which innovative is used in Section 2.4 and in [25]. Indeed, a packet can be informative, in the sense that in

The definition of innovative is designed to satisfy two properties: First, we require that $W(\Delta)$, the set of vectors in W when the scheme terminates, is linearly independent. Second, we require that, when a packet is received by node 3 and $|U(\tau)| > |W(\tau)| + \mu - 1$, it is innovative with high probability. The innovation order μ is an arbitrary factor that ensures that the latter property is satisfied.

Suppose $|U(\tau)| > |W(\tau)| + \mu - 1$. Since u is a random linear combination of vectors in $U(\tau)$, it follows that u is innovative with some non-trivial probability. More precisely, because β is uniformly-distributed over $q^{|U(\tau)|}$ possibilities, of which at least $q^{|U(\tau)|} - q^{|W(\tau)|}$ are not in $\text{span}(W(\tau))$, it follows that

$$\Pr(\beta \notin \text{span}(W(\tau))) \geq \frac{q^{|U(\tau)|} - q^{|W(\tau)|}}{q^{|U(\tau)|}} = 1 - q^{|W(\tau)|-|U(\tau)|} \geq 1 - q^{-\mu}.$$

Hence u is innovative with probability at least $1 - q^{-\mu}$. Since we can always discard innovative packets, we assume that the event occurs with probability exactly $1 - q^{-\mu}$. If instead $|U(\tau)| \leq |W(\tau)| + \mu - 1$, then we see that u cannot be innovative, and this remains true at least until another arrival occurs at node 2. Therefore, for an innovation order of μ, the propagation of innovative packets through node 2 is described by the propagation of jobs through a single-server queueing station with queue size $(|U(\tau)| - |W(\tau)| - \mu + 1)^+$.

The queueing station is serviced with probability $1 - q^{-\mu}$ whenever the queue is non-empty and a received packet arrives on arc $(2, 3)$. We can equivalently consider "candidate" packets that arrive with probability $1 - q^{-\mu}$ whenever a received packet arrives on arc $(2, 3)$ and say that the queueing station is serviced whenever the queue is non-empty and a candidate packet arrives on arc $(2, 3)$. We consider all packets received on arc $(1, 2)$ to be candidate packets.

The system we wish to analyze, therefore, is the following simple queueing system:

gives a node some new, as yet unknown, information about $v_1, v_2, \ldots, v_N$ (or about $w_1, w_2, \ldots, w_K$), and not satisfy this definition of innovative. In this appendix, we have defined innovative so that innovative packets are informative (with respect to other innovative packets at the node), but not necessarily conversely. This allows us to bound, or dominate, the behavior of the coding scheme, though we cannot describe it exactly.

Jobs arrive at node 2 according to the arrival of received packets on arc $(1, 2)$ and, with the exception of the first $\mu - 1$ jobs, enter node 2's queue. The jobs in node 2's queue are serviced by the arrival of candidate packets on arc $(2, 3)$ and exit after being serviced. The number of jobs exiting is a lower bound on the number of packets with linearly-independent auxiliary encoding vectors received by node 3.

We analyze the queueing system of interest using the fluid approximation for discrete-flow networks (see, e.g., [23, 24]). We do not explicitly account for the fact that the first $\mu - 1$ jobs arriving at node 2 do not enter its queue because this fact has no effect on job throughput. Let B_1, B, and C be the counting processes for the arrival of received packets on arc $(1, 2)$, of innovative packets on arc $(2, 3)$, and of candidate packets on arc $(2, 3)$, respectively. Let $Q(\tau)$ be the number of jobs queued for service at node 2 at time τ. Hence $Q = B_1 - B$. Let $X := B_1 - C$ and $Y := C - B$. Then

$$Q = X + Y. \tag{2.12}$$

Moreover, we have

$$Q(\tau)dY(\tau) = 0, \tag{2.13}$$

$$dY(\tau) \geq 0, \tag{2.14}$$

and

$$Q(\tau) \geq 0 \tag{2.15}$$

for all $\tau \geq 0$, and

$$Y(0) = 0. \tag{2.16}$$

We observe now that equations (2.12)–(2.16) give us the conditions for a Skorohod problem (see, e.g., [24, Section 7.2]) and, by the oblique reflection mapping theorem, there is a well-defined, Lipschitz-continuous mapping Φ such that $Q = \Phi(X)$.

Let

$$\bar{C}^{(K)}(\tau) := \frac{C(K\tau)}{K},$$
$$\bar{X}^{(K)}(\tau) := \frac{X(K\tau)}{K},$$

and

$$\bar{Q}^{(K)}(\tau) := \frac{Q(K\tau)}{K}.$$

Recall that A_{233} is the counting process for the arrival of received packets on arc $(2,3)$. Therefore, $C(\tau)$ is the sum of $A_{233}(\tau)$ Bernoulli-distributed random variables with parameter $1 - q^{-\mu}$. Hence

$$\begin{aligned}\bar{C}(\tau) &:= \lim_{K\to\infty} \bar{C}^{(K)}(\tau) \\ &= \lim_{K\to\infty} (1 - q^{-\mu}) \frac{A_{233}(K\tau)}{K} \qquad \text{a.s.} \\ &= (1 - q^{-\mu}) z_{233}\tau \qquad \text{a.s.},\end{aligned}$$

where the last equality follows by the assumptions of the model. Therefore

$$\bar{X}(\tau) := \lim_{K\to\infty} \bar{X}^{(K)}(\tau) = (z_{122} - (1 - q^{-\mu}) z_{233})\tau \qquad \text{a.s.}$$

By the Lipschitz-continuity of Φ, then, it follows that $\bar{Q} := \lim_{K\to\infty} \bar{Q}^{(K)} = \Phi(\bar{X})$, i.e., $\bar{Q}$ is, almost surely, the unique $\bar{Q}$ that satisfies, for some $\bar{Y}$,

$$\bar{Q}(\tau) = (z_{122} - (1 - q^{-\mu}) z_{233})\tau + \bar{Y}, \tag{2.17}$$
$$\bar{Q}(\tau) d\bar{Y}(\tau) = 0, \tag{2.18}$$
$$d\bar{Y}(\tau) \geq 0, \tag{2.19}$$

and

$$\bar{Q}(\tau) \geq 0 \tag{2.20}$$

for all $\tau \geq 0$, and

$$\bar{Y}(0) = 0. \tag{2.21}$$

A pair $(\bar{Q}, \bar{Y})$ that satisfies (2.17)–(2.21) is

$$\bar{Q}(\tau) = (z_{122} - (1 - q^{-\mu})z_{233})^{+}\tau \tag{2.22}$$

and

$$\bar{Y}(\tau) = (z_{122} - (1 - q^{-\mu})z_{233})^{-}\tau,$$

where, for a real number x, $(x)^{+} := \max(x, 0)$ and $(x)^{-} := \max(-x, 0)$. Hence $\bar{Q}$ is given by equation (2.22).

Recall that node 3 can recover the message packets with high probability if it receives $\lfloor K(1+\varepsilon) \rfloor$ packets with linearly-independent auxiliary encoding vectors and that the number of jobs exiting the queueing system is a lower bound on the number of packets with linearly-independent auxiliary encoding vectors received by node 3. Therefore, node 3 can recover the message packets with high probability if $\lfloor K(1+\varepsilon) \rfloor$ or more jobs exit the queueing system. Let ν be the number of jobs that have exited the queueing system by time Δ. Then

$$\nu = B_1(\Delta) - Q(\Delta).$$

Take $K = \lceil (1 - q^{-\mu})\Delta R_c R/(1+\varepsilon) \rceil$, where $0 < R_c < 1$. Then

$$\begin{aligned}
\lim_{K\to\infty} \frac{\nu}{\lfloor K(1+\varepsilon) \rfloor} &= \lim_{K\to\infty} \frac{B_1(\Delta) - Q(\Delta)}{K(1+\varepsilon)} \\
&= \frac{z_{122} - (z_{122} - (1 - q^{-\mu})z_{233})^{+}}{(1 - q^{-\mu})R_c R} \\
&= \frac{\min(z_{122}, (1 - q^{-\mu})z_{233})}{(1 - q^{-\mu})R_c R} \\
&\geq \frac{1}{R_c} \frac{\min(z_{122}, z_{233})}{R} > 1
\end{aligned}$$

provided that

$$R \leq \min(z_{122}, z_{233}). \tag{2.23}$$

Hence, for all R satisfying (2.23), $\nu \geq \lfloor K(1+\varepsilon) \rfloor$ with probability arbitrarily close to 1 for K sufficiently large. The rate achieved is

$$\frac{K}{\Delta} \geq \frac{(1-q^{-\mu})R_c}{1+\varepsilon} R,$$

which can be made arbitrarily close to R by varying μ, R_c, and ε.

2.A.2 L-link tandem network

For $i = 2, 3, \ldots, L+1$, we associate with node i the set of vectors V_i, which varies with time and is initially empty. We define $U := V_2$ and $W := V_{L+1}$. As in the case of the two-link tandem, all packets received by node 2 are considered innovative and, if packet u is received by node 2 at time τ, then its auxiliary encoding vector β is added to U at time τ. For $i = 3, 4, \ldots, L+1$, if packet u, with auxiliary encoding vector β, is received by node i at time τ, then we say u is innovative if $\beta \notin \text{span}(V_i(\tau))$ and $|V_{i-1}(\tau)| > |V_i(\tau)| + \mu - 1$. If u is innovative, then β is added to V_i at time τ.

This definition of innovative is a straightforward extension of that in Appendix 2.A.1. The first property remains the same: we continue to require that $W(\Delta)$ is a set of linearly-independent vectors. We extend the second property so that, when a packet is received by node i for any $i = 3, 4, \ldots, L+1$ and $|V_{i-1}(\tau)| > |V_i(\tau)| + \mu - 1$, it is innovative with high probability.

Take some $i \in \{3, 4, \ldots, L+1\}$. Suppose that packet u, with auxiliary encoding vector β, is received by node i at time τ and that $|V_{i-1}(\tau)| > |V_i(\tau)| + \mu - 1$. Thus, the auxiliary encoding vector β is a random linear combination of vectors in some set V_0 that contains $V_{i-1}(\tau)$. Hence, because β is uniformly-distributed over $q^{|V_0|}$

possibilities, of which at least $q^{|V_0|} - q^{|V_i(\tau)|}$ are not in $\text{span}(V_i(\tau))$, it follows that

$$\Pr(\beta \notin \text{span}(V_i(\tau))) \geq \frac{q^{|V_0|} - q^{|V_i(\tau)|}}{q^{|V_0|}} = 1 - q^{|V_i(\tau)|-|V_0|} \geq 1 - q^{|V_i(\tau)|-|V_{i-1}(\tau)|} \geq 1 - q^{-\mu}.$$

Therefore u is innovative with probability at least $1 - q^{-\mu}$.

Following the argument in Appendix 2.A.1, we see, for all $i = 2, 3, \ldots, L$, that the propagation of innovative packets through node i is described by the propagation of jobs through a single-server queueing station with queue size $(|V_i(\tau)| - |V_{i+1}(\tau)| - \mu + 1)^+$ and that the queueing station is serviced with probability $1 - q^{-\mu}$ whenever the queue is non-empty and a received packet arrives on arc $(i, i+1)$. We again consider candidate packets that arrive with probability $1 - q^{-\mu}$ whenever a received packet arrives on arc $(i, i+1)$ and say that the queueing station is serviced whenever the queue is non-empty and a candidate packet arrives on arc $(i, i+1)$.

The system we wish to analyze in this case is therefore the following simple queueing network: Jobs arrive at node 2 according to the arrival of received packets on arc $(1, 2)$ and, with the exception of the first $\mu - 1$ jobs, enter node 2's queue. For $i = 2, 3, \ldots, L-1$, the jobs in node i's queue are serviced by the arrival of candidate packets on arc $(i, i+1)$ and, with the exception of the first $\mu - 1$ jobs, enter node $(i+1)$'s queue after being serviced. The jobs in node L's queue are serviced by the arrival of candidate packets on arc $(L, L+1)$ and exit after being serviced. The number of jobs exiting is a lower bound on the number of packets with linearly-independent auxiliary encoding vectors received by node $L + 1$.

We again analyze the queueing network of interest using the fluid approximation for discrete-flow networks, and we again do not explicitly account for the fact that the first $\mu - 1$ jobs arriving at a queueing node do not enter its queue. Let B_1 be the counting process for the arrival of received packets on arc $(1, 2)$. For $i = 2, 3, \ldots, L$, let B_i, and C_i be the counting processes for the arrival of innovative packets and candidate packets on arc $(i, i+1)$, respectively. Let $Q_i(\tau)$ be the number of jobs queued for service at node i at time τ. Hence, for $i = 2, 3, \ldots, L$, $Q_i = B_{i-1} - B_i$.

Let $X_i := C_{i-1} - C_i$ and $Y_i := C_i - B_i$, where $C_1 := B_1$. Then, we obtain a Skorohod problem with the following conditions: For all $i = 2, 3, \ldots, L$,

$$Q_i = X_i - Y_{i-1} + Y_i.$$

For all $\tau \geq 0$ and $i = 2, 3, \ldots, L$,

$$Q_i(\tau) dY_i(\tau) = 0,$$
$$dY_i(\tau) \geq 0,$$

and

$$Q_i(\tau) \geq 0.$$

For all $i = 2, 3, \ldots, L$,

$$Y_i(0) = 0.$$

Let

$$\bar{Q}_i^{(K)}(\tau) := \frac{Q_i(K\tau)}{K}$$

and $\bar{Q}_i := \lim_{K\to\infty} \bar{Q}_i^{(K)}$ for $i = 2, 3, \ldots, L$. Then the vector $\bar{Q}$ is, almost surely, the unique $\bar{Q}$ that satisfies, for some $\bar{Y}$,

$$\bar{Q}_i(\tau) = \begin{cases} (z_{122} - (1 - q^{-\mu}) z_{233})\tau + \bar{Y}_2(\tau) & \text{if } i = 2, \\ (1 - q^{-\mu})(z_{(i-1)ii} - z_{i(i+1)(i+1)})\tau + \bar{Y}_i(\tau) - \bar{Y}_{i-1}(\tau) & \text{otherwise,} \end{cases} \tag{2.24}$$

$$\bar{Q}_i(\tau) d\bar{Y}_i(\tau) = 0, \tag{2.25}$$

$$d\bar{Y}_i(\tau) \geq 0, \tag{2.26}$$

and

$$\bar{Q}_i(\tau) \geq 0 \tag{2.27}$$

for all $\tau \geq 0$ and $i = 2, 3, \ldots, L$, and

$$\bar{Y}_i(0) = 0 \tag{2.28}$$

for all $i = 2, 3, \ldots, L$.

A pair $(\bar{Q}, \bar{Y})$ that satisfies (2.24)–(2.28) is

$$\bar{Q}_i(\tau) = (\min(z_{122}, \min_{2 \leq j < i}\{(1 - q^{-\mu})z_{j(j+1)(j+1)}\}) - (1 - q^{-\mu})z_{i(i+1)(i+1)})^+ \tau \tag{2.29}$$

and

$$\bar{Y}_i(\tau) = (\min(z_{122}, \min_{2 \leq j < i}\{(1 - q^{-\mu})z_{j(j+1)(j+1)}\}) - (1 - q^{-\mu})z_{i(i+1)(i+1)})^- \tau.$$

Hence $\bar{Q}$ is given by equation (2.29).

The number of jobs that have exited the queueing network by time Δ is given by

$$\nu = B_1(\Delta) - \sum_{i=2}^{L} Q_i(\Delta).$$

Take $K = \lceil (1 - q^{-\mu})\Delta R_c R/(1 + \varepsilon) \rceil$, where $0 < R_c < 1$. Then

$$\begin{aligned} \lim_{K \to \infty} \frac{\nu}{\lfloor K(1+\varepsilon) \rfloor} &= \lim_{K \to \infty} \frac{B_1(\Delta) - \sum_{i=2}^{L} Q(\Delta)}{K(1+\varepsilon)} \\ &= \frac{\min(z_{122}, \min_{2 \leq i \leq L}\{(1 - q^{-\mu})z_{i(i+1)(i+1)}\})}{(1 - q^{-\mu})R_c R} \\ &\geq \frac{1}{R_c} \frac{\min_{1 \leq i \leq L}\{z_{i(i+1)(i+1)}\}}{R} > 1 \end{aligned} \tag{2.30}$$

provided that

$$R \leq \min_{1 \leq i \leq L}\{z_{i(i+1)(i+1)}\}. \tag{2.31}$$

Hence, for all R satisfying (2.31), $\nu \geq \lfloor K(1 + \varepsilon) \rfloor$ with probability arbitrarily close to 1 for K sufficiently large. The rate can again be made arbitrarily close to R by

varying μ, R_c, and ε.

