Benefits of Opportunistic Routing, Implicit Acknowledgments, and Network Coding on a Linear Broadcast Network

by

Daniel Montgomery Whisman

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

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author . . . .
Department of Electrical Engineering and Computer Science
May 20, 2016

Certified by . . . .
Muriel Médard
Cecil H. Green Professor of EECS
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor
Chair, EECS Committee on Graduate Students

Benefits of Opportunistic Routing, Implicit Acknowledgments, and Network Coding on a Linear Broadcast Network

by

Daniel Montgomery Whisman

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis studies the benefits of using opportunistic routing, implicit acknowledgments, and network coding on a linear broadcast packet network. Nodes are arranged in a line, and the first node wishes to communicate with the end node. When node i transmits, it is received at node j with a probability $P_{i,j}$. Several communication protocols are proposed and their performance studied using the mean and variance of the completion time as metrics. The protocols studied use end-to-end retransmission, end-to-end coding, and link-by-link retransmission with network coding both with and without opportunistic routing. Simulation and analytical results are presented. End-to-end coding significantly outperforms end-to-end retransmission on both metrics, and the link-by-link protocols outperform both. Opportunistic routing shows a mixed benefit over link-by-link protocols without it. When using opportunistic routing, the variance of the completion time is higher, and the mean is either similar or lower, depending on the channel conditions. When the loss probabilities are higher, opportunistic routing shows little benefit, whereas with a lower probability of packet loss, opportunistic routing shows a significant reduction in mean completion time.

Thesis Supervisor: Muriel Médard
Title: Cecil H. Green Professor of EECS

Acknowledgments

I would like to thank my advisor, Professor Muriel Médard, for her patience and wisdom while writing my thesis. I would also like to thank Drs. Julius Kusuma and Arnaud Croux for the motivation for this work and their help with the many technical details. Finally, I would like to thank all my colleagues in the group, past and present, for their support, knowledge, insights, and perspectives during my two years at M.I.T.: Arman Rezaee, Salman Salamatian, Soheil Feizi, Jason Cloud, Flavio du Pin Calmon, Weifei Zeng, Vitaly Abdrashitov, Georgios Angelopoulos, and Ahmad Beirami.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Setup

Consider a network of N nodes arranged in a line (see figure 1-1). Node 1 wishes to communicate with node N, and does so using some, possibly all, of the intermediate nodes as relays. The nodes share a single broadcast channel with a fixed bandwidth W. Nodes transmit packets of fixed duration, and a packet transmitted from node i is received by node j with probability $P_{i,j}$. Since the channel is broadcast, there are multiple nodes j such that $P_{i,j} > 0$, with the possible exception of the end nodes. Any transmissions that overlap interfere and can not be received. It is assumed that a transmission cannot be heard more than δ nodes away. That is, for some node i in the network, the probability of successful transmission from i to $i + \delta + 1$ is zero.

For example, in figure 1-1, when node 4 transmits, nodes 3 and 5 overhear with 95% probability, nodes 2 and 6, with 75% probability, and nodes 1 and 7, with 45% probability. It is important to note that, although a node may not be able to receive a transmission successfully, that transmission interferes with others on the channel. For example, if node 4 transmits, even if node 7 cannot successfully receive the transmission from 4, it is not able to successfully receive a transmission from node 8.

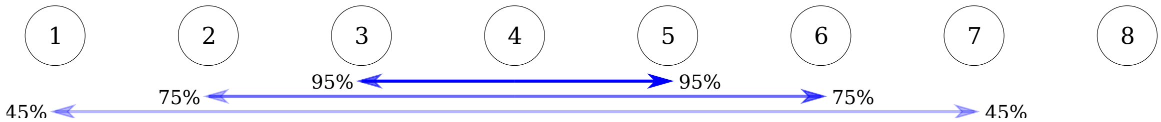

Figure 1-1: Linear mesh network with $N = 8$ nodes. The blue arrows represent a transmission from node 4. The transmission reaches nodes 3 and 5 with a probability of 95%, nodes 2 and 6 with a probability of 75%, nodes 1 and 7 with a probability of 45%, and node 8 with a probability of 0.

1.2 Prior Work

1.2.1 Improved Routing

Siqueira et al. [22] demonstrates a throughput advantage of up to 22% when using LIBR (Linear ID-Based Routing), a routing protocol tailored to linear mesh networks, over existing ad-hoc routing protocols such as OLSR (Optimized Link State Routing), AODV (Ad hoc On-Demand Distance Vector Routing), and DSDV (Destination-Sequenced Distance-Vector Routing). LIBR makes use of the known structure of the network to reduce the number of control packets needed.

1.2.2 Network Coding

Network coding was introduced by Ahlswede, et al. in [1]. The authors characterized the admissible coding rate region for one information source by treating packets as information which can be combined or coded rather than physical entities which can only be routed or replicated. Network coding, as used in this thesis, is a way to introduce redundancy into transmissions so the receiver can recover from packet loss. Instead of transmitting raw data packets, nodes transmit linear combinations of packets in their memory, referred to as *degrees of freedom* (dofs). Each packet can be thought of as a vector in $\mathbb{F}_q$, where q is the size of the finite field. A single dof, $\vec{m}'$, is given by

$$\vec{m}' = \sum_{m \in \mathcal{M}} \alpha_m \vec{m}$$

where $\mathcal{M}$ is the set of all packets in the node's memory, and the coefficients $\alpha_m \in \mathbb{F}_q$ are chosen at random. Since $\vec{m}'$ is a linear combination of all $m \in \mathcal{M}$ and the coefficients are chosen at random, this type of network coding is referred to as random linear network coding, or RLNC.

Each dof must contain the coefficients of the original data packets that make it up. If the dofs are composed of M data packets, then the receiver can decode after it receives M dofs. In contrast to end-to-end coding, RLNC allows intermediate nodes to add redundancy *without having to decode*. For example (referring to figure 1-1), if the loss probability on each hop is 5%, then the probability of a packet loss along the entire network is about 30%. When end-to-end coding is used, the sender needs to transmit an extra 8 or 9 (coded) packets on average out of every 20 to compensate for the packet loss. When RLNC is used, each node only needs to transmit one extra dof out of 20, reducing the number of transmissions needed especially at nodes further from the destination.

Network coding also reduces the complexity of feedback for reliable transmission schemes. Instead of tracking which packets have been received, the sender only needs to know whether enough dofs have been received.

The benefit of using network coding on a lossy packet network was studied in [7], [16], and [17]. In [7], the problem of transmitting a large file to multiple receivers on a wireless channel was considered. For broadcast of a single flow to multiple receivers, random coding reduces the mean completion time by a factor of about three when the packet loss probability is 0.5. In [16], the authors presented a coding scheme similar to the network coding scheme used in this thesis, which is as follows. The source transmits a block of packets, one at a time. When a node receives a packet, it is stored in the node's memory. When a node has an opportunity to transmit, it sends a random linear combination of all packets in its memory. Decoding is performed using Gaussian elimination. It is important to note that, in the scheme presented in [16] as well as that proposed in this work, *intermediate nodes do not decode, nor do they wait to receive the whole block of packets.* They demonstrated that this RLNC scheme, and by extension that proposed in this work, is capacity achieving and specified the rate regions for a variety of situations. In [17], the authors compared the performance of end-to-end retransmission, end-to-end coding, link-by-link retransmission, path coding, and full coding in a wireless packet network. Using the average number of transmissions per packet as a metric, they demonstrated that network coding achieves about twice the performance of end-to-end coding and link-by-link retransmission, and about quadruple that of end-to-end retransmission. They also demonstrated that network coding is capacity achieving. That is, when the negative effects of a medium access control (MAC) layer are ignored, the throughput of network coding cannot be improved upon.

The benefits in terms of delay, especially when feedback is limited, slow, or unreliable, was shown in [9], [14], and [15]. In [9], the authors aimed to reduce the in-order delivery delay of selective repeat automatic repeat request (SR-ARQ) by adding redundant packets to the data stream. Automatic Repeat Request (ARQ) uses feedback to retransmit packets which are lost. SR-ARQ uses a sliding window to send multiple packets before receiving feedback, and retransmits only those which are lost. They provided analysis and numerical results which show a 50-75% reduction in expected in-order delivery delay, with a greater benefit for links with higher bandwidth delay products. Reference [14] showed that, when using time division duplexing, transmitting the optimal number of coded packets before stopping to listen for an acknowledgment provides performance close to that when using a full duplex channel. Reference [15] established a trade-off between the memory usage at intermediate nodes and the achievable rate.

Application of network coding to enhance the throughput of the transmission control protocol (TCP), an end-to-end retransmission scheme providing reliability and congestion control, was studied in [11] and [23], where network coding was used to recover from packet losses, which cause TCP to unnecessarily reduce the congestion window. Similar application to WiMAX (Worldwide Interoperability for Microwave Access) was studied in [24] and [25]. The trade-off between physical layer error correction coding and network layer erasure coding on fading channels was studied in [2] and [5]. Both works found that there is a significant benefit to using a combination of outer layer erasure codes and inner layer error correcting codes, but that this benefit diminishes on channels with more diversity. Reference [19] studied the trade-off between initial waiting time and probability of interruption of media streaming. Reference [8] studied different uses of feedback with network coding. Specifically, it studied using feedback to adjust the generation size to achieve a higher rate or reduce delay. It also compared the performance of several methods of achieving reliability, such as forward error correction, ARQ, and network coding, using delay, block size, amount of feedback, memory, and achievable rate as metrics.

1.2.3 Implicit Acknowledgments

Implicit acknowledgments allow the receiver to acknowledge a packet without an extra (explicit) transmission. For example, in a network with three nodes that share a broadcast channel, node 1 transmits a packet it wishes to send to node 3 via node 2. Node 2 receives it and transmits it to node 3. However, node 1 overhears this transmission, and can deduce that node 2 has successfully received the packet, without node 2 having to send an explicit acknowledgment.

Implicit acknowledgments require fewer transmissions than explicit acknowledgments. This means nodes spend more time transmitting data packets instead of acknowledgments, and there is less interference preventing other nodes from transmitting or receiving, both of which increase overall throughput.

The benefits of implicit acknowledgments were demonstrated in [21], and in [13] for a linear network. Reference [21] proposed a coordination scheme for a wireless sensor network that increases the energy efficiency of the network. Reference [13] used implicit acknowledgments to improve the performance of network coded schemes in underwater linear acoustic networks, using delay and average power consumption as metrics. It also considers the benefits on lightly loaded networks.

1.2.4 Opportunistic Routing

Opportunistic routing (OR) is a way to dynamically determine a which nodes forward a packet. On each hop, OR seeks to transmit the packet as far as possible as opposed to transmitting the packet along a fixed set of nodes. This potentially skips unnecessary hops, thus lowering the expected completion time for a packet to be sent from the source to the destination. Although this might not be considered routing since the network topology is linear, this thesis will use "opportunistic routing" to refer to the practice of selecting the forwarding node based on which nodes receive the packet.

For example, if opportunistic routing is used in figure 1-1 and node 4 transmits a packet, it will be overheard, with higher loss probability, by its neighbors further away from node 4. If node 7 receives the packet, it immediately transmits it further downstream. This also sends an implicit acknowledgment back upstream. Nodes 6, 5, and 4 will receive this acknowledgment and not transmit. If node 7 does not receive it and node 6 does, node 6 will wait for a short timeout to listen for node 7's implicit acknowledgment, and upon not hearing it, transmits the packet. Nodes 5 and 4 behave in a similar manner, but with double and triple the timeout, respectively.

Opportunistic routing was shown to reduce the total number of transmissions, and thus the average completion time, by Biswas and Morris [3] with Extremely Opportunistic Routing (ExOR). MAC-independent Opportunistic Routing (MORE) [4] built upon ExOR by using network coding to avoid the strict scheduling of transmissions. MORE increases throughput by 22% over ExOR, and by 45% when spatial reuse was possible. Koutsonikolas et al. [12] solved one of the major challenges in MORE—how many packets a node should forward—by improving the feedback mechanism to better communicate the state of each node. None of these works, however, make use of implicit acknowledgments. Furthermore, the main challenge in MORE simplifies in a linear network. In [6], Lucani, et al. showed that the node which should transmit is the one which has the greatest impact on the network, i.e., the node which can give the most other nodes information.

1.3 Model

Let $\mathcal{N}$ denote the set of nodes in the network. The nodes share a single broadcast channel with a fixed bandwidth W. Nodes transmit packets of a fixed duration, d, and a packet transmitted from node i is received by node j with probability $P_{i,j}$. Since the channel is broadcast, there are multiple nodes j such that $P_{i,j} > 0$, with the possible exception of the end nodes. Any transmissions that overlap interfere and can not be received. It is assumed that a transmission cannot be heard more than δ nodes away. That is, for some node i in the network, the probability of successful transmission from i to $i + \delta + 1$ is zero. This allows for spatial reuse of the channel.

For each transmission scheme, the transmitter wishes to reliably send M degrees of freedom (dofs) to the receiver. The network is modeled as a set of active nodes, $\mathcal{N}_a$, a set of passive nodes, $\mathcal{N}_p$, a set of links between nodes, $\mathcal{L}$, and a set of states, $\mathcal{S}$. An active node is one which is capable of generating new dofs or acknowledgments. This includes the transmitter and receiver plus any intermediate node capable of recoding or transmitting a packet more than once. Passive nodes (i.e., non-active nodes) are incapable of recoding, and can only forward packets they receive. $\mathcal{N}_a$ and $\mathcal{N}_p$ are determined by the protocol used, but $\mathcal{N}$ is fixed by the network topology. Note that $\mathcal{N}_a \cap \mathcal{N}_p = \emptyset$. Let $\mathcal{N} = \mathcal{N}_a \cup \mathcal{N}_p$.

Let $N_a = |\mathcal{N}_a|$ and $N = |\mathcal{N}|$. Active nodes $i \in \mathcal{N}_a$ are indexed 1 through N_a, where 1 is the transmitter and N_a is the receiver, and nodes further from the transmitter have a higher index. All nodes $j \in \mathcal{N}$ are similarly indexed 1 through N, where 1 is the transmitter and N is the receiver, and nodes further from the transmitter have a higher index. For simplicity, a node and its index are interchangeable, that is, $i \in \mathcal{N}_a$ is equivalent to $i \in \{1, 2, \ldots, N_a\}$, and $j \in \mathcal{N}$ is equivalent to $j \in \{1, 2, \ldots, N\}$.

Each link, $\ell_{i,j} \in \mathcal{L}$, $i, j \in \mathcal{N}$, has an associated $P_{i,j}$, which is the probability of successful reception at node j, but no further, when i transmits. The matrix $P \in \mathbb{R}^{N \times N}$ is defined to have elements $P_{i,j}$, and is a right stochastic matrix. $P_{i,i}$ is the probability that the packet is not received by any node when i transmits. $P_{N,N}$ is defined to be 1. $P_{i,j} = 0$ if $j < i$.

States are indexed from 1 to $|\mathcal{S}|$. For simplicity, a state and its index are interchangeable, so $s \in \mathcal{S}$ is equivalent to $s \in \{1, 2, \ldots |\mathcal{S}|\}$.

For implicit acknowledgment schemes, state $s \in \mathcal{S}$ is a vector $\vec{v} \in \mathbb{N}_a^N$ where each element v_i, $i \in \mathcal{N}_a$ is the number of degrees of freedom at node i. For non-coding schemes, $v_i \in \{0, 1\}$. For explicit acknowledgment schemes, the state is identical, except each state also contains a vector $\vec{a} \in \{0, 1\}^2$ where a_1 is an indicator that an acknowledgment has been sent from the receiver and a_2 is an indicator that an acknowledgment has been received by the transmitter. For full duplex schemes, the state also includes a variable $c \in \{0, 1\}$ which changes every time the transmitter sends a packet.

The network transitions through a stochastic sequence of states, $\{S(t)\}$, where $t \in \mathbb{N} \cup \{0\}$. Furthermore $\forall i, j, k, l \in \mathcal{S}$

$$\begin{aligned} &Pr\{S(t) = j|S(t-1) = i, S(t-2) = k, \ldots, S(0) = l\} \\ &\quad = Pr\{S(t) = j|S(t-1) = i\} \end{aligned}$$

and $Pr\{S(t) = j|S(t-1) = i\}$ depends only on i and j, not on t, so the process is a finite Markov chain (since $|\mathcal{S}|$ is finite). The transition matrix T gives the probability of state transitions, and has elements $T_{i,j}$, where

$$T_{i,j} = Pr\{S(t) = j|S(t-1) = i\}.$$

Without loss of generality, assume the Markov chain starts in state 1 and state $|\mathcal{S}|$ is a trapping state, where state 1 is when nothing has been received by any active node, and state $|\mathcal{S}|$ is when all packets or dofs have been received by the receiver and an acknowledgment has been received by the transmitter, if applicable.

Each state has an associated cost ξ_s, and $\mathcal{S}$ has an associated cost vector $\vec{\xi} \in \mathbb{R}^{|\mathcal{S}|}$. The cost of each state represents the amount of time it takes to transition to that state and is determined by the protocol.

Using the transition matrix T and the cost vector $\vec{\xi}$, it is possible to compute the expected cost of the Markov chain, which corresponds to the expected time to send M degrees of freedom from the transmitter to the receiver. This is given by the first entry in the vector

$$\vec{\tau} = F\vec{\xi'} \tag{1.1}$$

minus the cost of the first state (ξ_1) plus the cost of the last state ($\xi_{\mathcal{S}}$) and any fixed costs, u, due to the propagation delay of the initial packet. The vector $\vec{\xi'} \in \mathbb{R}^{|\mathcal{S}|-1}$ is $\vec{\xi}$ with the last element removed, and F is the fundamental matrix, given by

$$F = (I_{|\mathcal{S}|-1} - T')^{-1},$$

where $I_{|\mathcal{S}|-1}$ is the $|\mathcal{S}|-1 \times |\mathcal{S}|-1$ identity matrix and $T' \in \mathbb{R}^{|\mathcal{S}|-1 \times |\mathcal{S}|-1}$ is the matrix T with the last row and last column removed. The expected completion time is given by

$$E[T_c] = \vec{\tau}_1 - \xi_1 + \xi_{|\mathcal{S}|} + u \tag{1.2}$$

where $\vec{\tau}$ is defined in 1.1.

The variance is the first entry in the vector

$$\vec{\tau}^{(2)} = 2F[(T'\vec{\tau}) \circ \vec{\xi}] + F\vec{\xi}_{\text{sq}} - \vec{\tau}_{\text{sq}} \tag{1.3}$$

where $\vec{\tau}_{\text{sq}}$ is the vector $\vec{\tau}$ with each element squared, and $\circ$ denotes the Hadamard product. The variance is thus

$$\text{Var}(T_c) = \vec{\tau}_1^{(2)} \tag{1.4}$$

where $\vec{\tau}^{(2)}$ is given in 1.3.

Chapter 2

Analysis

2.1 End-to-End Retransmission

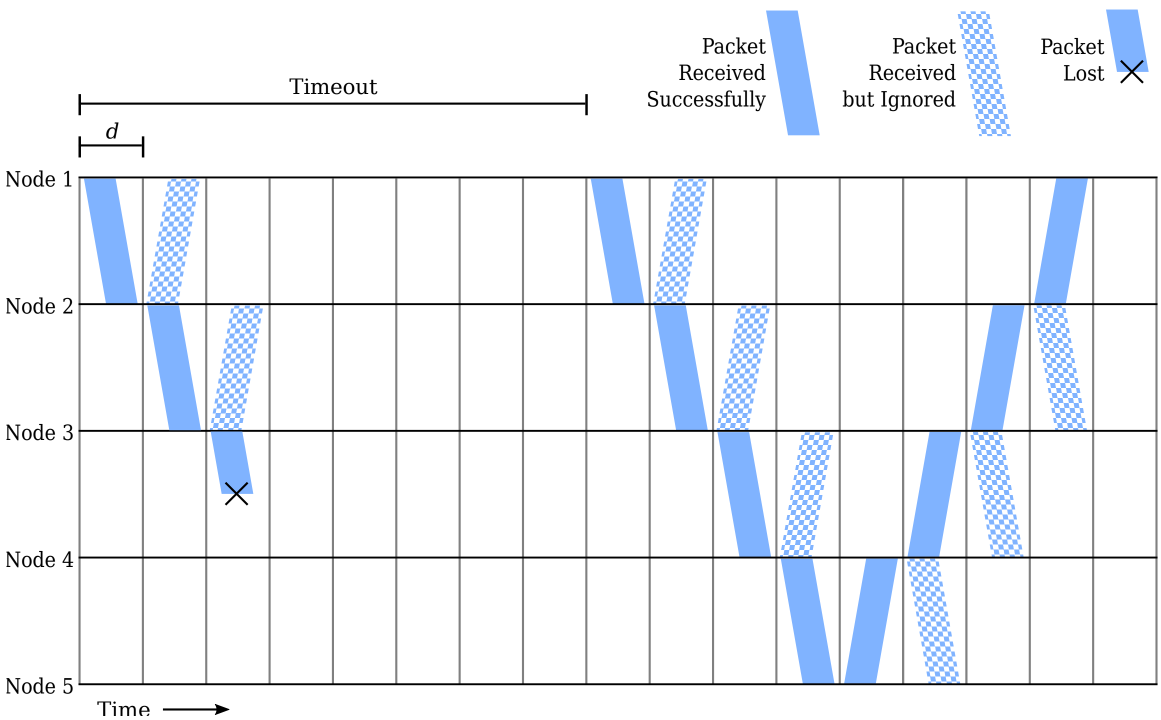

Figure 2-1: Timing diagram illustrating end-to-end retransmission. The protocol is described in §2.1.

The transmitter wishes to send one packet ($M = 1$) to the receiver. The packet is transmitted at intervals of T_0 until an acknowledgment is heard. Each intermediate node acts only to forward packets it receives, and does not use any explicit feedback. See figure 2-1 for an example.

The set $\mathcal{N}_a$ contains two nodes, the transmitter and receiver. The set $\mathcal{S}$ consists of four distinct states, described by the vector (v, a), where $v \in \{0, 1\}$ indicates whether the packet has been received, and $a \in \{0, 1\}$ indicates either an acknowledgment has been received by the transmitter ($a = 1$) or a timeout has occurred ($a = 0$). Explicitly, the states are

State	Description
$s_1 = (0, 0)$	Packet transmitted
$s_2 = (1, 0)$	Packet received, acknowledgment sent
$s_3 = (0, 1)$	Packet not received
$s_4 = (1, 1)$	Packet and acknowledgment received

where s_1 is the starting state, and s_4 is a trapping state.