2.A.3 General unicast connection

Consider a single path p_m. We write $p_m = \{i_1, i_2, \ldots, i_{L_m}, i_{L_m+1}\}$, where $i_1 = s$ and $i_{L_m+1} = t$. For $l = 2, 3, \ldots, L_m + 1$, we associate with node i_l the set of vectors $V_l^{(p_m)}$, which varies with time and is initially empty. We define $U^{(p_m)} := V_2^{(p_m)}$ and $W^{(p_m)} := V_{L_m+1}^{(p_m)}$.

We note that the constraint (2.3) can also be written as

$$x_{iJj} \leq \sum_{\{L \subset J | j \in L\}} \alpha_{iJL}^{(j)} z_{iJL}$$

for all $(i, J) \in \mathcal{A}$ and $j \in J$, where $\sum_{j \in L} \alpha_{iJL}^{(j)} = 1$ for all $(i, J) \in \mathcal{A}$ and $L \subset J$, and $\alpha_{iJL}^{(j)} \geq 0$ for all $(i, J) \in \mathcal{A}$, $L \subset J$, and $j \in L$. Suppose packet u, with auxiliary encoding vector β, is placed on hyperarc (i_1, J) and received by $K \subset J$, where $K \ni i_2$, at time τ. We associate with u the independent random variable P_u, which takes the value m with probability $R_m \alpha_{i_1 JK}^{(i_2)} / \sum_{\{L \subset J | i_2 \in L\}} \alpha_{i_1 JL}^{(i_2)} z_{iJL}$. If $P_u = m$, then we say u is innovative on path p_m, and β is added to $U^{(p_m)}$ at time τ.

Take $l = 2, 3, \ldots, L_m$. Now suppose packet u, with auxiliary encoding vector β, is placed on hyperarc (i_l, J) and received by $K \subset J$, where $K \ni i_{l+1}$, at time τ. We associate with u the independent random variable P_u, which takes the value m with probability $R_m \alpha_{i_l JK}^{(i_{l+1})} / \sum_{\{L \subset J | i_{l+1} \in L\}} \alpha_{i_l JL}^{(i_{l+1})} z_{iJL}$. We say u is innovative on path p_m if $P_u = m$, $\beta \notin \text{span}(\cup_{n=1}^{m-1} W^{(p_n)}(\Delta) \cup V_{l+1}^{(p_m)}(\tau) \cup \cup_{n=m+1}^{M} U^{(p_n)}(\Delta))$, and $|V_l^{(p_m)}(\tau)| > |V_{l+1}^{(p_m)}(\tau)| + \mu - 1$.

This definition of innovative is somewhat more complicated than that in Appendices 2.A.1 and 2.A.2 because we now have M paths that we wish to analyze separately. We have again designed the definition to satisfy two properties: First, we require that $\cup_{m=1}^{M} W^{(p_m)}(\Delta)$ is linearly-independent. This is easily verified: Vectors are added to $W^{(p_1)}(\tau)$ only if they are linearly independent of existing ones; vectors

are added to $W^{(p_2)}(\tau)$ only if they are linearly independent of existing ones and ones in $W^{(p_1)}(\Delta)$; and so on. Second, we require that, when a packet is received by node i_l, $P_u = m$, and $|V_{l-1}^{(p_m)}(\tau)| > |V_l^{(p_m)}(\tau)| + \mu - 1$, it is innovative on path p_m with high probability.

Take $l \in \{3, 4, \ldots, L_m+1\}$. Suppose that packet u, with auxiliary encoding vector β, is received by node i_l at time τ, that $P_u = m$, and that $|V_{l-1}^{(p_m)}(\tau)| > |V_l^{(p_m)}(\tau)| + \mu - 1$. Thus, the auxiliary encoding vector β is a random linear combination of vectors in some set V_0 that contains $V_{l-1}^{(p_m)}(\tau)$. Hence β is uniformly-distributed over $q^{|V_0|}$ possibilities, of which at least $q^{|V_0|} - q^d$ are not in $\text{span}(V_l^{(p_m)}(\tau) \cup \tilde{V}_{\setminus m})$, where $d := \dim(\text{span}(V_0) \cap \text{span}(V_l^{(p_m)}(\tau) \cup \tilde{V}_{\setminus m}))$. Note that $V_{l-1}^{(p_m)}(\tau) \cup \tilde{V}_{\setminus m}$ forms a linearly-independent set, so

$$\begin{aligned} d - |V_0| &\leq \dim(\text{span}(V_{l-1}^{(p_m)}(\tau)) \cap \text{span}(V_l^{(p_m)}(\tau) \cup \tilde{V}_{\setminus m})) - |V_{l-1}^{(p_m)}(\tau)| \\ &= \dim(\text{span}(V_{l-1}^{(p_m)}(\tau)) \cap \text{span}(V_l^{(p_m)}(\tau))) - |V_{l-1}^{(p_m)}(\tau)| \\ &\leq |V_l^{(p_m)}(\tau)| - |V_{l-1}^{(p_m)}(\tau)| \leq -\mu. \end{aligned}$$

Therefore, it follows that

$$\Pr(\beta \notin \text{span}(V_l^{(p_m)}(\tau) \cup \tilde{V}_{\setminus m})) \geq \frac{q^{|V_0|} - q^d}{q^{|V_0|}} = 1 - q^{d-|V_0|} \geq 1 - q^{-\mu}.$$

We see then that, if we consider only those packets such that $P_u = m$, the conditions that govern the propagation of innovative packets are exactly those of an L_m-link tandem network, which we dealt with in Appendix 2.A.2. By recalling the distribution of P_u, it follows that the propagation of innovative packets along path p_m behaves like an L_m-link tandem network with average arrival rate R_m on every link. Since we have assumed nothing special about m, this statement applies for all $m = 1, 2, \ldots, M$.

Take $K = \lceil (1-q^{-\mu})\Delta R_c R/(1+\varepsilon) \rceil$, where $0 < R_c < 1$. Then, by equation (2.30),

$$\lim_{K\to\infty} \frac{|W^{(p_m)}(\Delta)|}{\lfloor K(1+\varepsilon) \rfloor} > \frac{R_m}{R}.$$

Hence

$$\lim_{K\to\infty} \frac{|\cup_{m=1}^{M} W^{(p_m)}(\Delta)|}{\lfloor K(1+\varepsilon) \rfloor} = \sum_{m=1}^{M} \frac{|W^{(p_m)}(\Delta)|}{\lfloor K(1+\varepsilon) \rfloor} > \sum_{m=1}^{M} \frac{R_m}{R} = 1.$$

As before, the rate can be made arbitrarily close to R by varying μ, R_c, and ε.

Chapter 3

Subgraph Selection

We now turn to the subgraph selection part of the efficient operation problem. This is the problem of determining the coding subgraph to use given that the network code is decided. In our case, we assume that the network code is given by the scheme examined in the previous chapter. Since this scheme achieves the capacity of a single multicast connection in a given subgraph, in using it and determining the coding subgraph independently, there is no loss of optimality in the efficient operation problem provided that we are constrained to only coding packets within a single connection.[1] Relaxing this constraint, and allowing coded packets to be formed using packets from two or more connections, is known to afford an improvement, but finding capacity-achieving codes is a very difficult problem—one that, in fact, currently remains open with only cumbersome bounds on the capability of coding [99] and examples that demonstrate the insufficiency of various classes of linear codes [32, 82, 90, 93]. Constraining coding to packets within a single connection is called superposition coding [115], and there is evidence to suggest that it may be near-optimal [65]. We therefore content ourselves with coding only within a single connection, allowing us to separate network coding from subgraph selection without

[1] This statement assumes that no information is conveyed by the timing of packets. In general, the timing of packets can be used to convey information, but the amount of information communicated by timing does not grow in the size of packets, so the effect of such "timing channels" is negligible for large packet sizes.

loss of optimality.

We formulate the subgraph selection problem in Section 3.1. The problem we describe is rich one and the direction we take is simply the one that we believe is most appropriate. Certainly, there are many more directions to take, and our work has lead to follow-on work that extend the problem and explore other facets of it (see, e.g., [15, 18, 66, 101, 108, 112, 113]). In Section 3.2, we discuss distributed algorithms for solving the problem. Such algorithms allow subgraphs to be computed in a distributed manner, with each node making computations based only on local knowledge and knowledge acquired from information exchanges. Perhaps the most well-known distributed algorithm in networking is the distributed Bellman-Ford algorithm (see, e.g., [13, Section 5.2]), which is used to find routes in routed packet networks. Designing algorithms that can be run in a distributed manner is not an easy task and, though we do manage to do so, they apply only in cases where links essentially behave independently and medium access issues do not pose significant constraints, either because they are non-existent or because they are dealt with separately (in contrast to the slotted Aloha relay channel of Section 1.2.1, where medium access issues form a large part of the problem and must be dealt with directly). In Section 3.3, we introduce a dynamic component into the problem. Dynamics, such as changes in the membership of the multicast group or changes in the positions of the nodes, are often present in problems of interest. We consider the scenario where membership of the multicast group changes in time, with nodes joining and leaving the group, and continuous service to all members of the group must be maintained—a problem we call dynamic multicast.

3.1 Problem formulation

We specify a multicast connection with a triplet $(s, T, \{R_t\}_{t\in T})$, where s is the source of the connection, T is the set of sinks, and $\{R_t\}_{t\in T}$ is the set of rates to the sinks (see Section 2.2.2). Suppose we wish to establish C multicast connections,

$(s_1, T_1, \{R_{t,1}\}), \ldots, (s_C, T_C, \{R_{t,C}\})$. Using Theorem 2.1 and the max-flow/min-cut theorem, we see that the efficient operation problem can now be phrased as the following mathematical programming problem:

$$\begin{aligned}
&\text{minimize } f(z) \\
&\text{subject to } z \in Z, \\
&\sum_{c=1}^{C} y_{iJK}^{(c)} \le z_{iJK}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J, \\
&\sum_{j \in K} x_{iJj}^{(t,c)} \le \sum_{\{L \subset J | L \cap K \neq \emptyset\}} y_{iJL}^{(c)}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J,\ t \in T_c,\ c = 1, \ldots, C, \\
&x^{(t,c)} \in F^{(t,c)}, \qquad \forall\ t \in T_c,\ c = 1, \ldots, C,
\end{aligned} \tag{3.1}$$

where $x^{(t,c)}$ is the vector consisting of $x_{iJj}^{(t,c)}$, $(i,J) \in \mathcal{A}$, $j \in J$, and $F^{(t,c)}$ is the bounded polyhedron of points $x^{(t,c)}$ satisfying the conservation of flow constraints

$$\sum_{\{J | (i,J) \in \mathcal{A}\}} \sum_{j \in J} x_{iJj}^{(t,c)} - \sum_{\{j | (j,I) \in \mathcal{A}, i \in I\}} x_{jIi}^{(t,c)} = \begin{cases} R_{t,c} & \text{if } i = s_c, \\ -R_{t,c} & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall\ i \in \mathcal{N},$$

and non-negativity constraints

$$x_{iJj}^{(t,c)} \ge 0, \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J.$$

In this formulation, $y_{iJK}^{(c)}$ represents the average rate of packets that are injected on hyperarc (i,J) and received by exactly the set of nodes K (which occurs with average rate z_{iJK}) and that are allocated to connection c.

For simplicity, let us consider the case where $C = 1$. The extension to $C > 1$ is conceptually straightforward and, moreover, the case where $C = 1$ is interesting in its own right: whenever each multicast group has a selfish cost objective, or when the network sets link weights to meet its objective or enforce certain policies and each

multicast group is subject to a minimum-weight objective, we wish to establish single efficient multicast connections.

Let

$$b_{iJK} := \frac{\sum_{\{L \subset J | L \cap K \neq \emptyset\}} z_{iJL}}{z_{iJ}},$$

which is the fraction of packets injected on hyperarc (i, J) that are received by a set of nodes that intersects K. Problem (3.1) is now

$$\begin{aligned} &\text{minimize } f(z) \\ &\text{subject to } z \in Z, \\ &\qquad \sum_{j \in K} x^{(t)}_{iJj} \leq z_{iJ} b_{iJK}, \qquad \forall\ (i, J) \in \mathcal{A},\ K \subset J,\ t \in T, \\ &\qquad x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T. \end{aligned} \tag{3.2}$$

In the lossless case, problem (3.2) simplifies to the following problem:

$$\begin{aligned} &\text{minimize } f(z) \\ &\text{subject to } z \in Z, \\ &\qquad \sum_{j \in J} x^{(t)}_{iJj} \leq z_{iJ}, \qquad \forall\ (i, J) \in \mathcal{A},\ t \in T, \\ &\qquad x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T. \end{aligned} \tag{3.3}$$

As an example, consider the network depicted in Figure 3.1, which consists only of point-to-point links. Suppose that the network is lossless, that we wish to achieve multicast of unit rate from s to two sinks, t_1 and t_2, and that we have $Z = [0, 1]^{|\mathcal{A}|}$ and $f(z) = \sum_{(i,j) \in \mathcal{A}} z_{ij}$. An optimal solution to problem (3.3) is shown in the figure. We have flows $x^{(1)}$ and $x^{(2)}$ of unit size from s to t_1 and t_2, respectively, and, for each arc (i, j), $z_{ij} = \max(x^{(1)}_{ijj}, x^{(2)}_{ijj})$, as we expect from the optimization.

The same multicast problem in a routed packet network would entail minimizing the number of arcs used to form a tree that is rooted at s and that reaches t_1 and t_2—in other words, solving the Steiner tree problem on directed graphs [89]. The Steiner

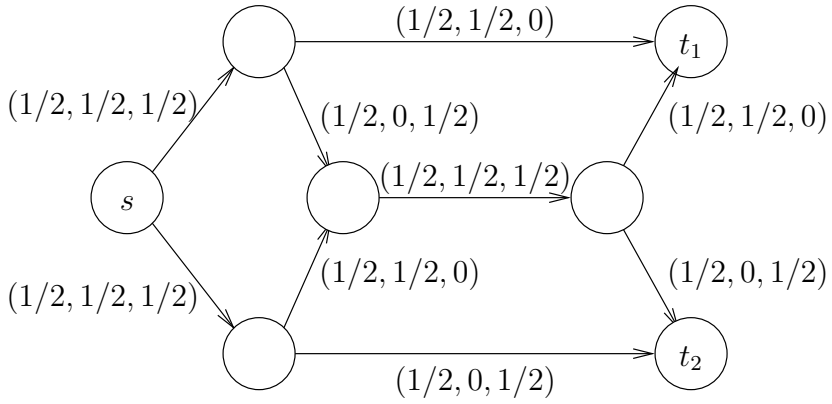

Figure 3.1: A network of lossless point-to-point links with multicast from s to $T = \{t_1, t_2\}$. Each arc is marked with the triple $(z_{ij}, x^{(1)}_{ijj}, x^{(2)}_{ijj})$.

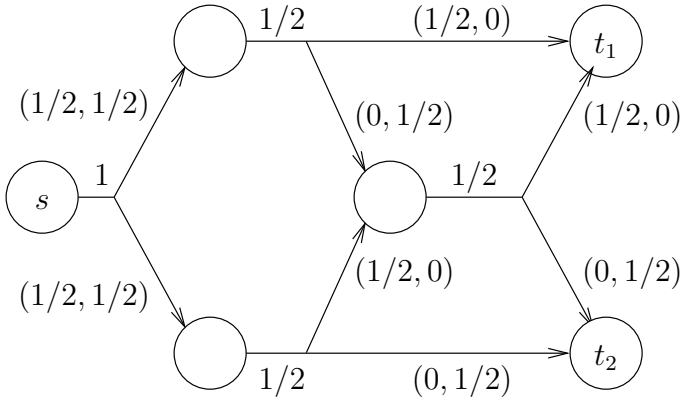

Figure 3.2: A network of lossless broadcast links with multicast from s to $T = \{t_1, t_2\}$. Each hyperarc is marked with z_{iJ} at its start and the pair $(x^{(1)}_{iJj}, x^{(2)}_{iJj})$ at its ends.

tree problem on directed graphs is well-known to be NP-complete, but solving problem (3.3) is not. In this case, problem (3.3) is in fact a linear optimization problem. It is a linear optimization problem that can be thought of as a fractional relaxation of the Steiner tree problem [117]. This example illustrates one of the attractive features of the coded approach: it allows us avoid an NP-complete problem and instead solve its fractional relaxation. In Section 4.2, we examine the efficiency improvements that we can achieve from this feature.

For an example with broadcast links, consider the network depicted in Figure 3.2. Suppose again that the network is lossless, that we wish to achieve multicast of unit rate from s to two sinks, t_1 and t_2, and that we have $Z = [0,1]^{|\mathcal{A}|}$ and $f(z) = \sum_{(i,J)\in\mathcal{A}} z_{iJ}$. An optimal solution to problem (3.3) is shown in the figure. We still have flows $x^{(1)}$ and $x^{(2)}$ of unit size from s to t_1 and t_2, respectively, but now, for each hyperarc (i,J), we determine z_{iJ} from the various flows passing through hyperarc (i,J), each destined toward a single node j in J, and the optimization gives $z_{iJ} = \max(\sum_{j\in J} x^{(1)}_{iJj}, \sum_{j\in J} x^{(2)}_{iJj})$.

Neither problem (3.2) nor (3.3) as it stands is easy to solve. But the problems are very general. Their complexities improve if we assume that the cost function is separable and possibly even linear, i.e., if we suppose $f(z) = \sum_{(i,J)\in A} f_{iJ}(z_{iJ})$, where f_{iJ} is a convex or linear function, which is a very reasonable assumption in many practical situations. For example, packet latency is usually assessed with a separable, convex cost function and energy, monetary cost, and total weight are usually assessed with separable, linear cost functions. The problems examined in our performance evaluation in Chapter 4, which we believe reflect problems of practical interest, all involve separable, linear cost functions.

The complexities of problems (3.2) and (3.3) also improve if we make some assumptions on the form of the constraint set Z, which is the case in most practical situations.

A particular simplification applies if we assume that, when nodes transmit in a

lossless network, they reach all nodes in a certain region, with cost increasing as this region is expanded. This applies, for example, if we are interested in minimizing energy consumption, and the region in which a packet is reliably received expands as we expend more energy in its transmission. More precisely, suppose that we have separable cost, so $f(z) = \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ})$. Suppose further that each node i has M_i outgoing hyperarcs $(i, J_1^{(i)}), (i, J_2^{(i)}), \ldots, (i, J_{M_i}^{(i)})$ with $J_1^{(i)} \subsetneq J_2^{(i)} \subsetneq \cdots \subsetneq J_{M_i}^{(i)}$. (We assume that there are no identical links, as duplicate links can effectively be treated as a single link.) Then, we assume that $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$ for all $\zeta \geq 0$ and nodes i.

Let us introduce, for $(i,j) \in \mathcal{A}' := \{(i,j)|(i,J) \in A, J \ni j\}$, the variables

$$\hat{x}_{ij}^{(t)} := \sum_{m=m(i,j)}^{M_i} x_{iJ_m^{(i)}j}^{(t)},$$

where $m(i,j)$ is the unique m such that $j \in J_m^{(i)} \setminus J_{m-1}^{(i)}$ (we define $J_0^{(i)} := \emptyset$ for all $i \in \mathcal{N}$ for convenience). Now, problem (3.3) can be reformulated as the following problem, which has substantially fewer variables:

$$\begin{aligned}
&\text{minimize} \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ}) \\
&\text{subject to } z \in Z \\
&\qquad \sum_{k\in J_{M_i}^{(i)}\setminus J_{m-1}^{(i)}} \hat{x}_{ik}^{(t)} \leq \sum_{n=m}^{M_i} z_{iJ_n^{(i)}}, \qquad \forall\, i \in \mathcal{N},\, m = 1,\ldots,M_i,\, t \in T, \\
&\qquad \hat{x}^{(t)} \in \hat{F}^{(t)}, \qquad \forall\, t \in T,
\end{aligned} \tag{3.4}$$

where $\hat{F}^{(t)}$ is the bounded polyhedron of points $\hat{x}^{(t)}$ satisfying the conservation of flow

constraints

$$\sum_{\{j|(i,j)\in\mathcal{A}'\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}'\}} \hat{x}_{ji}^{(t)} = \begin{cases} R_t & \text{if } i = s, \\ -R_t & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall\ i \in N,$$

and non-negativity constraints

$$0 \le \hat{x}_{ij}^{(t)}, \qquad \forall\ (i,j) \in \mathcal{A}'.$$

Proposition 3.1. *Suppose that* $f(z) = \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ})$ *and that* $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$ *for all* $\zeta \ge 0$ *and* $i \in \mathcal{N}$. *Then problem (3.3) and problem (3.4) are equivalent in the sense that they have the same optimal cost and* z *is part of an optimal solution for (3.3) if and only if it is part of an optimal solution for (3.4).*

Proof. Suppose (x, z) is a feasible solution to problem (3.3). Then, for all $(i,j) \in \mathcal{A}'$ and $t \in T$,

$$\begin{aligned}
\sum_{m=m(i,j)}^{M_i} z_{iJ_m^{(i)}} &\ge \sum_{m=m(i,j)}^{M_i} \sum_{k\in J_m^{(i)}} x_{iJ_m^{(i)}k}^{(t)} \\
&= \sum_{k\in J_{M_i}^{(i)}} \sum_{m=\max(m(i,j),m(i,k))}^{M_i} x_{iJ_m^{(i)}k}^{(t)} \\
&\ge \sum_{k\in J_{M_i}^{(i)}\setminus J_{m(i,j)-1}^{(i)}} \sum_{m=\max(m(i,j),m(i,k))}^{M_i} x_{iJ_m^{(i)}k}^{(t)} \\
&= \sum_{k\in J_{M_i}^{(i)}\setminus J_{m(i,j)-1}^{(i)}} \sum_{m=m(i,k)}^{M_i} x_{iJ_m^{(i)}k}^{(t)} \\
&= \sum_{k\in J_{M_i}^{(i)}\setminus J_{m(i,j)-1}^{(i)}} \hat{x}_{ik}^{(t)}.
\end{aligned}$$

Hence $(\hat{x}, z)$ is a feasible solution of problem (3.4) with the same cost.

Now suppose $(\hat{x}, z)$ is an optimal solution of problem (3.4). Since $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$ for all $\zeta \geq 0$ and $i \in \mathcal{N}$ by assumption, it follows that, for all $i \in \mathcal{N}$, the sequence $z_{iJ_1^{(i)}}, z_{iJ_2^{(i)}}, \ldots, z_{iJ_{M_i}^{(i)}}$ is given recursively, starting from $m = M_i$, by

$$z_{iJ_m^{(i)}} = \max_{t \in T} \left\{ \sum_{k \in J_{M_i}^{(i)} \setminus J_{m-1}^{(i)}} \hat{x}_{ik}^{(t)} \right\} - \sum_{m'=m+1}^{M_i} z_{iJ_{m'}^{(i)}}.$$

Hence $z_{iJ_m^{(i)}} \geq 0$ for all $i \in \mathcal{N}$ and $m = 1, 2, \ldots, M_i$. We then set, starting from $m = M_i$ and $j \in J_{M_i}^{(i)}$,

$$x_{iJ_m^{(i)}j}^{(t)} := \min \left(\hat{x}_{ij}^{(t)} - \sum_{l=m+1}^{M_i} x_{iJ_l^{(i)}j}, z_{iJ_m^{(i)}} - \sum_{k \in J_{M_i}^{(i)} \setminus J_{m(i,j)}^{(i)}} x_{iJ_m^{(i)}k}^{(t)} \right).$$

It is now not difficult to see that (x, z) is a feasible solution of problem (3.3) with the same cost.

Therefore, the optimal costs of problems (3.3) and (3.4) are the same and, since the objective functions for the two problems are the same, z is part of an optimal solution for problem (3.3) if and only if it is part of an optimal solution for problem (3.4). □

3.1.1 An example

Let us return again to the slotted Aloha relay channel described in Section 1.2.1. The relevant optimization problem to solve in this case is (3.2), and it reduces to (cf. Section 2.2.3)

$$\begin{aligned} &\text{minimize } z_{1(23)} + z_{23} \\ &\text{subject to } 0 \leq z_{1(23)}, z_{23} \leq 1, \\ &\qquad R \leq z_{1(23)}(1 - z_{23})(p_{1(23)2} + p_{1(23)3} + p_{1(23)(23)}), \\ &\qquad R \leq z_{1(23)}(1 - z_{23})(p_{1(23)3} + p_{1(23)(23)}) + (1 - z_{1(23)}) z_{23} p_{233}. \end{aligned}$$

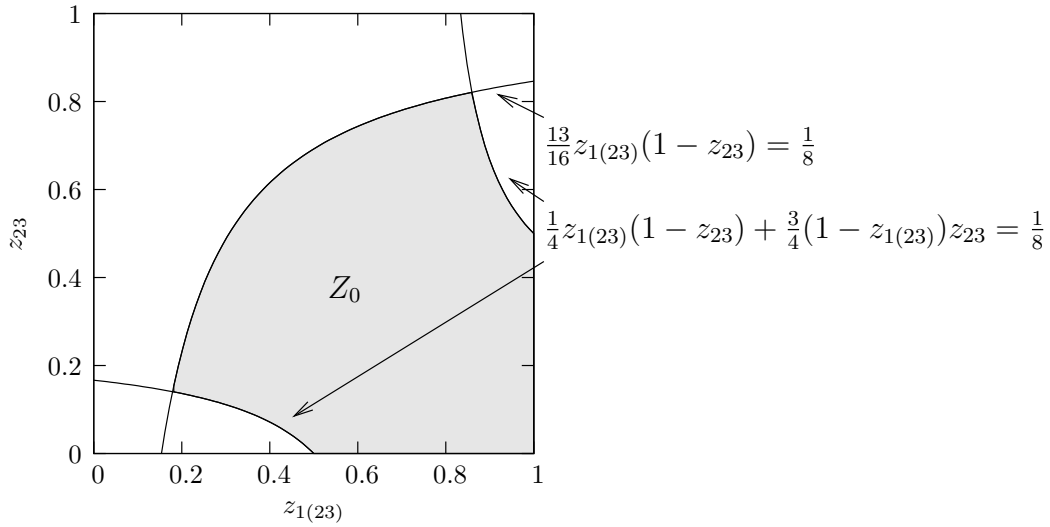

Figure 3.3: Feasible set of problem (3.5).

Let us assume some values for the parameters of the problem and work through it. Let $R := 1/8$, $p_{1(23)2} := 9/16$, $p_{1(23)3} := 1/16$, $p_{1(23)(23)} := 3/16$, and $p_{233} := 3/4$. Then the optimization problem we have is

$$\begin{aligned} &\text{minimize } z_{1(23)} + z_{23} \\ &\text{subject to } 0 \le z_{1(23)}, z_{23} \le 1, \\ &\qquad \frac{1}{8} \le \frac{13}{16} z_{1(23)}(1 - z_{23}), \\ &\qquad \frac{1}{8} \le \frac{1}{4} z_{1(23)}(1 - z_{23}) + \frac{3}{4}(1 - z_{1(23)}) z_{23}. \end{aligned} \tag{3.5}$$

The feasible set of this problem is shown in Figure 3.3. It is the shaded region labeled Z_0. By inspection, the optimal solution of (3.5) is the lesser of the two intersections between the curves defined by

$$\frac{13}{16} z_{1(23)}(1 - z_{23}) = \frac{1}{8}$$

and

$$\frac{1}{4} z_{1(23)}(1 - z_{23}) + \frac{3}{4}(1 - z_{1(23)}) z_{23} = \frac{1}{8}.$$

We obtain $z^*_{1(23)} \simeq 0.179$ and $z^*_{23} \simeq 0.141$.

The problem we have just solved is by no means trivial. We have taken a wireless packet network subject to losses that are determined by a complicated set of conditions—including medium contention—and found a way of establishing a given unicast connection of fixed throughput using the minimum number of transmissions per message packet. The solution is that node 1 transmits a packet every time slot with probability 0.179, and node 2 transmits a packet every time slot independently with probability 0.141. Whenever either node transmits a packet, they follow the coding scheme of Section 2.1.

The network we dealt with was, unfortunately, only a small one, and the solution method we used will not straightforwardly scale to larger problems. But the solution method is conceptually simple, and there are cases where the solution to large problems is computable—and computable in a distributed manner. This is the topic of the next section.

3.2 Distributed algorithms

In many cases, the optimization problems (3.2), (3.3), and (3.4) are convex or linear problems and their solutions can, in theory, be computed. For practical network applications, however, it is often important that solutions can be computed in a distributed manner, with each node making computations based only on local knowledge and knowledge acquired from information exchanges. Thus, we seek distributed algorithms to solve optimization problems (3.2), (3.3), and (3.4), which, when paired with the random linear coding scheme of the previous chapter, yields a distributed approach to efficient operation. The algorithms we propose will generally take some time to converge to an optimal solution, but it is not necessary to wait until the algorithms have converged before transmission—we can apply the coding scheme to the coding subgraph we have at any time, optimal or otherwise, and continue doing so while it converges. Such an approach is robust to dynamics such as changes in

network topology that cause the optimal solution to change, because the algorithms will simply converge toward the changing optimum.

To this end, we simplify the problem by assuming that the objective function is of the form $f(z) = \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ})$, where f_{iJ} is a monotonically increasing, convex function, and that, as z_{iJ} is varied, z_{iJK}/z_{iJ} is constant for all $K \subset J$. Therefore, b_{iJK} is a constant for all $(i, J) \in \mathcal{A}$ and $K \subset J$. We also drop the constraint set Z, noting that separable constraints, at least, can be handled by making f_{iJ} approach infinity as z_{iJ} approaches its upper constraint. These assumptions apply if, at least from the perspective of the connection we wish to establish, links essentially behave independently and medium access issues do not pose significant constraints, either because they are non-existent or because they are dealt with separately. The assumptions certainly restrict the range of applicable cases, but they are not impractical; they apply, in particular, to all of the problems examined in our performance evaluation in Chapter 4.

With these assumptions, problem (3.2) becomes

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ}) \\ \text{subject to} \quad & \sum_{j\in K} x_{iJj}^{(t)} \le z_{iJ} b_{iJK}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J,\ t \in T, \\ & x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T. \end{aligned} \tag{3.6}$$

Since the f_{iJ} are monotonically increasing, the constraint

$$\sum_{j\in K} x_{iJj}^{(t)} \le z_{iJ} b_{iJK}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J,\ t \in T \tag{3.7}$$

gives

$$z_{iJ} = \max_{K\subset J, t\in T} \left\{ \frac{\sum_{j\in K} x_{iJj}^{(t)}}{b_{iJK}} \right\}. \tag{3.8}$$

Expression (3.8) is, unfortunately, not very useful for algorithm design because the

max function is difficult to deal with, largely as a result of its not being differentiable everywhere. One way to overcome this difficulty is to approximate z_{iJ} by replacing the max in (3.8) with an l^m-norm (see [31]), i.e., to approximate z_{iJ} with z'_{iJ}, where

$$z'_{iJ} := \left(\sum_{K \subset J, t \in T} \left(\frac{\sum_{j \in K} x^{(t)}_{iJj}}{b_{iJK}} \right)^m \right)^{1/m}.$$

The approximation becomes exact as $m \to \infty$. Moreover, since $z'_{iJ} \geq z_{iJ}$ for all $m > 0$, the coding subgraph z' admits the desired connection for any feasible solution.

Now the relevant optimization problem is

$$\begin{aligned} &\text{minimize} \sum_{(i,J) \in \mathcal{A}} f_{iJ}(z'_{iJ}) \\ &\text{subject to } x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T, \end{aligned}$$

which is no more than a convex multicommodity flow problem. There are many algorithms for convex multicommodity flow problems (see [84] for a survey), some of which (e.g., the algorithms in [8, 12]) are well-suited for distributed implementation. The primal-dual approach to internet congestion control (see [100, Section 3.4]) can also be used to solve convex multicommodity flow problems in a distributed manner, and we examine this method in Section 3.2.1.

There exist, therefore, numerous distributed algorithms for the subgraph selection problem—or, at least, for an approximation of the problem. What about distributed algorithms for the true problem? One clear tactic for finding such algorithms is to eliminate constraint (3.7) using Lagrange multipliers. Following this tactic, we obtain a distributed algorithm that we call the subgradient method. We describe the subgradient method in Section 3.2.2.

3.2.1 Primal-dual method

For the primal-dual method, we assume that the cost functions f_{iJ} are strictly convex and differentiable. Hence there is a unique optimal solution to problem (3.6). We present the algorithm for the lossless case, with the understanding that it can be straightforwardly extended to the lossy case. Thus, the optimization problem we address is

$$\begin{aligned} &\text{minimize} \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z'_{iJ}) \\ &\text{subject to } x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T, \end{aligned} \tag{3.9}$$

where

$$z'_{iJ} := \left(\sum_{t\in T} \left(\sum_{j\in J} x^{(t)}_{iJj} \right)^m \right)^{1/m}.$$

Let $(y)^+_a$ denote the following function of y:

$$(y)^+_a = \begin{cases} y & \text{if } a > 0, \\ \max\{y, 0\} & \text{if } a \le 0. \end{cases}$$

To solve problem (3.9) in a distributed fashion, we introduce additional variables p and λ and consider varying x, p, and λ in time τ according to the following time derivatives:

$$\dot{x}^{(t)}_{iJj} = -k^{(t)}_{iJj}(x^{(t)}_{iJj}) \left(\frac{\partial f_{iJ}(z'_{iJ})}{\partial x^{(t)}_{iJj}} + q^{(t)}_{ij} - \lambda^{(t)}_{iJj} \right), \tag{3.10}$$

$$\dot{p}^{(t)}_i = h^{(t)}_i(p^{(t)}_i)(y^{(t)}_i - \sigma^{(t)}_i), \tag{3.11}$$

$$\dot{\lambda}^{(t)}_{iJj} = m^{(t)}_{iJj}(\lambda^{(t)}_{iJj}) \left(-x^{(t)}_{iJj} \right)^+_{\lambda^{(t)}_{iJj}}, \tag{3.12}$$

where

$$q_{ij}^{(t)} := p_i^{(t)} - p_j^{(t)},$$
$$y_i^{(t)} := \sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)},$$

and $k_{iJj}^{(t)}(x_{iJj}^{(t)}) > 0$, $h_i^{(t)}(p_i^{(t)}) > 0$, and $m_{iJj}^{(t)}(\lambda_{iJj}^{(t)}) > 0$ are non-decreasing continuous functions of $x_{iJj}^{(t)}$, $p_i^{(t)}$, and $\lambda_{iJj}^{(t)}$ respectively.