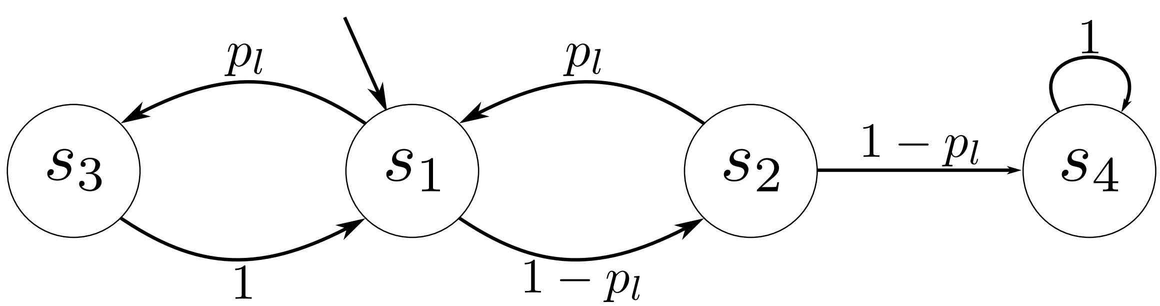

Figure 2-2: Markov chain model for end-to-end retransmission scheme. The starting state is s_1, and s_4 is a trapping state. s_1 transitions to s_3 when the packet is lost or s_2 if the packet is successfully received. s_2 transitions to s_1 if the acknowledgment is lost or s_4 if it is received.

The probability of a packet loss when transmitting from node $i \in \mathcal{N}$ to $i + 1$ is given by $P_{i,i}$. Since the packet losses along each link are independent, the probability of a packet loss on the network one way is given by

$$p_l = 1 - (1 - P_{i,i})^{N-1} . \tag{2.1}$$

The transition matrix T is

$$\begin{bmatrix} 0 & 1-p_l & p_l & 0 \\ p_l & 0 & 0 & 1-p_l \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the Markov chain is shown in figure 2-2.

The cost vector, $\vec{\xi}$, has all elements equal to $T_0/2$, so

$$\vec{\xi} = \left(\frac{T_0}{2}, \frac{T_0}{2}, \frac{T_0}{2}, \frac{T_0}{2}\right)^T.$$

The expected completion time and variance can be computed using equations 1.2 and 1.4.

$$E[T_c] = \frac{T_0}{(1-p_l)^2} \tag{2.2}$$

$$\mathrm{Var}(T_c) = T_0^2 \left[\frac{1-(1-p_l)^2}{(1-p_l)^4}\right]. \tag{2.3}$$

2.2 End-to-End Coding

In this scheme, each relay node again acts only to forward packets it receives, but there are M packets to send and a rateless end-to-end packet erasure code is used. The transmitter sends packets until it receives an acknowledgment. If the receiver receives a packet and has more than M dofs afterward, it sends an acknowledgment. A packet is only advanced one hop per successful transmission. If a node further downstream overhears a packet, it does not store or forward it. Packet losses on each hop are assumed to be independent. Furthermore, it is assumed that transmissions do not cause any interference two or more hops away (that is, $\delta = 1$). Thus, a node can only communicate with its immediate neighbors.

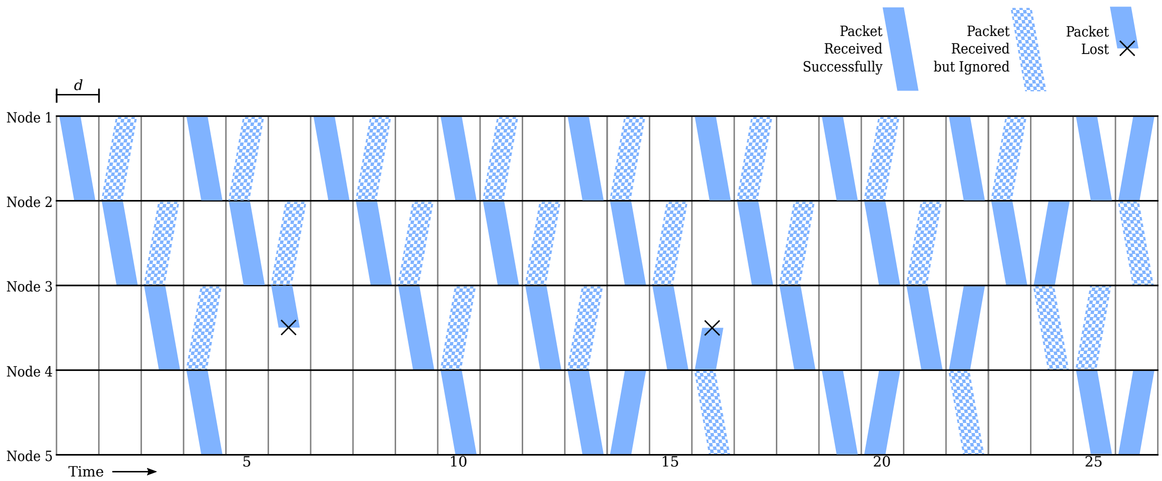

Figure 2-3: Timing diagram illustrating end-to-end coding with a half duplex channel using time division multiplexing to avoid interference. The generation size is $M = 3$. The protocol is described in §2.2.1.

2.2.1 Time Division Multiplexing

All nodes share a single broadcast channel and a single frequency band. A node may not transmit and receive at the same time, and a node cannot successfully receive two transmissions simultaneously. More precisely, if node $i - 1$ and $i + 1$ transmit, the two transmissions will overlap at node i, and i cannot successfully receive either transmission.

Protocol

The protocol for intermediate nodes is as follows. Each node has a cycle of three time slots and two queues from which to transmit packets, the upstream queue and the downstream queue. The upstream queue holds packets from upstream (closer to the transmitter), and the downstream queue holds packets from downstream (closer to the receiver). In the first time slot, a node will receive a transmission (if there is one) from its upstream neighbor and place it in the upstream queue. In the second time slot, the node will receive a transmission (if there is one) from its downstream neighbor and place it in its downstream queue. In the third time slot, the node will attempt to transmit the packet at the head of its downstream queue. If that queue is empty, it will attempt to transmit the packet at the head of its upstream queue.

If both queues are empty, nothing is transmitted. See figure 2-3. A summary of this cycle is given below.

Time	Action at Node n
$t = 1$	Receive from node $n - 1$
$t = 2$	Receive from node $n + 1$
$t = 3$	Transmit

The transmitter will send packets every three time slots and stop when an acknowledgment is received. The receiver will send an acknowledgment (in the appropriate time slot) whenever it has received a packet in the previous time slot and it has received M linearly independent dofs.

Each node's cycle (receive from upstream neighbor, receive from downstream neighbor, transmit) is staggered so no two transmissions interfere.

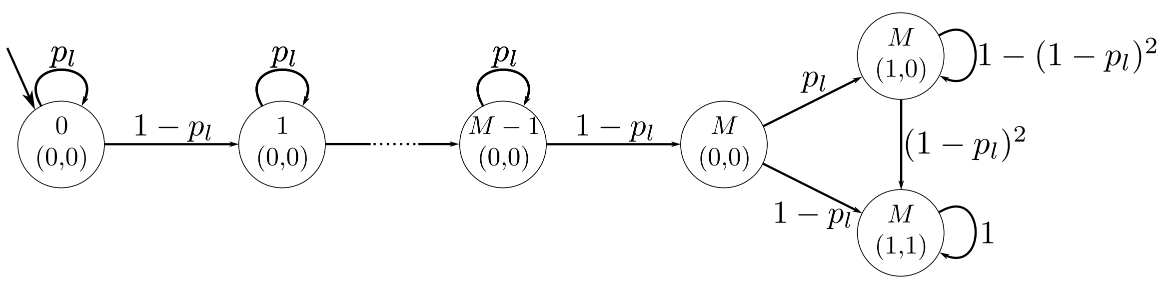

Figure 2-4: Markov chain describing end-to-end coding with a half-duplex channel. The chain starts in $(0, 0, 0)$ and ends in $(M, 1, 1)$.

Analysis

There are two active nodes in this scheme, the transmitter and the receiver. The state of the system can be described as a triple (v, a_1, a_2) where v is the number of dofs at the receiver, a_1 is an indicator that an acknowledgment has been sent, and a_2 is an indicator that an acknowledgment has been received by the transmitter. The valid states of the system can be enumerated:

$$\mathcal{S} = \{(0, 0, 0), (1, 0, 0), \ldots, (M, 0, 0), (M, 1, 0), (M, 1, 1)\}.$$

Let

$$s_i = \begin{cases} (i-1,0,0) & 1 \le i \le M+1 \\ (M,1,0) & i = M+2 \\ (M,1,1) & i = M+3 \end{cases},$$

where $s_i \in \mathcal{S}$. The transition matrix $T \in \mathbb{R}^{(M+3)\times(M+3)}$ can be constructed, and is given by

$$T = \begin{bmatrix} p_l & 1-p_l & 0 & \cdots & 0 & 0 & 0 \\ 0 & p_l & 1-p_l & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & p_l & 1-p_l \\ 0 & 0 & 0 & \cdots & 0 & 1-(1-p_l)^2 & (1-p_l)^2 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

where p_l is given in equation 2.1. Using the states, $\mathcal{S}$ and the transition matrix, T, it is possible to model the protocol as a Markov chain, shown in figure 2-4.

The cost of transitioning to each state is given by

$$\xi_i = \begin{cases} 3 & 1 \le i \le M+1 \\ 6 & i = M+2 \\ 2(N-1) & i = M+3 \end{cases}.$$

The values for ξ_i for $0 \le i \le M+1$ arise from the fact that each node can only transmit every three time slots. If the acknowledgment is lost, it takes twice as long to recover, hence $\xi_{M+2} = 6$. The cost of transmitting the acknowledgment includes the propagation delay, hence $\xi_{M+3} = 2(N-1)$. The value of $\vec{\xi}$ can be seen in the timing diagram (figure 2-3). The fixed costs include the propagation of the first packet sent, and

$$u = N - 1.$$

Using equations 1.2 and 1.4, the mean and variance of the completion time for M

dofs can be computed.

$$\tau_1 = 3\left(\frac{M}{1-p_l}+1\right)+6\frac{p_l}{(1-p_l)^2}$$

thus

$$E[T_c^{\mathrm{TDM}}] = \tau_1 + 3(N-1) - 3, \tag{2.4}$$

and

$$\mathrm{Var}(T_c^{\mathrm{TDM}}) = \tau_1^{(2)} = \frac{9}{(1-p_l)^4}\left[(1-p_l)^3(4-M)+(1-p_l)^2(M-8)+4\right]. \tag{2.5}$$

The fundamental matrix, F, can be found by noting that $(I_{M+2} - T')$ is a bidiagonal matrix, and using well known techniques (such as that presented in [10]) the inverse can be found.

$$\{F\}_{i,j} = \begin{cases} \frac{1}{1-p_l} & i \le j, 1 \le j \le M \\ 1 & i \le M+1, j = M+1 \\ \frac{p_l}{(1-p_l)^2} & i \le M+1, j = M+2 \\ \frac{1}{(1-p_l)^2} & i = M+2, j = M+2 \end{cases}$$

τ can be found by multiplying F by $\vec{\xi'}$. The variance can be found by substituting F, T', τ, and $\vec{\xi'}$ into equation 1.4.