Proposition 3.2. *The algorithm specified by Equations (3.10)–(3.12) is globally, asymptotically stable.*

Proof. We prove the stability of the primal-dual algorithm by using the theory of Lyapunov stability (see, e.g., [100, Section 3.10]). This proof is based on the proof of Theorem 3.7 of [100].

The Lagrangian for problem (3.9) is as follows:

$$\begin{aligned} L(x,p,\lambda) = &\sum_{(i,J)\in\mathcal{A}} f_{iJ}(z'_{iJ}) \\ &+ \sum_{t\in T} \left\{ \sum_{i\in\mathcal{N}} p_i^{(t)} \left(\sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)} - \sigma_i^{(t)} \right) \right. \\ &\qquad \left. - \sum_{(i,J)\in\mathcal{A}} \sum_{j\in J} \lambda_{iJj}^{(t)} x_{iJj}^{(t)} \right\}, \quad (3.13) \end{aligned}$$

where

$$\sigma_i^{(t)} = \begin{cases} R_t & \text{if } i = s, \\ -R_t & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

Since the objective function of problem (3.9) is strictly convex, it has a unique minimizing solution, say $\hat{x}$, and Lagrange multipliers, say $\hat{p}$ and $\hat{\lambda}$, which satisfy the

following Karush-Kuhn-Tucker conditions:

$$\frac{\partial L(\hat{x},\hat{p},\hat{\lambda})}{\partial x_{iJj}^{(t)}} = \left(\frac{\partial f_{iJ}(z'_{iJ})}{\partial x_{iJj}^{(t)}} + \left(\hat{p}_i^{(t)} - \hat{p}_j^{(t)}\right) - \hat{\lambda}_{iJj}^{(t)}\right) = 0, \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T, \tag{3.14}$$

$$\sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} \hat{x}_{iJj}^{(t)} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} \hat{x}_{jIi}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T, \tag{3.15}$$

$$\hat{x}_{iJj}^{(t)} \geq 0 \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T, \tag{3.16}$$

$$\hat{\lambda}_{iJj}^{(t)} \geq 0 \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T, \tag{3.17}$$

$$\hat{\lambda}_{iJj}^{(t)} \hat{x}_{iJj}^{(t)} = 0 \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T. \tag{3.18}$$

Using equation (3.13), we see that $(\hat{x},\hat{p},\hat{\lambda})$ is an equilibrium point of the primal-dual algorithm. We now prove that this point is globally, asymptotically stable.

Consider the following function as a candidate for the Lyapunov function:

$$\begin{aligned} &V(x,p,\lambda) \\ &= \sum_{t\in T} \left\{ \sum_{(i,J)\in\mathcal{A}} \sum_{j\in J} \left(\int_{\hat{x}_{iJj}^{(t)}}^{x_{iJj}^{(t)}} \frac{1}{k_{iJj}^{(t)}(\sigma)} (\sigma - \hat{x}_{iJj}^{(t)}) d\sigma + \int_{\hat{\lambda}_{iJj}^{(t)}}^{\lambda_{iJj}^{(t)}} \frac{1}{m_{iJj}^{(t)}(\gamma)} (\gamma - \hat{\lambda}_{iJj}^{(t)}) d\gamma \right) \right. \\ &\qquad \left. + \sum_{i\in\mathcal{N}} \int_{\hat{p}_i^{(t)}}^{p_i^{(t)}} \frac{1}{h_i^{(t)}(\beta)} (\beta - \hat{p}_i^{(t)}) d\beta \right\}. \end{aligned}$$

Note that $V(\hat{x},\hat{p},\hat{\lambda}) = 0$. Since, $k_{iJj}^{(t)}(\sigma) > 0$, if $x_{iJj}^{(t)} \neq \hat{x}_{iJj}^{(t)}$, we have $\int_{\hat{x}_{iJj}^{(t)}}^{x_{iJj}^{(t)}} \frac{1}{k_{iJj}^{(t)}(\sigma)} (\sigma - \hat{x}_{iJj}^{(t)}) d\sigma > 0$. This argument can be extended to the other terms as well. Thus, whenever $(x,p,\lambda) \neq (\hat{x},\hat{p},\hat{\lambda})$, we have $V(x,p,\lambda) > 0$.

Now,

$$\dot{V} = \sum_{t \in T} \left\{ \sum_{(i,J) \in \mathcal{A}} \sum_{j \in J} \left[\left(-x_{iJj}^{(t)} \right)^{+}_{\lambda_{iJj}^{(t)}} (\lambda_{iJj}^{(t)} - \hat{\lambda}_{iJj}^{(t)}) \right. \right.$$
$$\left. - \left(\frac{\partial f_{iJ}(z'_{iJ})}{\partial x_{iJj}^{(t)}} + q_{iJj}^{(t)} - \lambda_{iJj}^{(t)} \right) \cdot (x_{iJj}^{(t)} - \hat{x}_{iJj}^{(t)}) \right]$$
$$\left. + \sum_{i \in \mathcal{N}} (y_i^{(t)} - \sigma_i^{(t)})(p_i^{(t)} - \hat{p}_i^{(t)}) \right\}.$$

Note that

$$\left(-x_{iJj}^{(t)} \right)^{+}_{\lambda_{iJj}^{(t)}} (\lambda_{iJj}^{(t)} - \hat{\lambda}_{iJj}^{(t)}) \leq -x_{iJj}^{(t)} (\lambda_{iJj}^{(t)} - \hat{\lambda}_{iJj}^{(t)}),$$

since the inequality is an equality if either $x_{iJj}^{(t)} \leq 0$ or $\lambda_{iJj}^{(t)} \geq 0$; and, in the case when $x_{iJj}^{(t)} > 0$ and $\lambda_{iJj}^{(t)} < 0$, we have $(-x_{iJj}^{(t)})^{+}_{\lambda_{iJj}^{(t)}} = 0$ and, since $\hat{\lambda}_{iJj}^{(t)} \geq 0$, $-x_{iJj}^{(t)}(\lambda_{iJj}^{(t)} - \hat{\lambda}_{iJj}^{(t)}) \geq$

0. Therefore,

$$\begin{aligned}
\dot{V} \leq \sum_{t\in T}\Bigg\{ & \sum_{(i,J)\in\mathcal{A}}\sum_{j\in J}\Big[-x_{iJj}^{(t)}(\lambda_{iJj}^{(t)}-\hat{\lambda}_{iJj}^{(t)}) \\
& \qquad -\left(\frac{\partial f_{iJ}(z'_{iJ})}{\partial x_{iJj}^{(t)}}+q_{iJj}^{(t)}-\lambda_{iJj}^{(t)}\right)\cdot(x_{iJj}^{(t)}-\hat{x}_{iJj}^{(t)})\Big] \\
& \quad +\sum_{i\in\mathcal{N}}(y_i^{(t)}-\sigma_i^{(t)})(p_i^{(t)}-\hat{p}_i^{(t)})\Bigg\} \\
= {} & (\hat{q}-q)'(x-\hat{x})+(\hat{p}-p)'(y-\hat{y}) \\
& +\sum_{t\in T}\Bigg\{\sum_{(i,J)\in\mathcal{A}}\sum_{j\in J}\Big[-\hat{x}_{iJj}^{(t)}(\lambda_{iJj}^{(t)}-\hat{\lambda}_{iJj}^{(t)}) \\
& \qquad -\left(\frac{\partial f_{iJ}(z'_{iJ})}{\partial x_{iJj}^{(t)}}+\hat{q}_{iJj}^{(t)}-\hat{\lambda}_{iJj}^{(t)}\right)\cdot(x_{iJj}^{(t)}-\hat{x}_{iJj}^{(t)})\Big] \\
& \quad +\sum_{i\in\mathcal{N}}(\hat{y}_i^{(t)}-\sigma_i^{(t)})(p_i^{(t)}-\hat{p}_i^{(t)})\Bigg\} \\
= {} & \sum_{t\in T}\sum_{(i,J)\in\mathcal{A}}\sum_{j\in J}\left(\frac{\partial f_{iJ}(\hat{z}'_{iJ})}{\partial\hat{x}_{iJj}^{(t)}}-\frac{\partial f_{iJ}(z'_{iJ})}{\partial x_{iJj}^{(t)}}\right)(x_{iJj}^{(t)}-\hat{x}_{iJj}^{(t)})-\lambda'\hat{x},
\end{aligned}$$

where the last line follows from Karush-Kuhn-Tucker conditions (3.14)–(3.18) and the fact that

$$\begin{aligned}
p'y &= \sum_{t\in T}\sum_{i\in\mathcal{N}}p_i^{(t)}\left(\sum_{\{J|(i,J)\in\mathcal{A}\}}\sum_{j\in J}x_{iJj}^{(t)}-\sum_{\{j|(j,I)\in\mathcal{A},i\in I\}}x_{jIi}^{(t)}\right) \\
&= \sum_{t\in T}\sum_{(i,J)\in\mathcal{A}}\sum_{j\in J}x_{iJj}^{(t)}(p_i^{(t)}-p_j^{(t)})=q'x.
\end{aligned}$$

Thus, owing to the strict convexity of the functions $\{f_{iJ}\}$, we have $\dot{V}\leq-\lambda'\hat{x}$, with equality if and only if $x=\hat{x}$. So it follows that $\dot{V}\leq 0$ for all $\lambda\geq 0$, since $\hat{x}\geq 0$.

If the initial choice of λ is such that $\lambda(0)\geq 0$, we see from the primal-dual algorithm that $\lambda(\tau)\geq 0$. This is true since $\dot{\lambda}\geq 0$ whenever $\lambda\leq 0$. Thus, it follows by

the theory of Lyapunov stability that the algorithm is indeed globally, asymptotically stable. □

The global, asymptotic stability of the algorithm implies that no matter what the initial choice of (x, p) is, the primal-dual algorithm will converge to the unique solution of problem (3.9). We have to choose λ, however, with non-negative entries as the initial choice. Further, there is no guarantee that $x(\tau)$ yields a feasible solution for any given τ. Therefore, a start-up time may be required before a feasible solution is obtained.

The algorithm that we currently have is a continuous time algorithm and, in practice, an algorithm operating in discrete message exchanges is required. To discretize the algorithm, we consider time steps $n = 0, 1, \ldots$ and replace the derivatives by differences:

$$x_{iJj}^{(t)}[n+1] = x_{iJj}^{(t)}[n] - \alpha_{iJj}^{(t)}[n] \left(\frac{\partial f_{iJ}(z'_{iJ}[n])}{\partial x_{iJj}^{(t)}[n]} + q_{ij}^{(t)}[n] - \lambda_{iJj}^{(t)}[n] \right), \tag{3.19}$$

$$p_i^{(t)}[n+1] = p_i^{(t)}[n] + \beta_i^{(t)}[n](y_i^{(t)}[n] - \sigma_i^{(t)}), \tag{3.20}$$

$$\lambda_{iJj}^{(t)}[n+1] = \lambda_{iJj}^{(t)}[n] + \gamma_{iJj}^{(t)}[n] \left(-x_{iJj}^{(t)}[n] \right)_{\lambda_{iJj}^{(t)}[n]}^{+}, \tag{3.21}$$

where

$$q_{ij}^{(t)}[n] := p_i^{(t)}[n] - p_j^{(t)}[n],$$

$$y_i^{(t)}[n] := \sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)}[n] - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)}[n],$$

and $\alpha_{iJj}^{(t)}[n] > 0$, $\beta_i^{(t)}[n] > 0$, and $\gamma_{iJj}^{(t)}[n] > 0$ are step sizes. This discretized algorithm operates in synchronous rounds, with nodes exchanging information in each round. We expect that this synchronicity can be relaxed in practice, but this issue remains to be investigated.

We associate a processor with each node. We assume that the processor for node

1. Each node i initializes $p_i[0]$, $\{x_{iJj}[0]\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$, and $\{\lambda_{iJj}[0]\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$ such that $\lambda_{iJj}[0] \geq 0$ for all (J,j) such that $(i,J) \in \mathcal{A}$ and $j \in J$. Each node i sends $p_i[0]$, $\{x_{iJj}[0]\}_{j\in J}$, and $\{\lambda_{iJj}[0]\}_{j\in J}$ over each outgoing hyperarc (i,J).

2. At the nth iteration, each node i computes $p_i[n+1]$, $\{x_{iJj}[n+1]\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$, and $\{\lambda_{iJj}[n+1]\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$ using equations (3.19)–(3.21). Each node i sends $p_i[n+1]$, $\{x_{iJj}[n+1]\}_{j\in J}$, and $\{\lambda_{iJj}[n+1]\}_{j\in J}$ over each outgoing hyperarc (i,J).

3. The current coding subgraph $z'[n]$ is computed. For each node i, we set
$$z'_{iJ}[n] := \left(\sum_{t\in T}\left(\sum_{j\in J} x^{(t)}_{iJj}[n]\right)^m\right)^{1/m}$$
for all outgoing hyperarcs (i,J).

4. Steps 2 and 3 are repeated until the sequence of coding subgraphs $\{z'[n]\}$ converges.

Figure 3.4: Summary of the primal-dual method.

i keeps track of the variables p_i, $\{x_{iJj}\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$, and $\{\lambda_{iJj}\}_{\{J,j|(i,J)\in\mathcal{A},j\in J\}}$. With such an assignment of variables to processors, the algorithm is distributed in the sense that a node exchanges information only with its neighbors at every iteration of the primal-dual algorithm. We summarize the primal-dual method in Figure 3.4.

3.2.2 Subgradient method

We present the subgradient method for linear cost functions; with some modifications, it may be made to apply also to convex ones. Thus, we assume that the objective function f is of the form
$$f(z) := \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ},$$
where $a_{iJ} > 0$.

Consider the Lagrangian dual of problem (3.6):

$$\begin{aligned} \text{maximize} \quad & \sum_{t\in T} q^{(t)}(p^{(t)}) \\ \text{subject to} \quad & \sum_{t\in T}\sum_{K\subset J} p^{(t)}_{iJK} = a_{iJ} \qquad \forall\ (i,J)\in\mathcal{A}, \\ & p^{(t)}_{iJK} \geq 0, \qquad \forall\ (i,J)\in\mathcal{A},\ K\subset J,\ t\in T, \end{aligned} \tag{3.22}$$

where

$$q^{(t)}(p^{(t)}) := \min_{x^{(t)}\in F^{(t)}} \sum_{(i,J)\in\mathcal{A}} \sum_{j\in J} \left(\sum_{\{K\subset J | K\ni j\}} \frac{p^{(t)}_{iJK}}{b_{iJK}} \right) x_{iJj}. \tag{3.23}$$

In the lossless case, the dual problem defined by equations (3.22) and (3.23) simplifies somewhat, and we require only a single dual variable $p^{(t)}_{iJJ}$ for each hyperarc (i,J). In the case that relates to optimization problem (3.4), the dual problem simplifies more still, as there are fewer primal variables associated with it. Specifically, we obtain, for the Lagrangian dual,

$$\begin{aligned} \text{maximize} \quad & \sum_{t\in T} \hat{q}^{(t)}(p^{(t)}) \\ \text{subject to} \quad & \sum_{t\in T} p^{(t)}_{iJ^{(i)}_m} = s_{iJ^{(i)}_m}, \qquad \forall\ i\in\mathcal{N},\ m=1,\ldots,M_i, \\ & p^{(t)}_{iJ} \geq 0, \qquad \forall\ (i,J)\in\mathcal{A},\ t\in T, \end{aligned} \tag{3.24}$$

where

$$s_{iJ^{(i)}_m} := a_{iJ^{(i)}_m} - a_{iJ^{(i)}_{m-1}},$$

and

$$\hat{q}^{(t)}(p^{(t)}) := \min_{\hat{x}^{(t)}\in\hat{F}^{(t)}} \sum_{(i,j)\in\mathcal{A}'} \left(\sum_{m=1}^{m(i,j)} p^{(t)}_{iJ^{(i)}_m} \right) \hat{x}^{(t)}_{ij}. \tag{3.25}$$

Note that, by the assumptions of the problem, $s_{iJ} > 0$ for all $(i,J)\in\mathcal{A}$.

In all three cases, the dual problems are very similar, and essentially the same algorithm can be used to solve them. We present the subgradient method for the

case that relates to optimization problem (3.4)—namely, the primal problem

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ} \\ \text{subject to} \quad & \sum_{k\in J_{M_i}^{(i)}\setminus J_{m-1}^{(i)}} \hat{x}_{ik}^{(t)} \leq \sum_{n=m}^{M_i} z_{iJ_n^{(i)}}, \qquad \forall\ i\in\mathcal{N},\ m=1,\ldots,M_i,\ t\in T, \\ & \hat{x}^{(t)} \in \hat{F}^{(t)}, \qquad \forall\ t\in T \end{aligned} \tag{3.26}$$

with dual (3.24)—with the understanding that straightforward modifications can be made for the other cases.

We first note that problem (3.25) is, in fact, a shortest path problem, which admits a simple, asynchronous distributed solution known as the distributed asynchronous Bellman-Ford algorithm (see, e.g., [13, Section 5.2.4]).

Now, to solve the dual problem (3.24), we employ subgradient optimization (see, e.g., [9, Section 6.3.1] or [83, Section I.2.4]). We start with an iterate $p[0]$ in the feasible set of (3.24) and, given an iterate $p[n]$ for some non-negative integer n, we solve problem (3.25) for each t in T to obtain $x[n]$. Let

$$g_{iJ_m^{(i)}}^{(t)}[n] := \sum_{k\in J_{M_i}^{(i)}\setminus J_{m-1}^{(i)}} \hat{x}_{ik}^{(t)}[n].$$

We then assign

$$p_{iJ}[n+1] := \operatorname*{arg\,min}_{v\in P_{iJ}} \sum_{t\in T} (v^{(t)} - (p_{iJ}^{(t)}[n] + \theta[n] g_{iJ}^{(t)}[n]))^2 \tag{3.27}$$

for each $(i,J)\in\mathcal{A}$, where P_{iJ} is the $|T|$-dimensional simplex

$$P_{iJ} = \left\{ v \,\middle|\, \sum_{t\in T} v^{(t)} = s_{iJ},\ v \geq 0 \right\}$$

and $\theta[n] > 0$ is an appropriate step size. In other words, $p_{iJ}[n+1]$ is set to be the

Euclidean projection of $p_{iJ}[n] + \theta[n]g_{iJ}[n]$ onto P_{iJ}.

To perform the projection, we use the following proposition.

Proposition 3.3. *Let $u := p_{iJ}[n] + \theta[n]g_{iJ}[n]$. Suppose we index the elements of T such that $u^{(t_1)} \geq u^{(t_2)} \geq \ldots \geq u^{(t_{|T|})}$. Take $\hat{k}$ to be the smallest k such that*

$$\frac{1}{k}\left(s_{iJ} - \sum_{r=1}^{t_k} u^{(r)}\right) \leq -u^{(t_{k+1})}$$

or set $\hat{k} = |T|$ if no such k exists. Then the projection (3.27) is achieved by

$$p_{iJ}^{(t)}[n+1] = \begin{cases} u^{(t)} + \frac{s_{iJ} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}}{\hat{k}} & \text{if } t \in \{t_1, \ldots, t_{\hat{k}}\}, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We wish to solve the following problem.

$$\begin{aligned} &\text{minimize} \sum_{t \in T} (v^{(t)} - u^{(t)})^2 \\ &\text{subject to } v \in P_{iJ}. \end{aligned}$$

First, since the objective function and the constraint set P_{iJ} are both convex, it is straightforward to establish that a necessary and sufficient condition for global optimality of $\hat{v}^{(t)}$ in P_{iJ} is

$$\hat{v}^{(t)} > 0 \Rightarrow (u^{(t)} - \hat{v}^{(t)}) \geq (u^{(r)} - \hat{v}^{(r)}), \qquad \forall\ r \in T \tag{3.28}$$

(see, e.g., [9, Section 2.1]). Suppose we index the elements of T such that $u^{(t_1)} \geq u^{(t_2)} \geq \ldots \geq u^{(t_{|T|})}$. We then note that there must be an index k in the set $\{1, \ldots, |T|\}$ such that $v^{(t_l)} > 0$ for $l = 1, \ldots, k$ and $v^{(t_l)} = 0$ for $l > k + 1$, for, if not, then a feasible solution with lower cost can be obtained by swapping around components of the vector. Therefore, condition (3.28) implies that there must exist some d such that

$\hat{v}^{(t)} = u^{(t)} + d$ for all $t \in \{t_1, \ldots, t_k\}$ and that $d \leq -u^{(t)}$ for all $t \in \{t_{k+1}, \ldots, t_{|T|}\}$, which is equivalent to $d \leq -u^{(t_{k+1})}$. Since $\hat{v}^{(t)}$ is in the simplex P_{iJ}, it follows that

$$kd + \sum_{t=1}^{t_k} u^{(t)} = s_{iJ},$$

which gives

$$d = \frac{1}{k}\left(s_{iJ} - \sum_{t=1}^{t_k} u^{(t)}\right).$$

By taking $k = \hat{k}$, where $\hat{k}$ is the smallest k such that

$$\frac{1}{k}\left(s_{iJ} - \sum_{r=1}^{t_k} u^{(r)}\right) \leq -u^{(t_{k+1})},$$

(or, if no such k exists, then $\hat{k} = |T|$), we see that we have

$$\frac{1}{\hat{k}-1}\left(s_{iJ} - \sum_{t=1}^{t_{k-1}} u^{(t)}\right) > -u^{(t_k)},$$

which can be rearranged to give

$$d = \frac{1}{\hat{k}}\left(s_{iJ} - \sum_{t=1}^{t_k} u^{(t)}\right) > -u^{(t_k)}.$$

Hence, if $v^{(t)}$ is given by

$$v^{(t)} = \begin{cases} u^{(t)} + \frac{s_{iJ} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}}{\hat{k}} & \text{if } t \in \{t_1, \ldots, t_{\hat{k}}\}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.29}$$

then $v^{(t)}$ is feasible and we see that the optimality condition (3.28) is satisfied. Note

that, since $d \leq -u^{(t_{k+1})}$, equation (3.29) can also be written as

$$v^{(t)} = \max\left(0, u^{(t)} + \frac{1}{\hat{k}}\left(s_{iJ} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}\right)\right). \tag{3.30}$$

□

The disadvantage of subgradient optimization is that, whilst it yields good approximations of the optimal value of the Lagrangian dual problem (3.24) after sufficient iteration, it does not necessarily yield a primal optimal solution. There are, however, methods for recovering primal solutions in subgradient optimization. We employ the following method, which is due to Sherali and Choi [95].

Let $\{\mu_l[n]\}_{l=1,\ldots,n}$ be a sequence of convex combination weights for each non-negative integer n, i.e., $\sum_{l=1}^{n} \mu_l[n] = 1$ and $\mu_l[n] \geq 0$ for all $l = 1, \ldots, n$. Further, let us define

$$\gamma_{ln} := \frac{\mu_l[n]}{\theta[n]}, \quad l = 1, \ldots, n,\ n = 0, 1, \ldots,$$

and

$$\Delta\gamma_n^{\max} := \max_{l=2,\ldots,n} \{\gamma_{ln} - \gamma_{(l-1)n}\}.$$

Proposition 3.4. *If the step sizes $\{\theta[n]\}$ and convex combination weights $\{\mu_l[n]\}$ are chosen such that*

1. *$\gamma_{ln} \geq \gamma_{(l-1)n}$ for all $l = 2, \ldots, n$ and $n = 0, 1, \ldots$,*
2. *$\Delta\gamma_n^{\max} \to 0$ as $n \to \infty$, and*
3. *$\gamma_{1n} \to 0$ as $n \to \infty$ and $\gamma_{nn} \leq \delta$ for all $n = 0, 1, \ldots$ for some $\delta > 0$,*

then we obtain an optimal solution to the primal problem from any accumulation point of the sequence of primal iterates $\{\tilde{x}[n]\}$ given by

$$\tilde{x}[n] := \sum_{l=1}^{n} \mu_l[n]\hat{x}[l], \quad n = 0, 1, \ldots. \tag{3.31}$$

Proof. Suppose that the dual feasible solution that the subgradient method converges to is $\bar{p}$. Then, using (3.27), there exists some m such that for $n \geq m$

$$p_{iJ}^{(t)}[n+1] = p_{iJ}^{(t)}[n] + \theta[n] g_{iJ}^{(t)}[n] + c_{iJ}[n]$$

for all $(i, J) \in \mathcal{A}$ and $t \in T$ such that $\bar{p}_{iJ}^{(t)} > 0$.

Let $\tilde{g}[n] := \sum_{l=1}^{n} \mu_l[n] g[l]$. Consider some $(i, J) \in \mathcal{A}$ and $t \in T$. If $\bar{p}_{iJ}^{(t)} > 0$, then for $n > m$ we have

$$\begin{aligned}
\tilde{g}_{iJ}^{(t)}[n] &= \sum_{l=1}^{m} \mu_l[n] g_{iJ}^{(t)}[l] + \sum_{l=m+1}^{n} \mu_l[n] g_{iJ}^{(t)}[l] \\
&= \sum_{l=1}^{m} \mu_l[n] g_{iJ}^{(t)}[l] + \sum_{l=m+1}^{n} \frac{\mu_l[n]}{\theta[n]} (p_{iJ}^{(t)}[n+1] - p_{iJ}^{(t)}[n] - c_{iJ}[n]) \\
&= \sum_{l=1}^{m} \mu_l[n] g_{iJ}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln} (p_{iJ}^{(t)}[n+1] - p_{iJ}^{(t)}[n]) - \sum_{l=m+1}^{n} \gamma_{ln} c_{iJ}[n].
\end{aligned} \tag{3.32}$$

Otherwise, if $\bar{p}_{iJ}^{(t)} = 0$, then from equation (3.30), we have

$$p_{iJ}^{(t)}[n+1] \geq p_{iJ}^{(t)}[n] + \theta[n] g_{iJ}^{(t)}[n] + c_{iJ}[n],$$

so

$$\tilde{g}_{iJ}^{(t)}[n] \leq \sum_{l=1}^{m} \mu_l[n] g_{iJ}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln} (p_{iJ}^{(t)}[n+1] - p_{iJ}^{(t)}[n]) - \sum_{l=m+1}^{n} \gamma_{ln} c_{iJ}[n]. \tag{3.33}$$

It is straightforward to see that the sequence of iterates $\{\tilde{x}[n]\}$ is primal feasible, and that we obtain a primal feasible sequence $\{z[n]\}$ by setting

$$\begin{aligned}
z_{iJ_m^{(i)}}[n] &:= \max_{t \in T} \left\{ \sum_{k \in J_{M_i}^{(i)} \setminus J_{m-1}^{(i)}} \tilde{x}_{ik}^{(t)}[n] \right\} - \sum_{m'=m+1}^{M_i} z_{iJ_{m'}^{(i)}}[n] \\
&= \max_{t \in T} \tilde{g}_{iJ_m^{(i)}} - \sum_{m'=m+1}^{M_i} z_{iJ_{m'}^{(i)}}[n]
\end{aligned}$$

recursively, starting from $m = M_i$ and proceeding through to $m = 1$. Sherali and Choi [95] showed that, if the required conditions on the step sizes $\{\theta[n]\}$ and convex combination weights $\{\mu_l[n]\}$ are satisfied, then

$$\sum_{l=1}^{m} \mu_l[n] g_{iJ}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln}(p_{iJ}^{(t)}[n+1] - p_{iJ}^{(t)}[n]) \to 0$$

as $k \to \infty$; hence we see from equations (3.32) and (3.33) that, for k sufficiently large,

$$\sum_{m'=m}^{M_i} z_{iJ_{m'}^{(i)}}[n] = -\sum_{l=m+1}^{n} \gamma_{ln} c_{iJ_m^{(i)}}[n].$$

Recalling the primal problem (3.26), we see that complementary slackness with $\bar{p}$ holds in the limit of any convergent subsequence of $\{\tilde{x}[n]\}$. □

The required conditions on the step sizes and convex combination weights are satisfied by the following choices ([95, Corollaries 2–4]):

1. step sizes $\{\theta[n]\}$ such that $\theta[n] > 0$, $\lim_{n\to 0} \theta[n] = 0$, $\sum_{n=1}^{\infty} \theta_n = \infty$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = \theta[l] / \sum_{k=1}^{n} \theta[k]$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$;

2. step sizes $\{\theta[n]\}$ given by $\theta[n] = a/(b + cn)$ for all $n = 0, 1, \ldots$, where $a > 0$, $b \geq 0$ and $c > 0$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = 1/n$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$; and

3. step sizes $\{\theta[n]\}$ given by $\theta[n] = n^{-\alpha}$ for all $n = 0, 1, \ldots$, where $0 < \alpha < 1$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = 1/n$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$.

Moreover, for all three choices, we have $\mu_l[n+1]/\mu_l[n]$ independent of l for all n, so

primal iterates can be computed iteratively using

$$\begin{aligned}\tilde{x}[n] &= \sum_{l=1}^{n} \mu_l[n]\hat{x}[l] \\ &= \sum_{l=1}^{n-1} \mu_l[n]\hat{x}[l] + \mu_n[n]\hat{x}[n] \\ &= \phi[n-1]\tilde{x}[n-1] + \mu_n[n]\hat{x}[n],\end{aligned}$$

where $\phi[n] := \mu_l[n+1]/\mu_l[n]$.

This gives us our distributed algorithm. We summarize the subgradient method in Figure 3.5. We see that, although the method is indeed a distributed algorithm, it again operates in synchronous rounds. Again, we expect that this synchronicity can be relaxed in practice, but this issue remains to be investigated.

3.3 Dynamic multicast

In many applications, membership of the multicast group changes in time, with nodes joining and leaving the group, rather than remaining constant for the duration of the connection, as we have thus far assumed. Under these dynamic conditions, we often cannot simply re-establish the connection with every membership change because doing so would cause an unacceptable disruption in the service being delivered to those nodes remaining in the group. A good example of an application where such issues arise is real-time media distribution. Thus, we desire to find minimum-cost time-varying subgraphs that can deliver continuous service to dynamic multicast groups.

Although our objective is clear, our description of the problem is currently vague. Indeed, one of the principal hurdles to tackling the problem of dynamic multicast lies in formulating the problem in such a way that it is suitable for analysis and addresses our objective. For routed networks, the problem is generally formulated as the dynamic Steiner tree problem, which was first proposed in [52]. Under this formulation, the focus is on worst-case behavior and modifications of the multicast

1. Each node i computes s_{iJ} for its outgoing hyperarcs and initializes $p_{iJ}[0]$ to a point in the feasible set of (3.24). For example, we take $p_{iJ}^{(t)}[0] := s_{iJ}/|T|$. Each node i sends s_{iJ} and $p_{iJ}[0]$ over each outgoing hyperarc (i, J).

2. At the nth iteration, use $p^{(t)}[n]$ as the hyperarc costs and run a distributed shortest path algorithm, such as distributed Bellman-Ford, to determine $\hat{x}^{(t)}[n]$ for all $t \in T$.

3. Each node i computes $p_{iJ}[n+1]$ for its outgoing hyperarcs using Proposition 3.3. Each node i sends $p_{iJ}[n+1]$ over each outgoing hyperarc (i, J).

4. Nodes compute the primal iterate $\tilde{x}[n]$ by setting

$$\tilde{x}[n] := \sum_{l=1}^{n} \mu_l[n]\hat{x}[l].$$

5. The current coding subgraph $z[n]$ is computed using the primal iterate $\tilde{x}[n]$. For each node i, we set

$$z_{iJ_m^{(i)}}[n] := \max_{t \in T} \left\{ \sum_{k \in J_{M_i}^{(i)} \setminus J_{m-1}^{(i)}} \tilde{x}_{ik}^{(t)}[n] \right\} - \sum_{m'=m+1}^{M_i} z_{iJ_{m'}^{(i)}}[n]$$

   recursively, starting from $m = M_i$ and proceeding through to $m = 1$.

6. Steps 2–5 are repeated until the sequence of primal iterates $\{\tilde{x}[n]\}$ converges.

Figure 3.5: Summary of the subgradient method.

tree are allowed only when nodes join or leave the multicast group. The formulation is adequate, but not compelling—indeed, there is no compelling reason for the restriction on when the multicast tree can be modified.

In our formulation for coded networks, we draw some inspiration from [52], but we focus on expected behavior rather than worst-case behavior, and we do not restrict modifications of the multicast subgraph to when nodes join or leave the multicast tree. We formulate the problem as follows.

We employ a basic unit of time that is related to the time that it takes for changes in the multicast subgraph to settle. In particular, suppose that at a given time the multicast subgraph is z and that it is capable of supporting a multicast connection to sink nodes T. Then, in one unit time, we can change the multicast subgraph to z', which is capable of supporting a multicast connection to sink nodes T', without disrupting the service being delivered to $T \cap T'$ provided that (componentwise) $z \geq z'$ or $z \leq z'$. The interpretation of this assumption is that we allow, in one time unit, only for the subgraph to increase, meaning that any sink node receiving a particular stream will continue to receive it (albeit with possible changes in the code, depending on how the coding is implemented) and therefore facing no significant disruption to service; or for the subgraph to decrease, meaning that any sink node receiving a particular stream will be forced to reduce to a subset of that stream, but one that is sufficient to recover the source's transmission provided that the sink node is in T', and therefore again facing no significant disruption to service. We do not allow for both operations to take place in a single unit of time (which would allow for arbitrary changes) because, in that case, sink nodes may face temporary disruptions to service when decreases to the multicast subgraph follow too closely to increases.