Limit as $M \to \infty$

The limit of the per-packet completion time as M goes to infinity is

$$\lim_{M\to\infty} \frac{\tau_1}{M} = \frac{3}{1-p_l},$$

and the limit of the variance is

$$\lim_{M\to\infty} \frac{\tau_1^{(2)}}{M} = \frac{9p_l}{(1-p_l)^2}.$$

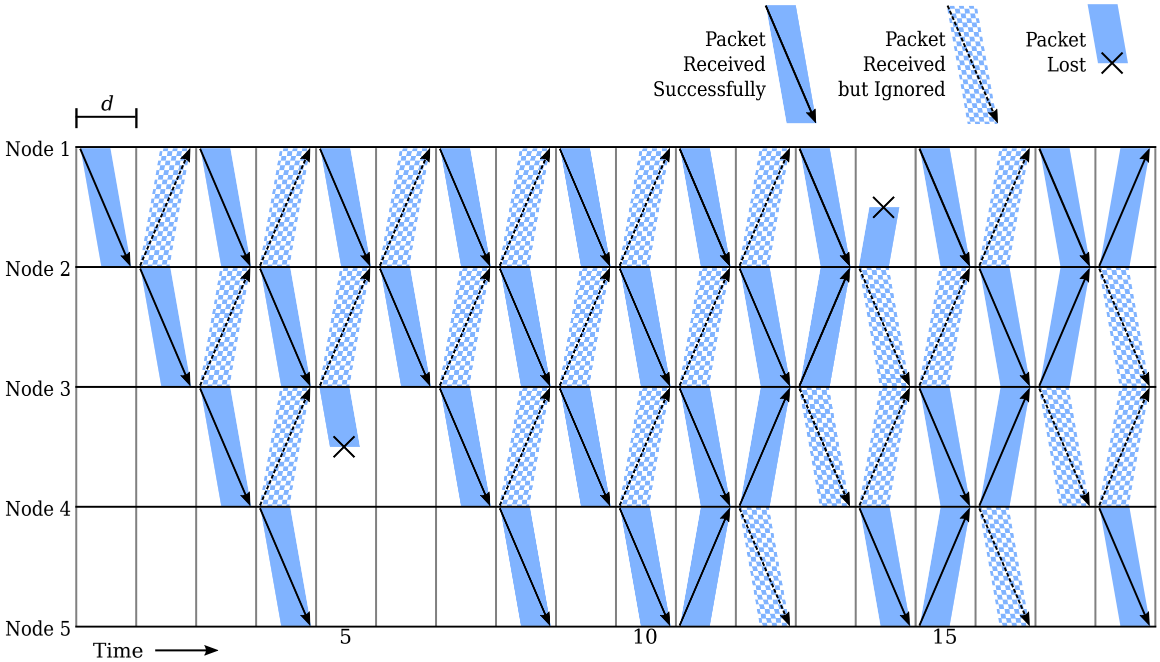

Figure 2-5: Timing diagram illustrating end-to-end coding with a half duplex channel using frequency division multiplexing to avoid interference. The generation size is $M = 3$. The protocol is described in §2.2.2.

2.2.2 Frequency Division Multiplexing

The channel's bandwidth is divided into two equal bands. Each node transmits in one of these frequency bands. A node can receive two packets at once only if they are on different bands. Nodes cannot transmit and receive at the same time.

Protocol

The protocol for intermediate nodes is as follows. Each node has a cycle of two time slots and two queues from which to transmit packets, the upstream queue and the downstream queue. The upstream queue holds packets from upstream (closer to the transmitter), and the downstream queue holds packets from downstream (closer to the receiver). In the first time slot, a node receives packets on both frequency bands if the packets exist. It then places packets from upstream in its upstream queue, and packets from downstream in its downstream queue. In the second time slot, the node attempts to transmit the packet at the head of its downstream queue. If that queue is empty, it will attempt to transmit the packet at the head of its upstream queue.

If both queues are empty, nothing is transmitted. See figure 2-5. A summary of this cycle is given below.

Time	Action at Node n
$t = 1$	Receive from node $n-1$ and $n+1$ simultaneously
$t = 2$	Transmit

The transmitter will send packets every two time slots and stop when an acknowledgment is received. The receiver will send an acknowledgment (in the appropriate time slot) whenever it has received a packet in the previous time slot and it has received M linearly independent dofs.

Each node's cycle (receive, transmit) is staggered so no two transmissions interfere. Transmit frequency bands are assigned such that they alternate every two nodes. For example, nodes 1 and 2 are assigned band 1, nodes 3 and 4 are assigned band 2, nodes 5 and 6 are assigned band 1, etc.

Analysis

The analysis of the completion time is the same as for TDM except a different cost vector $\vec{\xi}$ is used. The set of states, $\mathcal{S}$, and the transition probability matrix, T, are the same, and are given in §2.2.1. See figure 2-4 for the Markov chain.

The cost vector is given by

$$\xi_i = \begin{cases} 2\alpha & 1 \le i \le M+1 \\ 4\alpha & i = M+2 \\ \alpha(N-1) & i = M+3 \end{cases}$$

where α is the factor by which packet duration increases due to the reduced transmission bandwidth, i.e.,

$$\alpha = \frac{d_{\text{FDM}}}{d_{\text{TDM}}}$$

where d_{FDM} and d_{TDM} are the packet durations when using FDM and TDM, respectively. The values for ξ_i for $1 \le i \le M+1$ arise from the fact that each node can transmit once in every two time slots. If the acknowledgment is lost, it takes twice as long to recover, hence $\xi_{M+2} = 4\alpha$. The cost of transmitting the acknowledgment includes the propagation delay, hence $\xi_{M+3} = 2\alpha(N-1)$. The fixed costs include the propagation of the first packet sent, and

$$u = \alpha(N-1).$$

Using equations 1.2 and 1.4, the mean and variance of the completion time for M dofs can be computed.

$$\tau_1 = 2\alpha\left(\frac{M}{1-p_l}+1\right) + 4\alpha\frac{p_l}{(1-p_l)^2}$$

and thus

$$E[T_c^{\mathrm{TDM}}] = \tau_1 + 2\alpha(N-1) - 2\alpha, \tag{2.6}$$

and

$$\mathrm{Var}(T_c^{\mathrm{TDM}}) = \tau_1^{(2)} = \frac{4\alpha^2}{(1-p_l)^4}\left[(1-p_l)^3(4-M) + (1-p_l)^2(M-8) + 4\right]. \tag{2.7}$$

The analysis is similar to that for TDM and will not be repeated here.

Limit as $M \to \infty$

The limit of the per-packet completion time as M goes to infinity is

$$\lim_{M\to\infty}\frac{\tau_1}{M} = \frac{2\alpha}{1-p_l},$$

and the limit of the variance is

$$\lim_{M\to\infty}\frac{\tau_1^{(2)}}{M} = \frac{4\alpha^2 p_l}{(1-p_l)^2}.$$

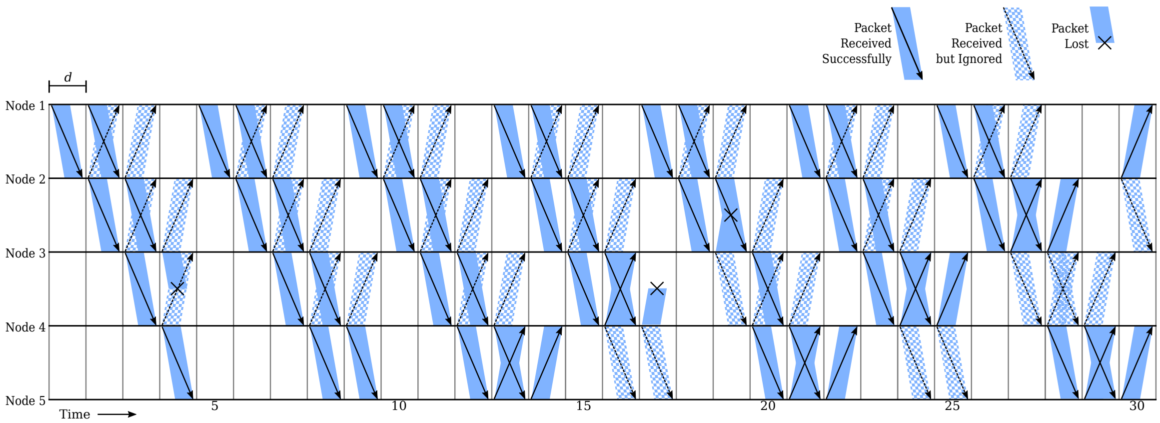

Figure 2-6: Timing diagram illustrating end-to-end coding with a full duplex channel. The generation size is $M = 4$. The protocol is described in §2.2.3.

2.2.3 Full Duplex Channel

All nodes share a single broadcast channel and a single frequency band. Nodes *can* transmit and receive at the same time, however, nodes cannot receive more than one packet at a time. If node $i - 1$ and $i + 1$ transmit, the two transmissions will overlap at node i, and i cannot successfully receive either transmission. Although there may be some cost to using a full duplex channel as opposed to a half duplex, it is not considered here.

Protocol

The protocol for intermediate nodes is as follows. Each node has a cycle of four time slots and two queues from which to transmit packets, the upstream queue and the downstream queue. The upstream queue holds packets from upstream (closer to the transmitter), and the downstream queue holds packets from downstream (closer to the receiver). In the first two time slots, the node receives packets from upstream and places them in its upstream queue. In the third and fourth time slots, the node receives packets from downstream and places them in its downstream queue. In the second and third time slots, the node will attempt to transmit packets from its downstream queue. If that queue is empty, it transmits packets from its upstream queue. If both queues are empty, nothing is transmitted. See figure 2-6 for an illustration of the protocol. A summary of this cycle is given below.

Time	Action at Node n
$t = 1$	Receive from node $n - 1$
$t = 2$	Receive from node $n - 1$ and transmit
$t = 3$	Receive from node $n + 1$ and transmit
$t = 4$	Receive from node $n + 1$

The transmitter also has a cycle of four time slots. It will send new dofs in its first two time slots and receive packets in its second and third. The fourth time slot is unused. The receiver will send an acknowledgment whenever it has received a packet in the previous time slot and it has received M linearly independent dofs.

Each node's cycle is staggered so no two transmissions interfere.

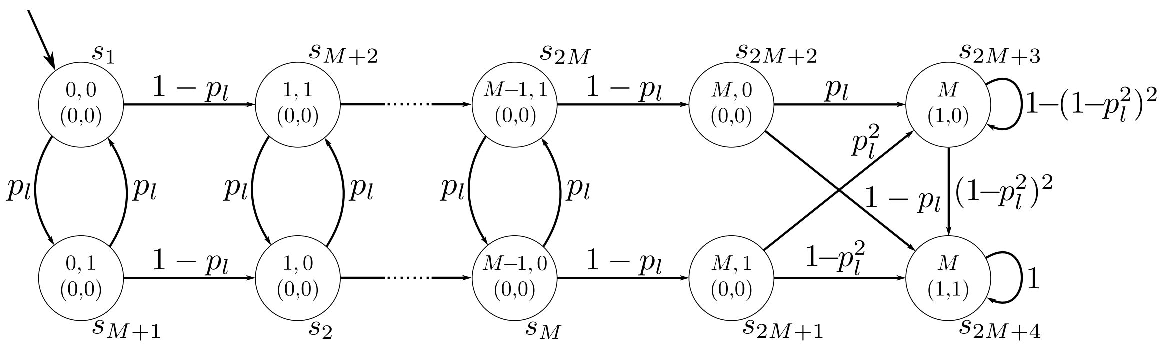

Figure 2-7: Markov chain describing end-to-end coding with a full-duplex channel. The chain starts in $(0, 0, 0, 0)$ and ends in $(M, 1, 1)$. Unlike a half-duplex channel, each state except the last two has a duplex bit (top row, second entry) which changes with each transmission. Note that in this case, M is odd. If M is even, then states s_M, s_{2M}, s_{2M+1}, and s_{2M+2} should have the duplex bit flipped.