As an example, consider the four-node lossless network shown in Figure 3.6. Suppose that $s = 1$ and that, at a given time, we have $T = \{2, 4\}$. We support a multicast of unit rate with the subgraph

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (1, 0, 1, 0).$$

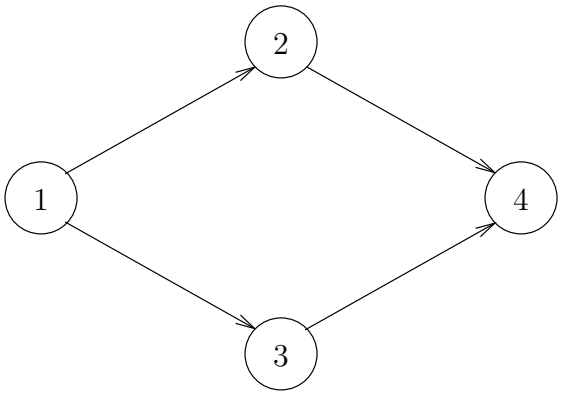

Figure 3.6: A four-node lossless network.

Now suppose that the group membership changes, and node 2 leaves while node 3 joins, so $T' = \{3, 4\}$. As a result, we decide that we wish to change to the subgraph

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (0, 1, 0, 1).$$

If we simply make the change naïvely in a single time unit, then node 4 may face a temporary disruption to its service because packets on $(2, 4)$ may stop arriving before packets on $(3, 4)$ start arriving. The assumption that we have made on allowed operations ensures that we must first increase the subgraph to

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (1, 1, 1, 1),$$

allow for the change to settle by waiting for one time unit, then decrease the subgraph to

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (0, 1, 0, 1).$$

With this series of operations, node 4 maintains continuous service throughout the subgraph change.

We discretize the time axis into time intervals of a single time unit. We suppose that, at the beginning of each time interval, we receive zero or more requests from sink nodes that are not currently part of the multicast group to join and zero or more

requests from sink nodes that are currently part of the multicast group to leave. We model these join and leave requests as a discrete stochastic process and make the assumption that, once all the members of the multicast group leave, the connection is over and remains in that state forever. Let T_m denote the sink nodes in the multicast group at the end of time interval m. Then, we assume that

$$\lim_{m\to\infty} \Pr(T_m \neq \emptyset | T_0 = T) = 0 \tag{3.34}$$

for any initial multicast group T. A possible, simple model of join and leave requests is to model $|T_m|$ as a birth-death process with a single absorbing state at state 0, and to choose a node uniformly from $\mathcal{N}' \setminus T_m$, where $\mathcal{N}' := \mathcal{N} \setminus \{s\}$, at each birth and from T_m at each death.

Let $z^{(m)}$ be the multicast subgraph at the beginning of time interval m, which, by the assumptions made thus far, means that it supports a multicast connection to sink nodes T_{m-1}. Let V_{m-1} and W_{m-1} be the join and leave requests that arrive at the end of time interval $m-1$, respectively. Hence, $V_{m-1} \subset \mathcal{N}' \setminus T_{m-1}$, $W_{m-1} \subset T_{m-1}$, and $T_m = (T_{m-1} \setminus W_{m-1}) \cup V_{m-1}$. We choose $z^{(m+1)}$ from $z^{(m)}$ and T_m using the function μ_m, so $z^{(m+1)} = \mu_m(z^{(m)}, T_m)$, where $z^{(m+1)}$ must lie in a particular constraint set $U(z^{(m)}, T_m)$.

To characterize the constraint set $U(z, T)$, recall the optimization problem for minimum-cost multicast in Section 3.1:

$$\begin{aligned} &\text{minimize } f(z) \\ &\text{subject to } z \in Z, \\ &\qquad \sum_{j\in K} x_{iJj}^{(t)} \leq z_{iJ} b_{iJK}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J,\ t \in T, \\ &\qquad x^{(t)} \in F^{(t)}, \qquad \forall\ t \in T. \end{aligned} \tag{3.35}$$

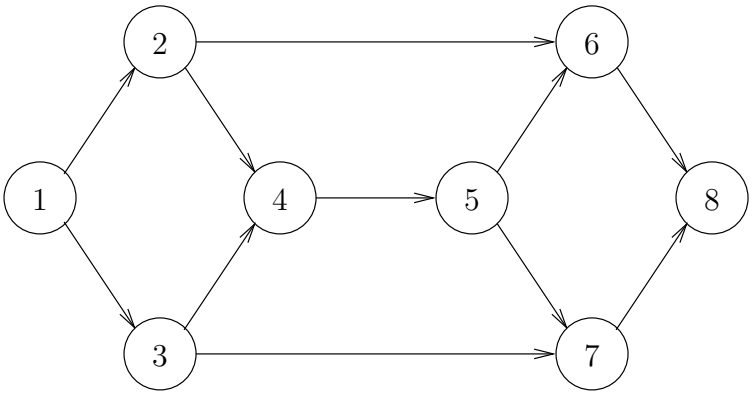

Figure 3.7: A lossless network used for dynamic multicast.

Therefore, it follows that we can write $U(z,T) = U_+(z,T) \cup U_-(z,T)$, where

$$U_+(z,T) = \{z' \in Z(T) | z' \geq z\},$$
$$U_-(z,T) = \{z' \in Z(T) | z' \leq z\},$$

and $Z(T)$ is the feasible set of z in problem (3.35) for a given T, i.e., if we have the subgraph z at the beginning of a time interval and we must go to a subgraph that supports multicast to T, then the allowable subgraphs are those that support multicast to T and either increase z (those in $U_+(z,T)$) or decrease z (those in $U_-(z,T)$).

Note that, if we have separable constraints, then $U(z^{(m)}, T_m) \neq \emptyset$ for all $z^{(m)} \in Z$ provided that $Z(T_m) \neq \emptyset$, i.e., from any feasible subgraph at stage m, it is possible to go to a feasible subgraph at stage $m+1$ provided that one exists for the multicast group T_m. But while this is the case for coded networks, it is not always the case for routed networks. Indeed, if multiple multicast trees are being used (as discussed in [109], for example), then it is definitely possible to find ourselves in a state where we cannot achieve multicast at stage $m+1$ even though static multicast to T_m is possible using multiple multicast trees.

As an example of this phenomenon, consider the lossless network depicted in Figure 3.7. Suppose that each arc is of unit capacity, that $s = 1$, and that, at a given time, we have $T = \{6, 8\}$. We support a multicast of rate 2 with the two

trees $\{(1,3),(3,4),(4,5),(5,6),(5,7),(7,8)\}$ and $\{(1,2),(2,6),(6,8)\}$, each carrying unit rate. Now suppose that the group membership changes, and node 6 leaves while node 7 joins, so $T' = \{7,8\}$. It is clear that static multicast to T' is possible using multiple multicast trees (we simply reflect the solution for T), but we cannot achieve multicast to T' by only adding edges to the two existing trees. Our only recourse at this stage is to abandon the existing trees and establish new ones, which causes a disruption to node 8's service, or to reconfigure slowly the existing trees, which causes a delay before node 7 is actually joined to the group.

Returning to the problem at hand, our objective is to find a policy $\pi = \{\mu_0, \mu_1, \dots, \}$ that minimizes the cost function

$$J_\pi(z^{(0)}, T_0) = \lim_{M\to\infty} \mathbb{E}\left[\sum_{m=0}^{M-1} f(z^{(m+1)})\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T_m)\right],$$

where $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}$ is the characteristic function for $2^{\mathcal{N}'} \setminus \{\emptyset\}$ (i.e., $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T) = 1$ if $T \neq \emptyset$, and $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T) = 0$ if $T = \emptyset$).

We impose the assumption that we have separable constraints and that $Z(\mathcal{N}') \neq \emptyset$, i.e., we assume that there exists a subgraph that supports broadcast. This assumption ensures that the constraint set $U(z,T)$ is non-empty for all $z \in Z$ and $T \subset \mathcal{N}'$. Thus, from condition (3.34), it follows that there exists at least one policy π such that $J_\pi(z^{(0)}, T_0) < \infty$ (namely, one that uses some fixed $z \in Z(\mathcal{N}')$ until the multicast group is empty).

It is now not difficult to see that we are dealing with an undiscounted, infinite-horizon dynamic programming problem (see, e.g., [11, Chapter 3]), and we can apply the theory developed for such problems to our problem. So doing, we first note that the optimal cost function $J^* := \min_\pi J_\pi$ satisfies Bellman's equation, namely, we have

$$J^*(z,T) = \min_{u\in U(z,T)} \{f(u) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)]\}$$

if $T \neq \emptyset$, and $J^*(z,T) = 0$ if $T = \emptyset$. Moreover, the optimal cost is achieved by the

stationary policy $\pi = \{\mu, \mu, \ldots\}$, where μ is given by

$$\mu(z, T) = \arg\min_{u \in U(z,T)} \{f(u) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)]\} \tag{3.36}$$

if $T \neq \emptyset$, and $\mu(z, T) = 0$ if $T = \emptyset$.

The fact that the optimal cost can be achieved by a stationary policy limits the space in which we need to search for optimal policies significantly, but we are still left with the difficulty that the state space is uncountably large—it is the space of all possible pairs (z, T), which is $Z \times 2^{\mathcal{N}'}$. The size of the state space more or less eliminates the possibility of using techniques such as value iteration to obtain J^*.

On the other hand, given J^*, it does not seem at all implausible that we can compute the optimal decision at the beginning of each time interval using (3.36). The constraint set is the union of two polyhedra, which can simply be handled by optimizing over each separately. The objective function can pose a difficulty because, even if f is convex, it may not necessarily be convex owing to the term $\mathbb{E}[J^*(u, (T \setminus V) \cup W)]$. But, since we are unable to obtain J^* precisely on account of the large state space, we can restrict our attention to approximations that make problem (3.36) tractable.

For dynamic programming problems, there are many approximations that have been developed to cope with large state spaces (see, e.g., [11, Section 2.3.3]). In particular, we can approximate $J^*(z, T)$ by $\tilde{J}(z, T, r)$, where $\tilde{J}(z, T, r)$ is of some fixed form, and r is a parameter vector that is determined by some form of optimization, which can be performed offline if the graph $\mathcal{G}$ is static. Depending upon the approximation that is used, we may even be able to solve problem (3.36) using the distributed algorithms described in Section 3.2 (or simple modifications thereof). The specific approximations $\tilde{J}(z, T, r)$ that we can use and their performance are beyond the scope of this thesis.

Chapter 4

Performance Evaluation

In the preceding two chapters, we laid out a solution to the efficient operation problem for coded packet networks. The solution we described has several attractive properties. In particular, it can be computed in a distributed manner and, in many cases, it is possible to solve the problem, as we have defined it in Section 1.2, optimally for a single multicast connection. But the question remains, is it actually useful? Is there a compelling reason to abandon the routed approach, with which we have so much experience, in favor of a new one?

We believe that for some applications the answer to both questions is yes and, in this chapter, we report on the results of several simulations that we conducted to assess the performance of the proposed techniques in situations of interest. Specifically, we consider three problems:

1. minimum-transmission wireless unicast: the problem of establishing a unicast connection in a lossy wireless network using the minimum number of transmissions per message packet;

2. minimum-weight wireline multicast: the problem of establishing a multicast connection in a lossless wireline network using the minimum weight, or artificial cost, per message packet;

3. minimum-energy wireless multicast: the problem of establishing a multicast connection in a lossless wireless network using the minimum amount of energy per message packet.

We deal with these problems in Sections 4.1, 4.2, and 4.3, respectively. We find that lossy wireless networks generally offer the most potential for the proposed techniques to improve on existing ones and that these improvements can indeed be significant.

4.1 Minimum-transmission wireless unicast

Establishing a unicast connection in a lossy wireless network is not trivial. Packets are frequently lost, and some mechanism to ensure reliable communication is required. Such a mechanism should not send packets unnecessarily, and we therefore consider the objective of minimizing the total number of transmissions per message packet.

There are numerous approaches to wireless unicast; we consider five, three of which (approaches 1–3) are routed approaches and two of which (approaches 4 and 5) are coded approaches:

1. **End-to-end retransmission:** A path is chosen from source to sink, and packets are acknowledged by the sink, or destination node. If the acknowledgment for a packet is not received by the source, the packet is retransmitted. This represents the situation where reliability is provided by a retransmission scheme above the link layer, e.g., by the transmission control protocol (TCP) at the transport layer, and no mechanism for reliability is present at the link layer.

2. **End-to-end coding:** A path is chosen from source to sink, and an end-to-end forward error correction (FEC) code, such as a Reed-Solomon code, an LT code [69], or a Raptor code [81, 96], is used to correct for packets lost between source and sink. This is the Digital Fountain approach to reliability [19].

3. **Link-by-link retransmission:** A path is chosen from source to sink, and ARQ

is used at the link layer to request the retransmission of packets lost on every link in the path. Thus, on every link, packets are acknowledged by the intended receiver and, if the acknowledgment for a packet is not received by the sender, the packet is retransmitted.

4. **Path coding:** A path is chosen from source to sink, and every node on the path employs coding to correct for lost packets. The most straightforward way of doing this is for each node to use an FEC code, decoding and re-encoding packets it receives. The drawback of such an approach is delay. Every node on the path codes and decodes packets in a block. A way of overcoming this drawback is to use codes that operate in more of a "convolutional" manner, sending out coded packets formed from packets received thus far, without decoding. The random linear coding scheme of Section 2.1 is such a code. A variation, with lower complexity, is described in [85].

5. **Full coding:** In this case, paths are eschewed altogether, and we use our solution to the efficient operation problem. Problem (3.2) is solved to find a subgraph, and the random linear coding scheme of Section 2.1 is used. This represents the limit of achievability provided that we are restricted from modifying the design of the physical layer and that we do not exploit the timing of packets to convey information.

4.1.1 Simulation set-up

Nodes were placed randomly according to a uniform distribution over a square region. The size of the square was set to achieve unit node density. We considered a network where transmissions were subject to distance attenuation and Rayleigh fading, but not interference (owing to scheduling). So, when node i transmits, the signal-to-noise ratio (SNR) of the signal received at node j is $\gamma d(i,j)^{-\alpha}$, where γ is an exponentially-distributed random variable with unit mean, $d(i,j)$ is the distance between node i

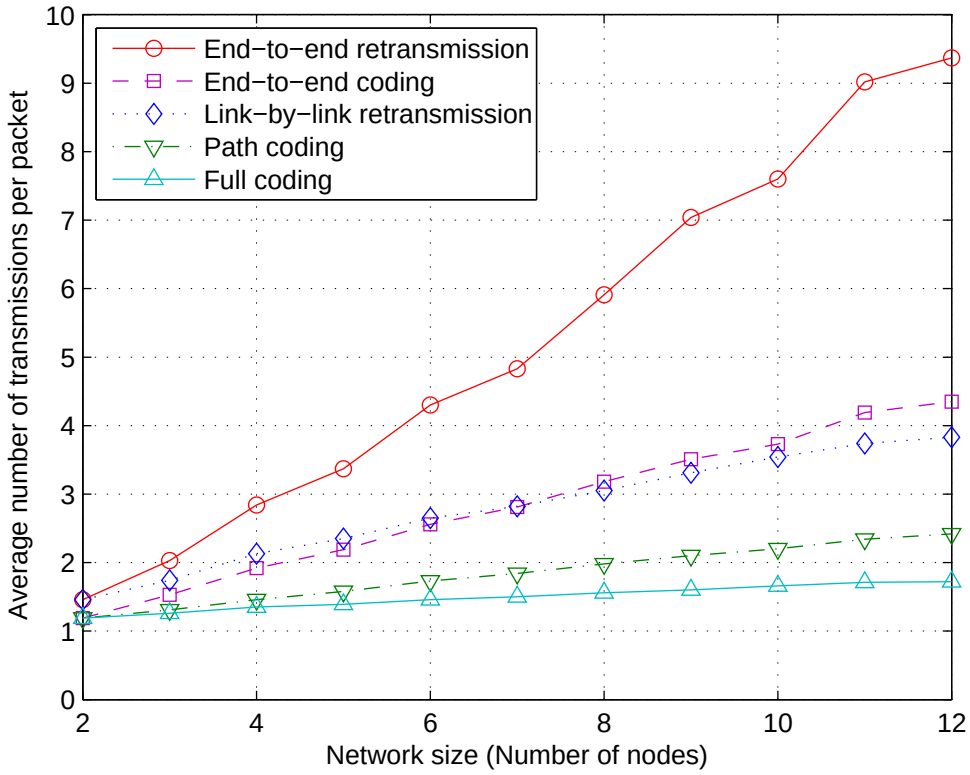

Figure 4.1: Average number of transmissions per packet as a function of network size for various wireless unicast approaches.

and node j, and α is an attenuation parameter that we took to be 2. We assumed that a packet transmitted by node i is successfully received by node j if the received SNR exceeds β, i.e., $\gamma d(i,j)^{-\alpha} \geq \beta$, where β is a threshold that we took to be 1/4. If a packet is not successfully received, then it is completely lost. If acknowledgments are sent, acknowledgments are subject to loss in the same way that packets are and follow the reverse path.