Analysis

There are two active nodes in this scheme, the transmitter and the receiver. The state of the system can be described as a quadruple (v, c, a_1, a_2) where v is the number of dofs at the receiver, a_1 is an indicator that an acknowledgment has been sent, a_2 is an indicator that an acknowledgment has been received by the transmitter, and $c \in \{0, 1\}$ is a duplex bit which changes after every transmission. The valid states of the system can be enumerated:

$$\begin{aligned} \mathcal{S} \;= \{ &\; (0,0,0,0),(0,1,0,0),(1,0,0,0),(1,1,0,0)\ldots, \\ &\; (M,0,0,0),(M,1,0,0),(M,x,1,0),(M,x,1,1)\}. \end{aligned}$$

The final two states do not require a duplex bit. Let s_i be labeled as shown in figure 2-7. The transition matrix $T \in \mathbb{R}^{(2M+4)\times(2M+4)}$ can be constructed, and is given by

$$T = \begin{bmatrix} [0]_{M\times M} & R & X^{(1)} \\ R & [0]_{M\times M} & X^{(2)} \\ [0]_{4\times M} & [0]_{4\times M} & D \end{bmatrix}$$

where p_l is given in equation 2.1, $[0]_{x\times y}$ is an $x \times y$ matrix of all zeros, $R \in \mathbb{R}^{M\times M}$ is given by

$$R = \begin{bmatrix} p_l & 1-p_l & 0 & \cdots & 0 \\ 0 & p_l & 1-p_l & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1-p_l \\ 0 & 0 & 0 & \cdots & p_l \end{bmatrix} \tag{2.8}$$

where the entries on the main diagonal are p_l and the entries on the superdiagonal are $1-p_l$. $X^{(1)} \in \mathbb{R}^{M\times 4}$ has $X^{(1)}_{M,1} = 1-p_l$ and all other elements equal to zero. $X^{(2)} \in \mathbb{R}^{M\times 4}$ has $X^{(2)}_{M,2} = 1-p_l$ and all other elements equal to zero. That is,

$$X^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 1-p_l & 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

and

$$X^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 0 & 1-p_l & 0 & 0 \end{bmatrix}. \tag{2.10}$$

$D \in \mathbb{R}^{4\times 4}$ is given by

$$D = \begin{bmatrix} 1 & 0 & p_l^2 & 1-p_l^2 \\ 0 & 0 & p_l & 1-p_l \\ 0 & 0 & 1-(1-p_l^2)^2 & (1-p_l^2)^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

Using the states, $\mathcal{S}$, and the transition matrix, T, it is possible to model the protocol as a Markov chain, shown in figure 2-7.

The cost of transitioning to each state is given by

$$\xi_i = \begin{cases} 1 & 1 \le i \le M \\ 3 & M+1 \le i \le 2M \\ 1 & i = 2M+1 \\ 4 & i = 2M+2 \\ 7 & i = 2M+3 \\ 3N-4 & i = 2M+4 \end{cases}.$$

The values for ξ_i for $1 \le i \le M$ and $M+1 \le i \le 2M$ arise from the four time slot cycle of the protocol. The value for ξ_{M+1} is one since there is no cost to sending the acknowledgment one slot later (see the timing diagram, figure 2-6). The cost ξ_{2M+2} is the regular cost of transmitting (3) plus 1 that is not included in either of the next possible states. The value for ξ_{2M+3} represents the time it takes to receive another dof at the receiver to trigger an acknowledgment, and ξ_{2M+4} represents the time it takes to send an acknowledgment, including propagation delay. An extra $N-4$ must be added to account for the fixed propagation delay, so

$$u = N - 4.$$

Using equations 1.2 and 1.4, the mean and variance of the completion time for M dofs can be computed.

$$\tau_1 = \frac{2M-1}{1-p_l} + 2\left(1 - r^M\right) + \frac{1}{2}\left(1 + r^M\right) + \left[\frac{p_l + 1}{2(1-p_l)} + \frac{7p_l(p_l+1)}{2\left(1-p_l^2\right)^2}\right]\left(1 - r^{M+1}\right)$$

where

$$r = \frac{p_l - 1}{p_l + 1}.$$

The expected completion time is thus

$$E[T_c^{\mathrm{FD}}] = \tau_1 + 4N - 9. \tag{2.12}$$

The second moment of the completion time, $\tau_1^{(2)}$ admits no simple closed form.

The fundamental matrix, F, can be found for a general M by writing $I - T'$ in block form.

$$I - T' = \begin{bmatrix} I_M & -R & B_1 \\ -R & I_M & B_2 \\ [0]_{3\times M} & [0]_{3\times M} & I_3 - D' \end{bmatrix}$$

where I_k is the $k \times k$ identity matrix, $[0]_{i\times j}$ is the $i \times j$ matrix of all zeros, D' is the matrix D (given in 2.11) with the last row and last column removed, R is given in 2.8, and B_1 and B_2 are $X^{(1)}$ and $X^{(2)}$ (given in 2.9 and 2.10) with the last column removed. Define

$$A = \begin{bmatrix} I_M & -R \\ -R & I_M \end{bmatrix}$$

and

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

so

$$I - T' = \begin{bmatrix} A & B \\ [0]_{3\times 2M} & I_3 - D' \end{bmatrix}$$

and

$$F = (I - T')^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}B(I_3 - D')^{-1} \\ [0]_{3\times 2M} & (I_3 - D')^{-1} \end{bmatrix}.$$

Solving for A^{-1} yields

$$A^{-1} = \begin{bmatrix} (I - R^2)^{-1} & R(I - R^2)^{-1} \\ R(I - R^2)^{-1} & (I - R^2)^{-1} \end{bmatrix}$$

where the elements of each submatrix are

$$\left\{(I - R^2)^{-1}\right\}_{i,j} = \begin{cases} 0 & i > j \\ \frac{1}{2(1-p_l)}\left[1 - \left(\frac{p_l-1}{p_l+1}\right)^{j-i+1}\right] & i \leq j \end{cases}$$

$$\left\{R(I - R^2)^{-1}\right\}_{i,j} = \begin{cases} 0 & i > j \\ \frac{1}{2(1-p_l)}\left[1 + \left(\frac{p_l-1}{p_l+1}\right)^{j-i+1}\right] & i \leq j \end{cases}$$

$$\left\{-A^{-1}B(I_3 - D')^{-1}\right\}_{i,j} = \begin{cases} \frac{1}{2}\left[1 - r^{M-i+1}\right] & j = 1, 1 \leq i \leq M \\ \frac{1}{2}\left[1 + r^{M-i+1}\right] & j = 2, 1 \leq i \leq M \\ \frac{1}{2}\left[1 + r^{2M-i+1}\right] & j = 1, M < i \leq 2M \\ \frac{1}{2}\left[1 - r^{2M-i+1}\right] & j = 2, M < i \leq 2M \\ \frac{p_l(p_l+1)}{2(1-p_l^2)^2}\left[1 - r^{M-i+2}\right] & j = 3, 1 \leq i \leq M \\ \frac{p_l(p_l+1)}{2(1-p_l^2)^2}\left[1 + r^{2M-i+2}\right] & j = 3, M < i \leq 2M \end{cases}$$

and

$$(I_3 - D')^{-1} = \begin{bmatrix} 1 & 0 & \frac{p_l^2}{(1-p_l^2)^2} \\ 0 & 1 & \frac{p_l}{(1-p_l^2)^2} \\ 0 & 0 & \frac{1}{(1-p_l^2)^2} \end{bmatrix}.$$

Multiplying F by $\vec{\xi}$ yields τ.

Limit as $M \to \infty$

The limit of the per-packet completion time as M goes to infinity is

$$\lim_{M\to\infty} \frac{\tau_1}{M} = \frac{2}{1-p_l}.$$

This can be easily found by noting that $-1 < r < 1$ and using the squeeze theorem [20].

2.3 Link-by-Link Retransmissions with Opportunistic Routing

The transmitter wishes to send one packet ($M = 1$) to the receiver. Each node transmits the packet until it is acknowledged (implicitly) by a downstream node. The packet is forwarded by node i only if for any nodes j with the packet, $i \geq j$. In other words, the packet is forwarded by the closest node with the packet to the destination. It is assumed that the implicit acknowledgment is always heard, as bit errors which would normally trigger an erasure can be tolerated. Any transmission can only be successfully received by nodes at most δ hops away.

(Although this might not be considered routing since the network topology is linear, this thesis will use "opportunistic routing" to refer to the practice of selecting the forwarding node based on which nodes receive the packet.)

Protocol

Each packet contains the address of the source node in its header. For intermediate nodes, the protocol is as follows. Let the node's address be i, and the source address from the packet header be j. If $j > i$, the packet is discarded and nothing is done. The node computes its transmission priority, $r = \delta-(i-j)$. If $r < 0$ the packet is discarded. This corresponds to the packet being successfully received more than δ hops from the transmitting node. If $r = 0$, the packet is retransmitted immediately. Otherwise,

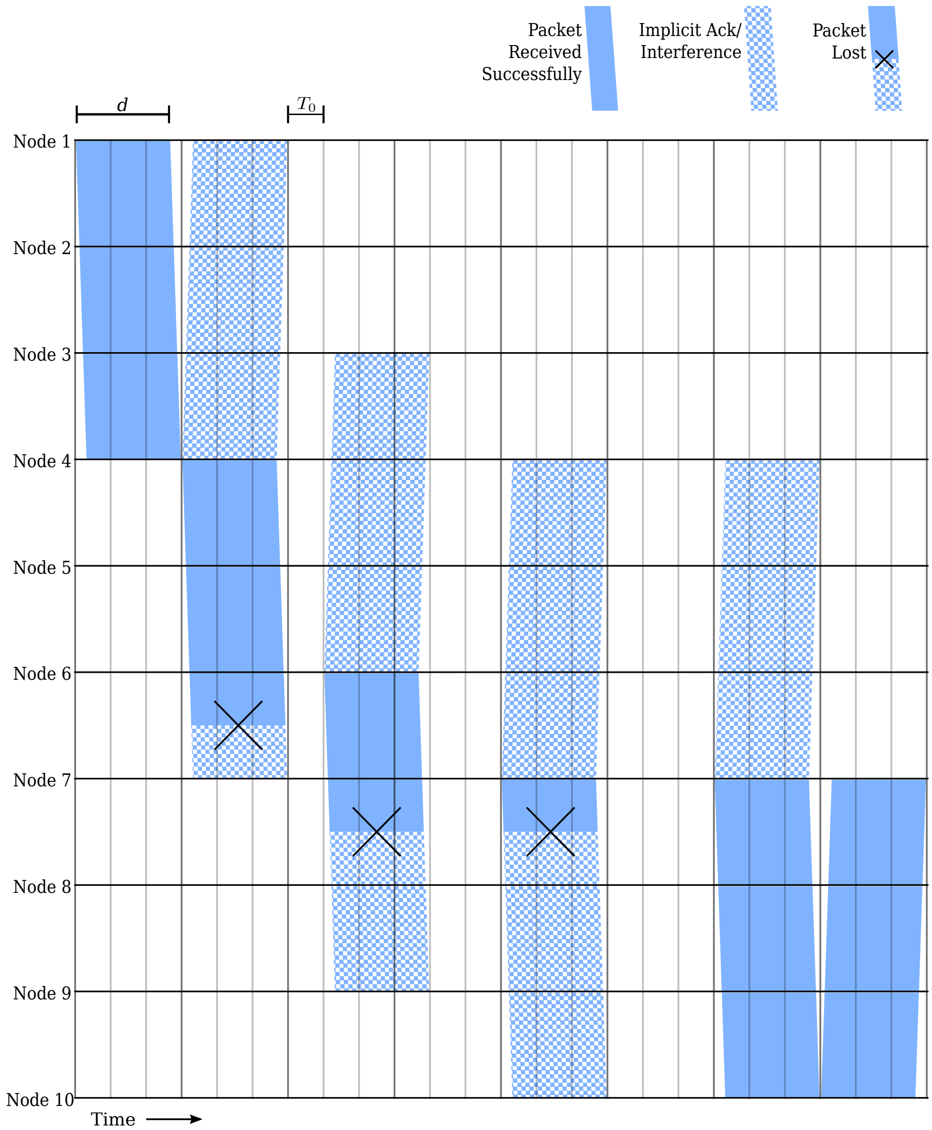

Figure 2-8: Timing diagram illustrating link-by-link retransmission with opportunistic routing. The protocol is described in 2.3.

the node waits for an interval rT_0. If the node receives an implicit acknowledgment from further downstream in this interval, the packet is discarded. Otherwise, it is transmitted at the end of the interval. After a node transmits a packet, it will wait for an interval δT_0. If an implicit acknowledgment is not heard within that interval, the packet is retransmitted at the end of that interval.

The transmitter (Node 1) will transmit the packet right away, and then wait for an interval δT_0. If an implicit acknowledgment is not heard within that interval, the packet is retransmitted at the end of that interval. When the receiver (Node N) receives the packet, it will retransmit it to acknowledge successful reception.

The interval T_0 is set to be twice the propagation delay between two nodes plus the time needed to determine if a packet has been transmitted from downstream.

See figure 2-8 for an illustration of the protocol.

Analysis

All N nodes are active ($N_a = N$ and $N_p = 0$). The state of the system can be described by a vector $\vec{v} \in \mathbb{R}^N$ where v_i represents the number of dofs at node $i \in \mathcal{N}$. In this case, $v_i \in \{0, 1\}$. However, since it is only necessary to know the furthest node to which the packet has arrived, the state can be described by the index of that node. (That is, if a packet makes it to node i, whether or not it makes it to node j, $j < i$ is irrelevant, so the state can be fully described by the index of the most downstream node to receive the packet.) Thus the states of the system are simply

$$\mathcal{S} = \{1, 2, \dots, N\}$$

where state 1 is the starting state and state N is the final state. Using the same indices for states as those for nodes, the transition matrix T is simply

$$T = P$$

where P has entries $P_{i,j}$, $i, j \in \mathcal{N}$, which represent the probability of successful reception at node j when i transmits. $P_{i,i}$ is the probability that the packet is not received by any node when i transmits. $P_{N,N}$ is defined to be 1. $P_{i,j} = 0$ if $j < i$.

The average cost of transitioning from each state is given by

$$\xi_i = d + \sum_{j=i}^{\min(i+\delta,N)} P_{i,j} \left[\min(i+\delta, N) - j\right]$$

where d is the duration of a packet, and $i, j \in \mathcal{N}$.

Using equation 1.2, the expected completion time can be computed with ξ and T defined above. Define

$$E[T_{N,1}] = \vec{\tau}_1 \tag{2.13}$$

as the expected time to send one packet N hops.

Since the cost of each transmission can vary, $\mathrm{Var}(T_{N,1})$ cannot be computed directly with equation 1.4. However, if $P_{i,j}$ depends only on $j - i$, the variance can be computed. Define $\{X_t\}_{t=1}^{\infty}$ to be a sequence of random variables where X_t is the duration of the t^{th} transmission, including the wait time, and H to be the total number of transmissions. $E[H]$ and $\mathrm{Var}(H)$ can be computed (these represent the mean and variance, respectively, of the absorption time for the Markov chain) and are

$$E[H] = \{F\vec{e}\}_1$$

and

$$\mathrm{Var}(H) = \{(2F - I_{N-1})F\vec{e} - (F\vec{e})_{\text{sq}}\}_1$$

where $\vec{e}$ is a vector of all ones.

$$E[X_t] = \sum_{i=0}^{\delta} P_{1,i}(d + \delta - i)$$

and

$$E[X_t^2] = \sum_{i=0}^{\delta} P_{1,i}(d + \delta - i)^2.$$

Let

$$E[X_t] = E[X] \qquad \forall t \in \mathbb{N}$$

and

$$E[X_t^2] = E[X^2] \qquad \forall t \in \mathbb{N}$$

and define

$$S_H = \sum_{t=1}^{H} X_t$$

which is equal to the completion time. Using the law of total variance [18],

$$\begin{aligned}
\mathrm{Var}(S_H) &= E_N[\mathrm{Var}(S_H|H)] + \mathrm{Var}_H\left(E[S_H|H]\right) \\
&= E_H[H\mathrm{Var}(X)] + \mathrm{Var}_H(HE[X]) \\
&= E[H]\mathrm{Var}(X) + E[X]^2\mathrm{Var}(H),
\end{aligned}$$

and

$$\mathrm{Var}(T_{N,1}) \leq \mathrm{Var}(S_H) = E[H]\mathrm{Var}(X) + E[X]^2\mathrm{Var}(H) \tag{2.14}$$

which is an upper bound the variance of the completion time to send one packet from node 1 to N. Equality holds if the cost of transmitting a packet is the same for each node. Although the cost, X_t, has the same distribution for most of the network, it is different near node N where there are fewer than δ hops left. Since the mean and variance decrease near node N, 2.14 is an upper bound to $\mathrm{Var}(T_{N,1})$.

Note that $P_{ij} = 0$ for $j - i > 1$ (i.e., $\delta = 1$) is the case when opportunistic routing is not used.

2.4 Opportunistic Routing with Network Coding

The transmitter wishes to send M packets to the receiver. Each node transmits the packets until they are all acknowledged (implicitly) by downstream nodes. Nodes transmit whenever there is an opportunity and are not listening for an implicit acknowledgment. Any transmission can only be successfully received by nodes at most

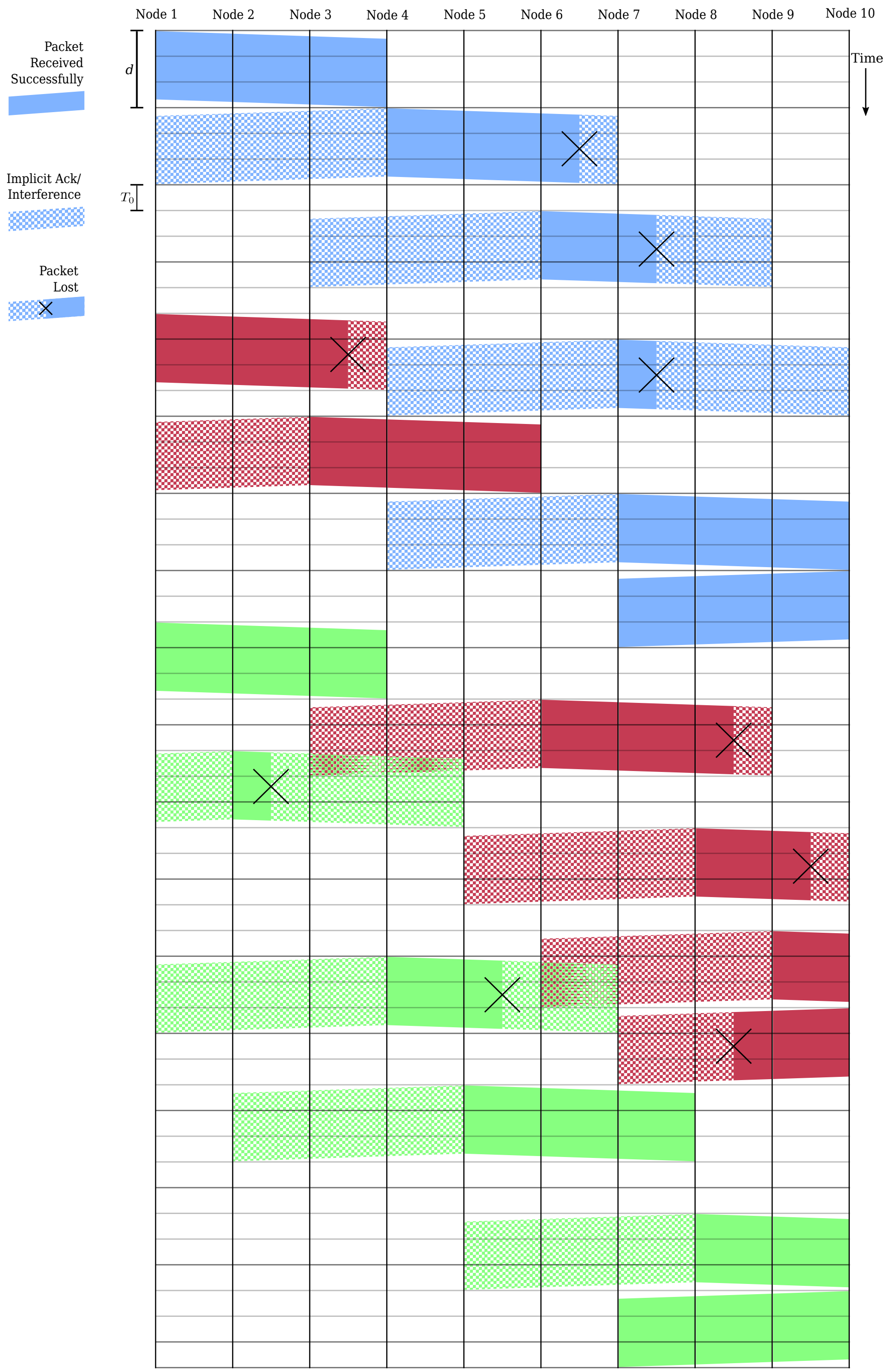

Figure 2-9: Timing diagram illustrating link-by-link retransmission with opportunistic routing and network coding. The protocol is described in 2.4. The different colors for the packets represent the number of dofs used to create them. The blue packets are created with one dof, the red, two, and the green, three.

δ hops away. Nodes cannot receive two packets at the same time. If two transmissions overlap at a node, the node cannot receive either transmission, but can detect that a transmission is occurring.

Protocol

Each packet contains the address of the source node in its header. Node 1 starts with M dofs. For all nodes except the last, the protocol is as follows. Let the node's address be i, and the source address from the packet header be j. Let

$$r = \min[\delta - (i - j), N - i].$$

Let d be the duration of a transmission and

$$T_0 = 2t_p + t_d$$

where t_p is the propagation time between two adjacent nodes, and t_d is the time required to detect whether or not a transmission is occurring. Each node, $i \in \mathcal{N}$ keeps a dof counter, dofs_i; an acknowledgment counter, acks_i; and a silent timer, $t_{s,i}$. If $t_{s,i} > 0$, the node will not transmit. Nodes also keep the dofs and their corresponding coefficients in memory.

When node i successfully receives a complete packet, the following occurs.