4.1.2 Simulation results

The average number of transmissions required per packet using the various approaches in random networks of varying size is shown in Figure 4.1. Paths or subgraphs

were chosen in each random instance to minimize the total number of transmissions required, except in the cases of end-to-end retransmission and end-to-end coding, where they were chosen to minimize the number of transmissions required by the source node (the optimization to minimize the total number of transmissions in these cases cannot be done straightforwardly by a shortest path algorithm). We see that, while end-to-end coding and link-by-link retransmission already represent significant improvements on end-to-end retransmission, the coded approaches represent more significant improvements still. By a network size of nine nodes, full coding already improves on link-by-link retransmission by a factor of two. Moreover, as the network size grows, the performance of the various schemes diverges.

Here, we discuss performance simply in terms of the number of transmissions required per packet; in some cases, e.g., congestion, the performance measure increases super-linearly in this quantity, and the performance improvement is even greater than that depicted in Figure 4.1. We see, at any rate, that our prescription for efficient operation promises significant improvements, particularly for large networks.

4.2 Minimum-weight wireline multicast

A common networking problem is that of minimizing the weight of a multicast connection in a lossless wireline network, where the weight of the connection is determined by weights, or artificial costs, placed on links to direct the flow of traffic. Since we consider a wireline network, the links are all point-to-point and all hyperarcs are simple arcs. The cost function is linear and separable, namely, it is $f(z) = \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij}$, where a_{ij} is the weight of the link represented by arc (i, j). The constraint set Z is the entire positive orthant, since it is generally assumed that the rate of the connection is much smaller than the capacity of the network.

For routed networks, the standard approach to establishing minimum-weight multicast connections is to find the shortest tree rooted at the source that reaches all the sinks, which equates to solving the Steiner tree problem on directed graphs [89]. For

coded networks, we see that optimization problem (3.3) is, in this case, a linear optimization problem and, as such, admits a polynomial-time solution. By contrast, the Steiner tree problem on directed graphs is well-known to be NP-complete. Although tractable approximation algorithms exist for the Steiner tree problem on directed graphs (e.g., [22, 89, 117]), the solutions thus obtained are suboptimal relative to minimum-weight multicast without coding, which in turn is suboptimal relative to when coding is used, since coding subsumes forwarding and replicating. Thus, coding promises potentially significant weight improvements.

4.2.1 Simulation set-up

We conducted simulations where we took graphs representing various internet service provider (ISP) networks and assessed the average total weight of random multicast connections using, first, our network-coding based solution to the efficient operation problem and, second, routing over the tree given by the directed Steiner tree (DST) approximation algorithm described in [22]. The graphs, and their associated link weights, were obtained from the Rocketfuel project of the University of Washington [80]. The approximation algorithm in [22] was chosen for comparison as it achieves a poly-logarithmic approximation ratio (it achieves an approximation ratio of $O(\log^2 |T|)$, where $|T|$ is the number of sink nodes), which is roughly as good as can be expected from any practical algorithm, since it has been shown that it is highly unlikely that there exists a polynomial-time algorithm that can achieve an approximation factor smaller than logarithmic [89].

4.2.2 Simulation results

The results of the simulations are tabulated in Table 4.1. We see that, depending on the network and the size of the multicast group, the average weight reduction ranges from 10% to 33%. Though these reductions are modest, it is important to keep in mind that our solution easily accommodates distributed operation and, by

Network	Approach	Average multicast weight			
		2 sinks	4 sinks	8 sinks	16 sinks
Telstra (au)	DST approximation	17.0	28.9	41.7	62.8
	Network coding	13.5	21.5	32.8	48.0
Sprint (us)	DST approximation	30.2	46.5	71.6	127.4
	Network coding	22.3	35.5	56.4	103.6
Ebone (eu)	DST approximation	28.2	43.0	69.7	115.3
	Network coding	20.7	32.4	50.4	77.8
Tiscali (eu)	DST approximation	32.6	49.9	78.4	121.7
	Network coding	24.5	37.7	57.7	81.7
Exodus (us)	DST approximation	43.8	62.7	91.2	116.0
	Network coding	33.4	49.1	68.0	92.9
Abovenet (us)	DST approximation	27.2	42.8	67.3	75.0
	Network coding	21.8	33.8	60.0	67.3

Table 4.1: Average weights of random multicast connections of unit rate and varying size for various approaches in graphs representing various ISP networks.

contrast, computing Steiner trees is generally done at a single point with full network knowledge.

4.3 Minimum-energy wireless multicast

Another problem of interest is that of minimum-energy multicast (see, e.g., [68, 107]). In this problem, we wish to achieve minimum-energy multicast in a lossless wireless network without explicit regard for throughput or bandwidth, so the constraint set Z is again the entire positive orthant. The cost function is linear and separable, namely, it is $f(z) = \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ}$, where a_{iJ} represents the energy required to transmit a packet to nodes in J from node i. Hence problem (3.3) becomes a linear optimization problem with a polynomial number of constraints, which can therefore be solved in polynomial time. By contrast, the same problem using traditional routing-based approaches is NP-complete—in fact, the special case of broadcast in itself is NP-complete, a result shown in [68, 3]. The problem must therefore be addressed using polynomial-time heuristics such as the Multicast Incremental Power (MIP) algorithm

Network size	Approach	Average multicast energy			
		2 sinks	4 sinks	8 sinks	16 sinks
20 nodes	MIP algorithm	30.6	33.8	41.6	47.4
	Network coding	15.5	23.3	29.9	38.1
30 nodes	MIP algorithm	26.8	31.9	37.7	43.3
	Network coding	15.4	21.7	28.3	37.8
40 nodes	MIP algorithm	24.4	29.3	35.1	42.3
	Network coding	14.5	20.6	25.6	30.5
50 nodes	MIP algorithm	22.6	27.3	32.8	37.3
	Network coding	12.8	17.7	25.3	30.3

Table 4.2: Average energies of random multicast connections of unit rate and varying size for various approaches in random wireless networks of varying size.

proposed in [107].

4.3.1 Simulation set-up

We conducted simulations where we placed nodes randomly, according to a uniform distribution, in a 10×10 square with a radius of connectivity of 3 and assessed the average total energy of random multicast connections using first, our network-coding based solution to the efficient operation problem and, second, the routing solution given by the MIP algorithm. The energy required to transmit at unit rate to a distance d was taken to be d^2.

4.3.2 Simulation results

The results of the simulations are tabulated in Table 4.2. We see that, depending on the size of the network and the size of the multicast group, the average energy reduction ranges from 13% to 49%. These reductions are more substantial than those reported in Section 4.2.2, but are still somewhat modest. Again, it is important to keep in mind that our solution easily accommodates distributed operation.

In Table 4.3, we tabulate the behavior of a distributed approach, specifically, an approach using the subgradient method (applied to problem (3.26)). The algorithm

Network size	Number of sinks	Average multicast energy				
		Optimal	25 iterations	50 iterations	75 iterations	100 iterations
30 nodes	2	16.2	16.7	16.3	16.3	16.2
	4	21.8	24.0	22.7	22.3	22.1
	8	27.8	31.9	29.9	29.2	28.8
40 nodes	2	14.4	15.0	14.5	14.5	14.4
	4	18.9	21.8	21.2	19.6	19.4
	8	25.6	31.5	29.2	28.0	27.4
50 nodes	2	12.4	13.1	12.6	12.5	12.5
	4	17.4	20.7	18.9	18.2	18.0
	8	22.4	29.0	26.8	25.5	24.8

Table 4.3: Average energies of random multicast connections of unit rate and varying size for the subgradient method in random wireless networks of varying size. The optimal energy was obtained using a linear program solver.

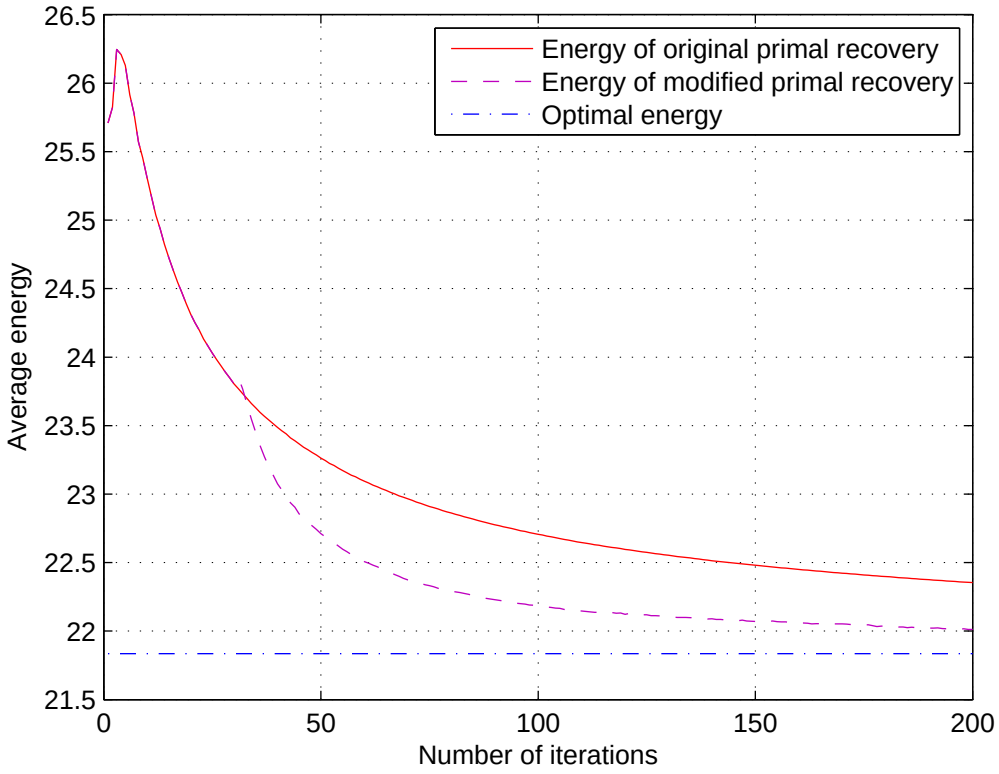

Figure 4.2: Average energy as a function of the number of iterations for the subgradient method on random 4-sink multicast connections of unit rate in random 30-node wireless networks.

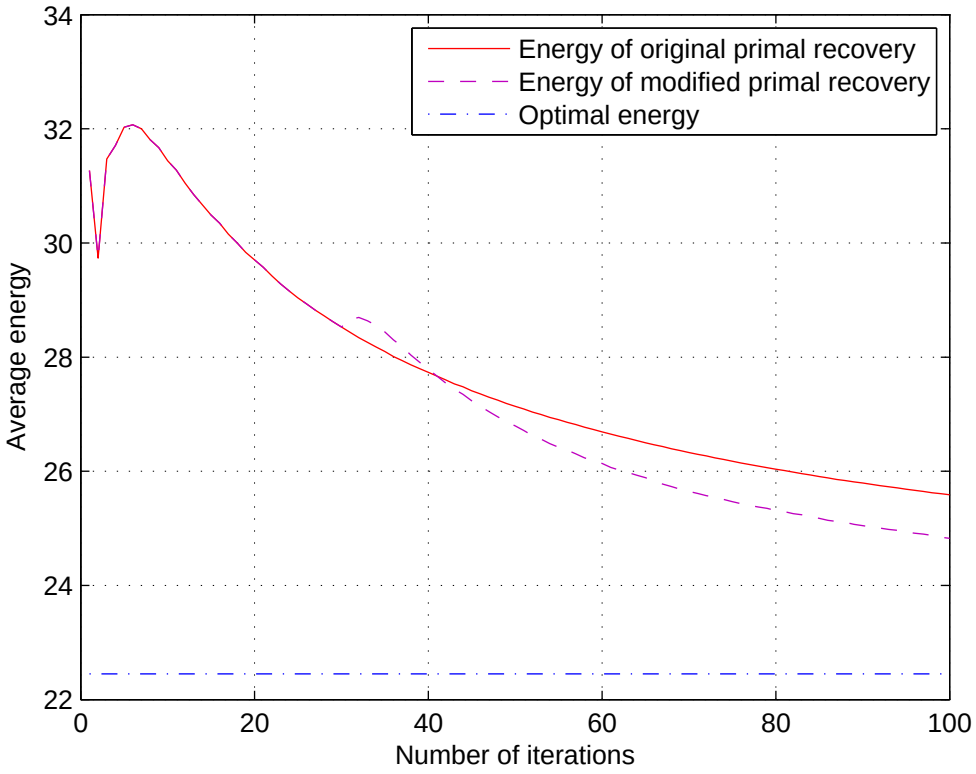

Figure 4.3: Average energy as a function of the number of iterations for the subgradient method on random 8-sink multicast connections of unit rate in random 50-node wireless networks.

was run under step sizes given by $\theta[n] = n^{-0.8}$ and convex combination weights by $\mu_l[n] = 1/n$, if $n < 30$, and $\mu_l[n] = 1/30$, if $n \geq 30$. We refer to this choice of parameters as the case of "modified primal recovery". Note that, despite our aiming to run sufficiently many trials to ascertain the true average with high probability, the simulations reported in Table 4.3 do not agree exactly with those in Table 4.2 because they were run on different sets of random instances.

Our first choice of parameters was step sizes given by $\theta[n] = n^{-0.8}$ and convex combination weights by $\mu_l[n] = 1/n$. This case, which we refer to as "original primal recovery", was found to suffer adversely from the effect of poor primal solutions obtained in early iterations. In Figures 4.2 and 4.3, we show the behavior of the subgradient method in the cases of a 4-sink multicast in a 30-node network and an 8-sink multicast in a 50-node network, respectively, in detail. In these figures, we show both parameter choices, and we see that modified primal recovery performs substantially better. For reference, the optimal energy of the problem is also shown.

We see that the subgradient method yields solutions that converge rapidly to an optimal one, and it appears to be a promising candidate for the basis of a protocol.

Chapter 5

Conclusion

ROUTING is undoubtedly a satisfactory way to operate packet networks. It clearly works. What is not clear is whether it should be used for all types of networks. As we mentioned, coding is a definite alternative at least for application-layer overlay networks and multi-hop wireless networks. To actually use coding, however, we must apply to coding the same considerations that we apply to routing. This thesis was motivated by exactly that. We took the basic premise of coding and addressed a fundamental problem in packet networks—efficient operation. We laid out a solution to the efficient operation problem, defined as it was to factor in packet loss, packet broadcast, and asynchronism in packet arrivals. That, we believe, is our main contribution.

From here, there is promising work both in expanding the scope of the problem and in examining the problem more deeply. We discuss first the former. One way of expanding the scope of the problem is by including more considerations from networking. In particular, an important issue outside the present scope is flow, or congestion, control. We have taken, as a starting point, messages admitted into the network at given rates and left aside the problem of determining which messages to admit and at what rates. This problem can be dealt with separately, e.g., using window flow control as in TCP, but it need not be. In routed packet networks, flow control can

be done jointly with optimal routing (see [13, Section 6.5.1]), and it may likewise be possible to extend the subgraph selection techniques that we proposed so that they jointly perform subgraph selection and flow control. Indeed, an extension of the primal-dual method of Section 3.2.1 to perform joint subgraph selection and flow control is given in [79, Section II-C]. Even if flow control is done separately, there has not, to our knowledge, been an earnest study of the flow control problem for coded packet networks.

As for examining the efficient operation problem more deeply, there are fundamental open questions relating to both network coding and subgraph selection. Let us first discuss network coding. As we mentioned in Section 2.4, the random linear coding scheme that we proposed as a solution to the network coding problem is good in that it maximizes throughput. But throughput may not be our principal concern. Other performance metrics that may be important are memory usage, computational load, and delay. Moreover, feedback may be present. Our true desire, then, is to optimize over a five-dimensional space whose five axes are throughput, memory usage, computational load, delay, and feedback usage.