1. If the new packet contains an innovative packet, store it and set

$$\text{dofs}_n = \text{dofs}_n + 1.$$

2. If the packet is from upstream ($i < j$), set $t_{s,i} = rT_0$.

3. If the packet did not contain a new dof and $j - i = 1$, then set a "send ack" flag.

4. If the packet is from downstream ($i > j$), set

$$t_{s,i} = \min[(\delta - 1)T_0 + d, (N - i - 1)T_0 + d]$$

and

$$\text{acks}_i = \max(\text{acks}_i, \text{dofs}_j).$$

When node i detects a complete packet has been sent but contains errors, $t_{s,i}$ is set to

$$t_{s,i} = \min[(\delta - 1)T_0 + d, (N - i - 1)T_0 + d]$$

and the packet is discarded.

Before node i can transmit, all of the following conditions must be met:

- $t_{s,i} = 0$
- Node i is neither transmitting nor receiving
- Either $\text{acks}_i < \text{dofs}_i$ or the "send ack" flag is set.

When all the conditions are met, node i generates a random linear combination of all packets in its memory and transmits it.

Analysis

All N nodes are active ($N_a = N$ and $N_p = 0$). The state of the system can be described by a vector $\vec{v} \in \mathbb{R}^N$ where v_i represents the number of dofs at node $i \in \mathcal{N}$. However, one assumption needs to be made in order for this process to be Markovian: that $\{v_i\}$ is non-increasing with respect to i. Stated differently, for any transmission from node i, if it is successfully received at node j, then it is also received at all nodes $k,\ i < k < j$.

Consider a state $s \in \mathcal{S}$ described by dof vector $\vec{v}$. Let $\mathcal{Q}_s$ be the set of possible transmitting nodes in this state, where

$$\begin{aligned} \mathcal{Q}_s = \{ \quad & n_i : n_i \in \mathcal{N}, n_i - n_j \geq \delta + 2\ \forall n_i, n_j, \\ & v_{n_j} > v_{n_i}\ \forall j > i, \\ & n_{i+1} - (n_i + 1) = \delta + 1 \vee v_{n_i+1} < v_{n_i} \}. \end{aligned}$$

In other words, each node n_i that transmits must obey the following rules:

1. the distance between any two transmitting nodes must be at least $\delta + 2$ hops,

2. the number of degrees of freedom at successive transmitters is strictly decreasing, and

3. the next node $(n_i + 1)$ cannot transmit because either it is too close to the next transmitting node downstream or it has fewer dofs than n_i.

Let $\mathcal{R}_s$ be the set of potential receiving nodes, where

$$\mathcal{R}_s = \{n_i : n_i \in \mathcal{N}, \exists n_j \in \mathcal{Q}_s : 0 \leq n_i - n_j \leq \delta, \nexists n_k \in \mathcal{Q}_s : 0 < n_j - n_i \leq \delta\}.$$

In other words, each potential receiver node n_i must meet the following conditions:

1. there exists a transmitting node upstream within δ hops of n_i or n_i is transmitting, and

2. there are no possible interfering nodes downstream.

The transition matrix, T, can be constructed by considering the transitions between every pair of states. Let s and s' be two states with respective dof vectors $\vec{v}$ and $\vec{v}'$. For any node $i \notin \mathcal{R}_s$, $v_i = v_i'$. For any node $i \in \mathcal{R}_s$, the probability that $v_i' = v_i + 1$ can be computed as

$$\Pr\{v_i' = v_i + 1\} = \begin{cases} 1 & v_{i+1}' = v_{i+1} + 1 \\ P_{n_t,i} & \text{otherwise} \end{cases} \tag{2.15}$$

where

$$n_t = \max(\mathcal{Q}_s) \text{ s.t. } n_t \leq i,$$

or put simply, the node which is transmitting to i. Define

$$\mathcal{R}_s' = \{i : i \in \mathcal{N}, v_i' = v_i + 1\} \cup \mathcal{Q}_s,$$

and note that $\mathcal{R}'_s \subseteq \mathcal{R}_s$, then using 2.15,

$$T_{s,s'} = \prod_{i \in \mathcal{R}'_s} \Pr\{v'_i = v_i + 1\}. \tag{2.16}$$

The cost vector, $\vec{\xi}$, is defined similarly to when $M = 1$ (see §2.3), and ξ_s represents the average cost of transitioning to state $s \in \mathcal{S}$, accounting for the packet duration, d, and the expected delay before the receiver transmits.

Although it is possible to define each element of T, the number of states increases exponentially, where

$$|\mathcal{S}| = \mathcal{O}[(M+1)^{N-1}]$$

so it is not feasible to perform any analysis for values of M and N of interest.

Chapter 3

Simulation

3.1 Notation

Variable	Description
N	Number of nodes
M	Generation size
$P_{i,j}$	Probability of successfully transmitting from i to j but no further
p_l	Probability of packet loss on whole network one way: $p_l = 1 - (1 - P_{i,i})^{N-1}$

3.2 Simulation Setup

3.2.1 End-to-End Retransmission

See §2.1 for a description of the protocol and figure 2-1 for a timing diagram. The simulation counts the number of attempts until the first successful transmission of one packet from the transmitter to the receiver and an acknowledgment from the receiver to the transmitter. For each attempt, a random vector of length $2(N-1)$ is generated, where each entry is a Uniform(0,1). If any entry is greater than $P_{i,i}$ then the attempted transmission fails. The number of attempts required for one successful transmission is recorded, and the process is repeated for 100 trials varying number of nodes in the network and loss probability, $P_{i,i}$.

3.2.2 End-to-End Coding

Time Division Multiplexing

See §2.2.1 for a description of the protocol and figure 2-3 for a timing diagram. The simulation sends dofs, spaced three time slots apart to avoid interference and waits for an acknowledgment. Because of the duplexing scheme, the transmitter does not need to stop sending packets to listen for the acknowledgment.

For each attempt, a random vector of length $N-1$ is generated, where each entry is a Uniform(0,1). If any entry is greater than $P_{i,i}$ then the attempted transmission fails. If M or more packets arrive at the receiver, the simulation attempts to send an acknowledgment to the transmitter using the same method.

The total time spent sending M dofs is recorded, and the process is repeated for 100 trials varying the number of nodes in the network and the loss probability, $P_{i,i}$.

Frequency Division Multiplexing

See §2.2.2 for a description of the protocol and figure 2-5 for a timing diagram. The simulation sends dofs, spaced two time slots apart and waits for an acknowledgment. Only a two time slot spacing is necessary because nodes can receive two packets at the same time due to the frequency division. Because of the duplexing scheme, the transmitter does not need to stop sending packets to listen for the acknowledgment.

For each attempt, a random vector of length $N-1$ is generated, where each entry is a Uniform(0,1). If any entry is greater than $P_{i,i}$ then the attempted transmission fails. If M or more packets arrive at the receiver, the simulation attempts to send an acknowledgment to the transmitter using the same method.

The total time spent sending M dofs is recorded, and the process is repeated for 100 trials varying the number of nodes in the network and the loss probability, $P_{i,i}$.

Full Duplex Transmission

See §2.2.3 for a description of the protocol and figure 2-6 for a timing diagram. The simulation counts the number of attempts until there are M successful transmissions, weighting each transmission with the appropriate cost. Packets are transmitted two at a time, where the first packet has a cost of 3, and the second, a cost of 1. Because of the duplexing scheme, the transmitter does not need to stop sending packets to listen for the acknowledgment.

For each attempt, a random vector of length $N-1$ is generated, where each entry is a Uniform(0,1). If any entry is greater than $P_{i,i}$ then the attempted transmission fails. If M or more packets arrive at the receiver, the simulation attempts to send an acknowledgment to the transmitter, sending one for each dof greater than or equal to M that is received.

The total time spent sending M dofs is recorded, and the process is repeated for 100 trials varying the number of nodes in the network and the loss probability, $P_{i,i}$.

3.2.3 Opportunistic Routing with Network Coding

See §2.4 for a description of the protocol. The simulation considers N nodes. Time is slotted, and the length of each packet is d time slots. It takes one time slot for a node to detect if a packet is being transmitted. The simulation keeps a data structure for each node which stores the following information:

- Node ID
- Silent timer
- Dof count
- Acknowledgment count
- "Send ack" flag
- Coefficient matrix

- Source ID (for received transmissions)

Each step of the simulation occurs in two phases—the transmit and process phases—and corresponds to one time step. In the transmit phase, each node decides whether it should transmit. When node i decides to transmit, it writes its ID and packet coefficients to each potential receiving node, indicating whether there is interference. In the process phase, each node updates its state as necessary, adjusting the silent timer, dof and acknowledgment counts, coefficient matrix, and "send ack" flag. The simulation ends when there are M dofs at the receiver (node N).

3.3 Simulation Results and Discussion

3.3.1 Single Packet Schemes

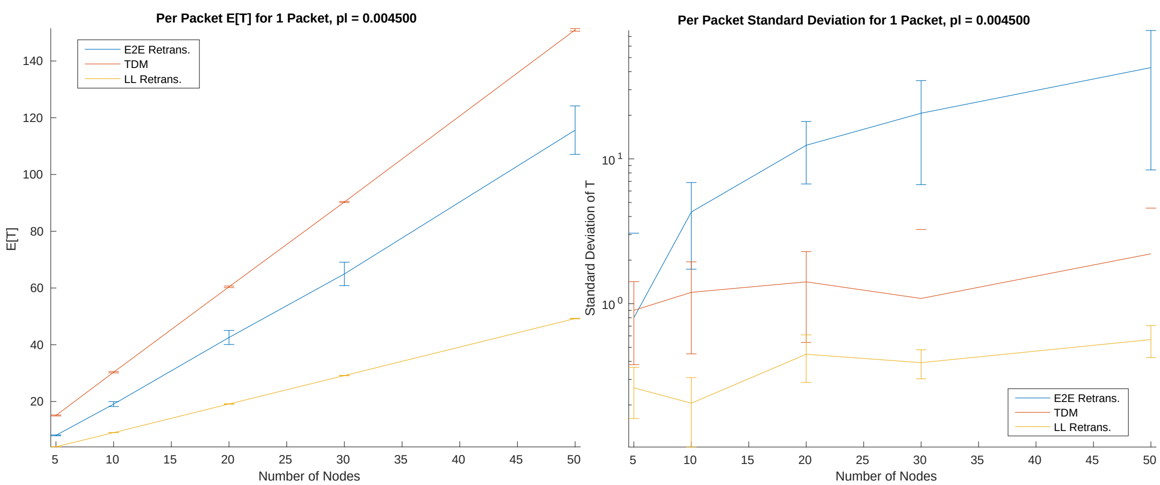

Figure 3-1: Per packet expected completion time and standard deviation versus number of nodes for single packet schemes using end-to-end retransmission, end-to-end coding (with $M = 1$), and link-by-link retransmission. The loss probability on a single link is 0.0045.