Some points in this five-dimensional space are known. We know, e.g., that random linear coding achieves maximum throughput; we can calculate or estimate its memory usage, computational load, and delay; and we know that its feedback usage is minimal or non-existent. For networks consisting only of point-to-point links, we have two other useful points. We know that, by using a retransmission scheme on each link (i.e., acknowledging the reception of packets on every link and retransmitting unacknowledged packets), we achieve maximum throughput and minimum memory usage, computational load, and delay at the cost of high feedback usage (we require a reliable feedback message for every received packet). We know also that, by using a low-complexity erasure code on each link (e.g., a Raptor code [81, 96] or an LT code [69]), we trade-off, with respect to random linear coding, computational load for delay. The challenge is to fill out this space more. In the context of channel

coding, such a challenge might seem absurd—an overly ambitious proposition. But, as the slotted Aloha relay channel (see Section 1.2.1) illustrates, network coding is different from channel coding, and problems intractable for the latter may not be for the former. A preliminary attempt at tackling this problem is made in [85].

Let us discuss, now, open questions relating to subgraph selection. In this thesis, we gave distributed algorithms that apply only if the constraints caused by medium access issues can essentially be disregarded. But these issues are important and often must be dealt with, and it remains to develop distributed algorithms that incorporate such issues explicitly. A good starting point would be to develop distributed algorithms for slotted Aloha networks of the type described in Section 1.2.1.

Much potential for investigation is also present in the cases for which our algorithms do apply. Other distributed algorithms are given in [28, 108, 112, 113], and no doubt more still can be developed. For example, our choice to approximate the maximum function with an l^m-norm in Section 3.2 is quite arbitrary, and it seems likely that there are other approximations that yield good, and possibly even better, distributed algorithms.

No matter how good the distributed algorithm, however, there will be some overhead in terms of information exchange and computation. What we would like ideally is to perform the optimization instantly without any overhead. That goal is impossible, but, failing that, we could content ourselves with optimization methods that have low overhead and fall short of the optimal cost. From such a suboptimal solution, we could then run a distributed algorithm to bring us to an optimal solution or, simply, use the suboptimal solution. A suboptimal, but simple, subgraph selection method for minimum-energy broadcast in coded wireless networks is given in [106]. Little else has been done. It might seem contradictory that we started this thesis by lamenting the use of ad hoc methods and heuristics, yet we now gladly contemplate their use. There is a difference, however, between proposing the use of ad hoc methods when the optimum is unknown or poorly defined and doing so when the optimum is known

but simply cannot be achieved practically. What we now call for is the latter.

Another point about the algorithms we have proposed is that they optimize based on rates—rates of the desired connections and rates of packet injections. But we do not necessarily need to optimize based on rates, and there is a body of work in networking theory where subgraph selection is done using queue lengths rather than rates [7, 102]. This work generally relates to routed networks, and the first that applies to coded networks is [51]. Adding such queue-length based optimization methods to our space of mechanisms for subgraph selection may prove useful in our search for practical methods. What we would like to know, ideally, is the most practical method for network x, given its particular capabilities and constraints. This might or might not be one of the methods proposed in this thesis; determining whether it is, and what is if it is not, is the challenge.

This drive toward practicality fits with the principal motivation of this thesis: we saw coding as a promising practical technique for packet networks, so we studied it. And we believe, on the basis of our results, that our initial hypothesis has been confirmed. Realizing coded packet networks, therefore, is a worthwhile goal, and we see our work as an integral step toward this goal. But that is not our only goal: Gallager's comment on the "art" of networking (see Chapter 1) is, we believe, indicative of a general consensus that current understanding of data networks is poor, at least in relation to current understanding of other engineered systems, such as communication channels. There is no clear reason why this disparity of understanding must exist, and the advances of networking theory have done much to reduce its extent. The study of coded networks may reduce the disparity further—as we have seen in this thesis, we are, in the context of coded packet networks, able to find optimal solutions to previously-intractable problems. This goal, of increasing our general understanding, is one of the goals of this thesis, and we hope to spawn more work toward this goal. Perhaps coding may be the ingredient necessary to finally put our understanding of data networks on par with our understanding of communication

channels.

Bibliography

[1] S. Acedański, S. Deb, M. Médard, and R. Koetter, "How good is random linear coding based distributed networked storage?" in *Proc. WINMEE, RAWNET and NETCOD 2005 Workshops*, Apr. 2005.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[3] A. Ahluwalia, E. Modiano, and L. Shu, "On the complexity and distributed construction of energy-efficient broadcast trees in static ad hoc wireless networks," in *Proc. 2002 Conference on Information Sciences and Systems (CISS 2002)*, Mar. 2002.

[4] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice Hall, 1993.

[5] M. Arai, S. Fukumoto, and K. Iwasaki, "Reliability analysis of a convolutional-code-based packet level FEC under limited buffer size," *IEICE Trans. Fundamentals*, vol. E88-A, no. 4, pp. 1047–1054, Apr. 2005.

[6] M. Arai, A. Yamaguchi, and K. Iwasaki, "Method to recover internet packet losses using $(n, n-1, m)$ convolutional codes," in *Proc. International Conference on Dependable Systems and Networks (DSN 2000)*, June 2000, pp. 382–389.

[7] B. Awerbuch and T. Leighton, "A simple local-control approximation algorithm for multicommodity flow," in *Proc. 34th Annual IEEE Symposium on Foundations of Computer Science*, Nov. 1993, pp. 459–468.

[8] D. P. Bertsekas, "A class of optimal routing algorithms for communication networks," in *Proc. 5th International Conference on Computer Communication (ICCC '80)*, Oct. 1980, pp. 71–76.

[9] ——, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[10] ——, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.

[11] ——, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2001, vol. 2.

[12] D. P. Bertsekas, E. M. Gafni, and R. G. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 911–919, Aug. 1984.

[13] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1992.

[14] D. Bertsimas, I. C. Paschalidis, and J. Tsitsiklis, "On the large deviations behavior of acyclic networks of $G/G/1$ queues," *Ann. Appl. Probab.*, vol. 8, no. 4, pp. 1027–1069, Nov. 1998.

[15] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Apr. 2006.

[16] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 343–351, Mar. 1983.

[17] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proc. WINMEE, RAWNET and NETCOD 2005 Workshops*, Apr. 2005.

[18] K. Bhattad, N. Ratnakar, R. Koetter, and K. R. Narayanan, "Minimal network coding for multicast," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1730–1734.

[19] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE J. Select. Areas Commun.*, vol. 20, no. 8, pp. 1528–1540, Oct. 2002.

[20] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. 2002 IEEE International Symposium on Information Theory (ISIT 2002)*, June/July 2002, p. 323.

[21] V. W. S. Chan, K. L. Hall, E. Modiano, and K. A. Rauschenbach, "Architectures and technologies for high-speed optical data networks," *J. Lightwave Technol.*, vol. 16, no. 12, pp. 2146–2168, Dec. 1998.

[22] M. Charikar, C. Chekuri, T.-y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," *J. Algorithms*, vol. 33, no. 1, pp. 73–91, Oct. 1999.

[23] H. Chen and A. Mandelbaum, "Discrete flow networks: Bottleneck analysis and fluid approximations," *Math. Oper. Res*, vol. 16, no. 2, pp. 408–446, May 1991.

[24] H. Chen and D. D. Yao, *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, ser. Applications of Mathematics. New York, NY: Springer, 2001, vol. 46.

[25] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York, NY: John Wiley & Sons, 1991.

[27] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proc. IEEE Infocom 2003*, vol. 1, Mar./Apr. 2003, pp. 702–711.

[28] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal distributed multicast routing using network coding: Theory and applications," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 2, pp. 47–49, 2004.

[29] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *Proc. International Workshop on Wireless Ad-hoc Networks (IWWAN) 2005*, May 2005, invited paper.

[30] S. Deb and M. Médard, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," submitted to *IEEE Trans. Inform. Theory.*

[31] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 274–285, Apr. 2004.

[32] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[33] P. Elias, "Coding for two noisy channels," in *Proc. Third London Symposium on Information Theory.* Academic Press, 1956, pp. 61–74.

[34] E. Erez and M. Feder, "Convolutional network codes," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, June/July 2004, p. 146.

[35] ——, "Efficient network codes for cyclic networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1982–1986.

[36] J. Feldman, T. Malkin, R. A. Servedio, and C. Stein, "On the capacity of secure network coding," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept./Oct. 2004.

[37] W. Feller, *An Introduction to Probability Theory and its Applications*, 3rd ed. New York, NY: John Wiley & Sons, 1968, vol. 1.

[38] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005, invited paper.

[39] C. Fragouli and E. Soljanin, "A connection between network coding and convolutional codes," in *Proc. 2004 IEEE International Conference on Communications (ICC 2004)*, vol. 2, June 2004, pp. 661–666.

[40] ——, "Decentralized network coding," in *Proc. 2004 IEEE Information Theory Workshop (ITW 2004)*, Oct. 2004, pp. 310–314.

[41] R. G. Gallager, "Claude E. Shannon: A retrospective on his life, work, and impact," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2681–2695, Nov. 2001.

[42] S. Ghez, S. Verdú, and S. C. Schwartz, "Stability properties of slotted Aloha with multipacket reception capability," *IEEE Trans. Automat. Contr.*, vol. 33, no. 7, pp. 640–649, July 1988.

[43] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE Infocom 2005*, vol. 4, Mar. 2005, pp. 2235–2245.

[44] R. Gowaikar, A. F. Dana, R. Palanki, B. Hassibi, and M. Effros, "On the capacity of wireless erasure networks," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, June/July 2004, p. 401.

[45] T. S. Han, "Slepian-Wolf-Cover theorem for networks of channels," *Inf. Control*, vol. 47, pp. 67–83, 1980.

[46] N. J. A. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, Jan. 2005, pp. 489–498.

[47] T. Ho, B. Leong, Y.-H. Chang, Y. Wen, and R. Koetter, "Network monitoring in multicast networks using network coding," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1977–1981.

[48] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, June/July 2004, p. 144.

[49] T. Ho, M. Médard, and R. Koetter, "An information-theoretic view of network management," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1295–1312, Apr. 2005.

[50] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://www.its.caltech.edu/~tho/itrandom-revision.pdf

[51] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

[52] M. Imase and B. M. Waxman, "Dynamic Steiner tree problem," *SIAM J. Disc. Math.*, vol. 4, no. 3, pp. 369–384, Aug. 1991.

[53] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1455–1459.

[54] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[55] K. Jain, L. Lovász, and P. A. Chou, "Building scalable and robust peer-to-peer overlay networks for broadcasting using network coding," in *PODC '05: Proc. 24th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 2005, pp. 51–59.

[56] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom '03: Proc. 9th Annual International Conference on Mobile Computing and Networking*, 2003, pp. 66–80.

[57] M. Johansson, L. Xiao, and S. Boyd, "Simultaneous routing and power allocation in CDMA wireless data networks," in *Proc. 2003 IEEE International Conference on Communications (ICC 2003)*, vol. 1, May 2003, pp. 51–55.

[58] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard, "The importance of being opportunistic: Practical network coding for wireless environments," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

[59] F. P. Kelly, *Reversibility and Stochastic Networks*. Chichester: John Wiley & Sons, 1979.

[60] R. Khalili and K. Salamatian, "On the capacity of erasure relay channel: Multi-relay case," in *Proc. 2005 IEEE Information Theory Workshop (ITW 2005)*, Aug. 2005.

[61] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels," *IEEE/ACM Trans. Networking*, vol. 13, no. 4, pp. 868–880, Aug. 2005.

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

[63] A. H. Lee and M. Médard, "Simplified random network codes for multicast networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1725–1729.

[64] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[65] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept./Oct. 2004.

[66] ——, "Efficient and distributed computation of maximum multicast rates," in *Proc. IEEE Infocom 2005*, vol. 3, Mar. 2005, pp. 1618–1628.

[67] Z. Li, B. Li, D. Jiang, and L. C. Lau, "On achieving optimal throughput with network coding," in *Proc. IEEE Infocom 2005*, vol. 3, Mar. 2005, pp. 2184–2194.

[68] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," in *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '02)*, June 2002, pp. 112–122.

[69] M. Luby, "LT codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, Nov. 2002, pp. 271–280.

[70] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept./Oct. 2004, invited paper.

[71] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, Oct. 2004, pp. 1232–1237.

[72] D. S. Lun, M. Médard, and D. R. Karger, "On the dynamic multicast problem for coded networks," in *Proc. WINMEE, RAWNET and NETCOD 2005 Workshops*, Apr. 2005.

[73] D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding," in *Proc. International Workshop on Convergent Technologies (IWCT) 2005*, June 2005, invited paper.

[74] ——, "Network coding for efficient wireless unicast," in *Proc. 2006 International Zurich Seminar on Communications (IZS 2006)*, Feb. 2006, pp. 74–77, invited paper.

[75] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/cs.IT/0510070

[76] ——, "Further results on coding for reliable communication over packet networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1848–1852.

[77] D. S. Lun, P. Pakzad, C. Fragouli, M. Médard, and R. Koetter, "An analysis of finite-memory random linear coding on packet streams," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Apr. 2006.

[78] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom 2005*, vol. 3, Mar. 2005, pp. 1608–1617.

[79] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," to appear in *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/cs.IT/0503064

[80] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurement (IMW 2002)*, Nov. 2002, pp. 231–236.

[81] P. Maymounkov, "Online codes," NYU, Technical Report TR2002-833, Nov. 2002.

[82] M. Médard, M. Effros, D. Karger, and T. Ho, "On coding for non-multicast networks," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[83] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY: John Wiley & Sons, 1999.

[84] A. Ouorou, P. Mahey, and J.-P. Vial, "A survey of algorithms for convex multicommodity flow problems," *Manage. Sci.*, vol. 46, no. 1, pp. 126–147, Jan. 2000.

[85] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1853–1857.

[86] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching: Switch fabrics, techniques, and architectures," *J. Lightwave Technol.*, vol. 21, no. 2, pp. 384–405, Feb. 2003.

[87] I. C. Paschalidis and Y. Liu, "Large deviations-based asymptotics for inventory control in supply chains," *Oper. Res.*, vol. 51, no. 3, pp. 437–460, May–June 2003.

[88] A. Ramamoorthy, J. Shi, and R. D. Wesel, "On the capacity of network coding for random networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2878–2885, Aug. 2005.

[89] S. Ramanathan, "Multicast tree generation in networks with asymmetric links," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 558–568, Aug. 1996.

[90] A. Rasala Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[91] N. Ratnakar and G. Kramer, "The multicast capacity of acyclic, deterministic relay networks with no interference," to appear in *IEEE Trans. Inform. Theory.*

[92] N. Ratnakar, D. Traskov, and R. Koetter, "Approaches to network coding for multiple unicasts," in *Proc. 2006 International Zurich Seminar on Communications (IZS 2006)*, Feb. 2006, pp. 70–73, invited paper.

[93] S. Riis, "Linear versus non-linear Boolean functions in network flow," in *Proc. 2004 Conference on Information Sciences and Systems (CISS 2004)*, Mar. 2004.

[94] Y. E. Sagduyu and A. Ephremides, "Crosslayer design and distributed MAC and network coding in wireless ad hoc networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1863–1867.

[95] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programs," *Oper. Res. Lett.*, vol. 19, pp. 105–113, 1996.

[96] A. Shokrollahi, "Raptor codes," Jan. 2004, preprint. [Online]. Available: http://algo.epfl.ch/contents/output/pubs/raptor.pdf

[97] N. Shulman, "Communication over an unknown channel via common broadcasting," Ph.D. dissertation, Tel Aviv University, July 2003.

[98] N. Shulman and M. Feder, "Static broadcasting," in *Proc. 2000 IEEE International Symposium on Information Theory (ISIT 2000)*, June 2000, p. 23.

[99] L. Song, R. W. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[100] R. Srikant, *The Mathematics of Internet Congestion Control*. Boston, MA: Brikhäuser, 2004.

[101] J. Tan and M. Médard, "Secure network coding with a cost criterion," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Apr. 2006.

[102] L. Tassiulas and A. F. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[103] J. N. Tsitsiklis, "Decentralized detection," in *Advances in Statistical Signal Processing*. Greenwich, CT: JAI Press, 1993, vol. 2, pp. 297–344.

[104] E. C. van der Meulen, "Three-terminal communication channels," *Adv. Appl. Probab.*, vol. 3, pp. 120–154, 1971.

[105] B. M. Waxman, "Routing of multicast connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[106] J. Widmer, C. Fragouli, and J.-Y. Le Boudec, "Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding," in *Proc. WINMEE, RAWNET and NETCOD 2005 Workshops*, Apr. 2005.

[107] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 481–492, 2002.

[108] Y. Wu, M. Chiang, and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A critical cut approach," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Apr. 2006.

[109] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, June/July 2004, p. 143.

[110] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proc. 2005 Conference on Information Sciences and Systems (CISS 2005)*, Mar. 2005.

[111] ——, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906–1918, Nov. 2005.

[112] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

[113] ——, "Distributed algorithms for minimum cost multicast with network coding in wireless networks," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Apr. 2006.

[114] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136–1144, July 2004.

[115] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 412–422, Mar. 1995.

[116] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 1, pp. 107–120, Jan. 2004.

[117] L. Zosin and S. Khuller, "On directed Steiner trees," in *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, Jan. 2002, pp. 59–63.