The expected completion time and standard deviation versus number of nodes for single packet protocols is shown in figure 3-1, and versus probability of success on a single link in figure 3-2. When end-to-end coding TDM is used, a single packet is repeated until it is acknowledged. The greater completion time for TDM over end-to-end retransmission is due to protocol design. If the acknowledgment is not lost,

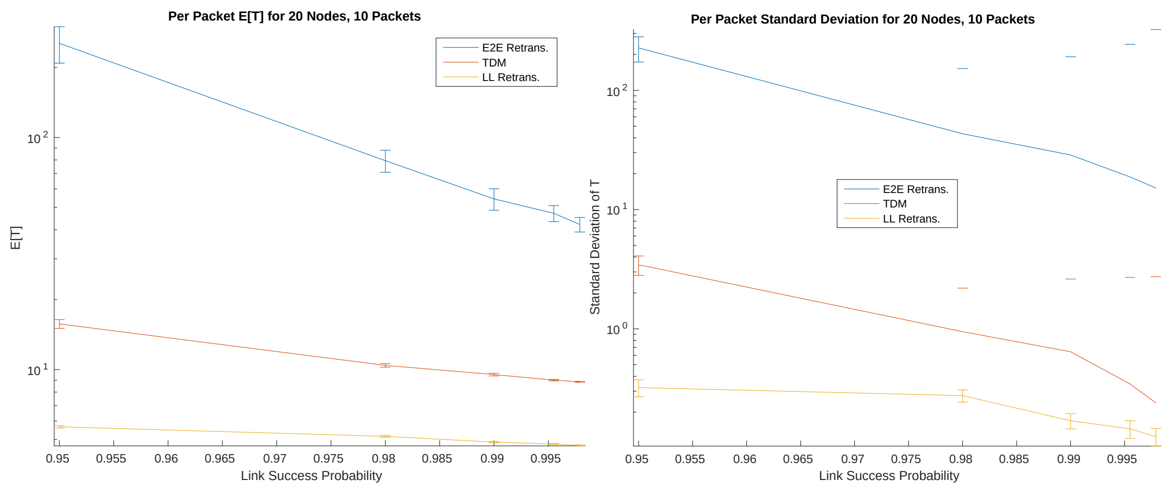

Figure 3-2: Per packet expected completion time and standard deviation versus probability of success on a single link for end-to-end coding using TDM, FDM, and full duplex channels. There are 20 nodes.

the time it takes to be sent is N for end-to-end retransmission and $2N$ for TDM. Although TDM has a much lower variance, since the packet and acknowledgment are not lost most of the time (65% for 50 nodes, 92% for 10), the added overhead of using TDM outweighs the expected cost of retransmission.

3.3.2 End-to-End Coding

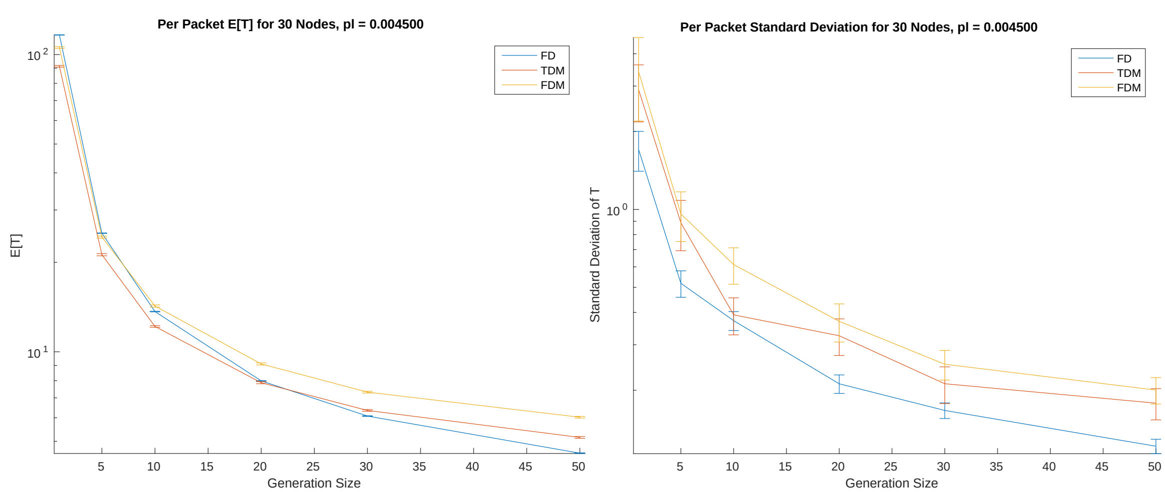

Figure 3-3: Per packet expected completion time and standard deviation versus generation size for end-to-end coding using TDM, FDM, and full duplex channels. There are 30 nodes, and the loss probability on a single link is 0.0045.

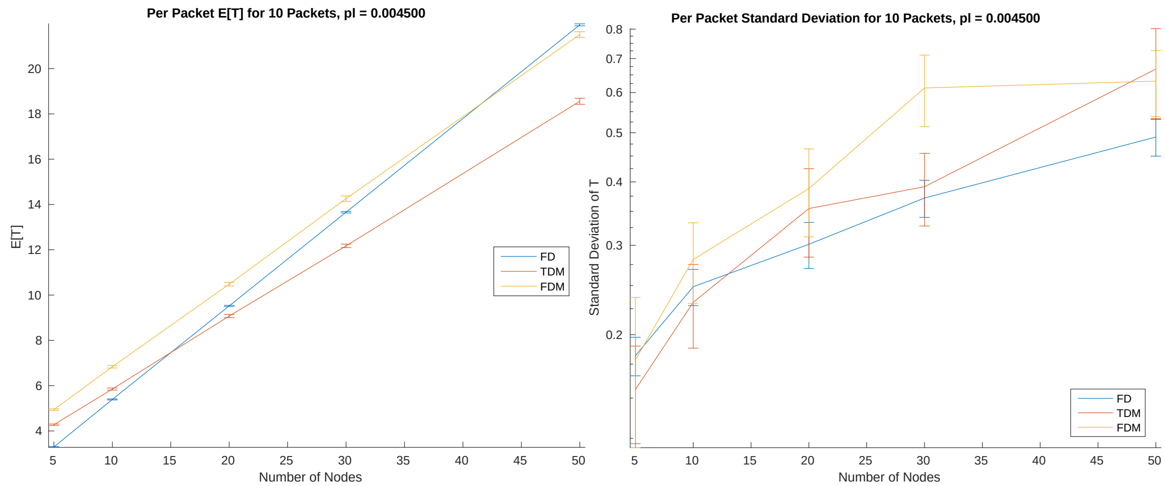

Figure 3-4: Per packet expected completion time and standard deviation versus number of nodes for end-to-end coding using TDM, FDM, and full duplex channels. The generation size is 10 packets, and the loss probability on a single link is 0.0045.

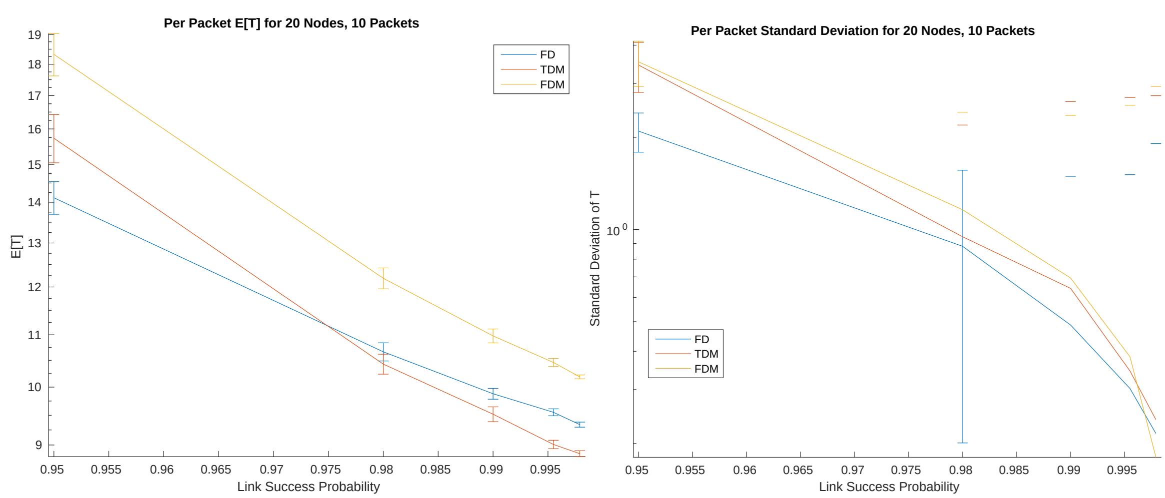

Figure 3-5: Per packet expected completion time and standard deviation versus probability of success on a single link for end-to-end coding using TDM, FDM, and full duplex channels. The generation size is 10 packets, and there are 20 nodes.

Per packet expected completion time and standard deviation versus generation size for end-to-end coding schemes is shown in figure 3-3. Per packet expected completion time and standard deviation versus number of nodes is shown in figure 3-4, and versus probability of success in figure 3-5. The FDM simulation used a value for α of 1.74.

The expected completion time and standard deviation of TDM is strictly better than that of FDM in this case. In order to see a benefit from using FDM, the reduction in noise due to lower bandwidth must outweigh the decrease in rate when using a lower bandwidth. The critical value is $\alpha = 1.5$.

The results for TDM and a full-duplex channel are somewhat surprising, given that the full-duplex channel is strictly better. However, they are due entirely to the construction of the protocols. Assuming the final acknowledgment is not lost, the time it takes to be sent is $2N$ for TDM and $3N$ for the full-duplex protocol. When the generation size is small and the number of nodes is large, this difference becomes significant. When the generation size grows, the cost of the final acknowledgment is amortized over a larger number of packets, so becomes less significant.

3.3.3 Multi-Packet Coding Schemes

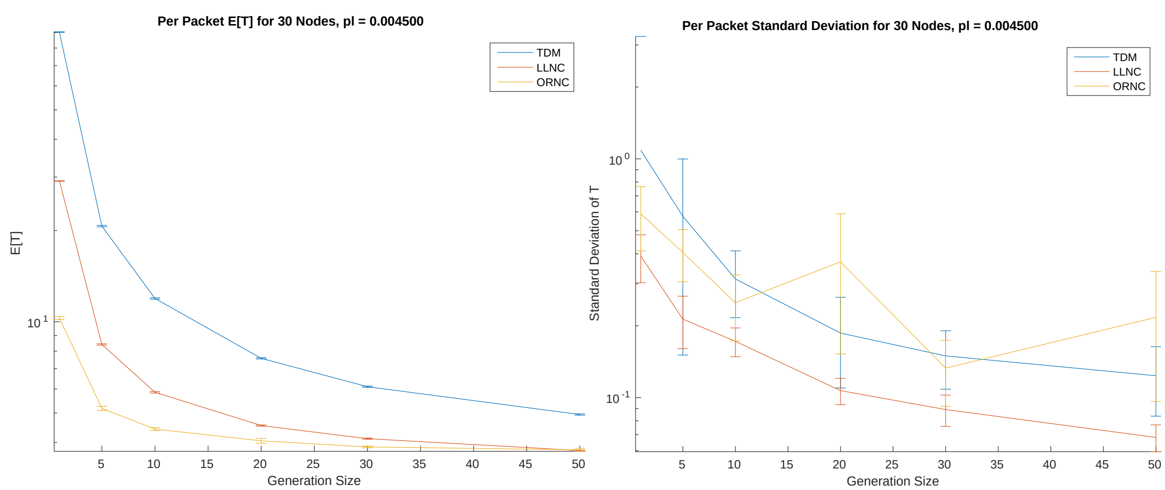

Figure 3-6: Per packet expected completion time and standard deviation versus generation size for end-to-end coding using TDM and network coding without and with opportunistic routing. There are 30 nodes, and the loss probability on a single link is 0.0045. When opportunistic routing is used, the loss probabilities for one, two, and three hops are 0.0045, 0.0065, and 0.0779 respectively.

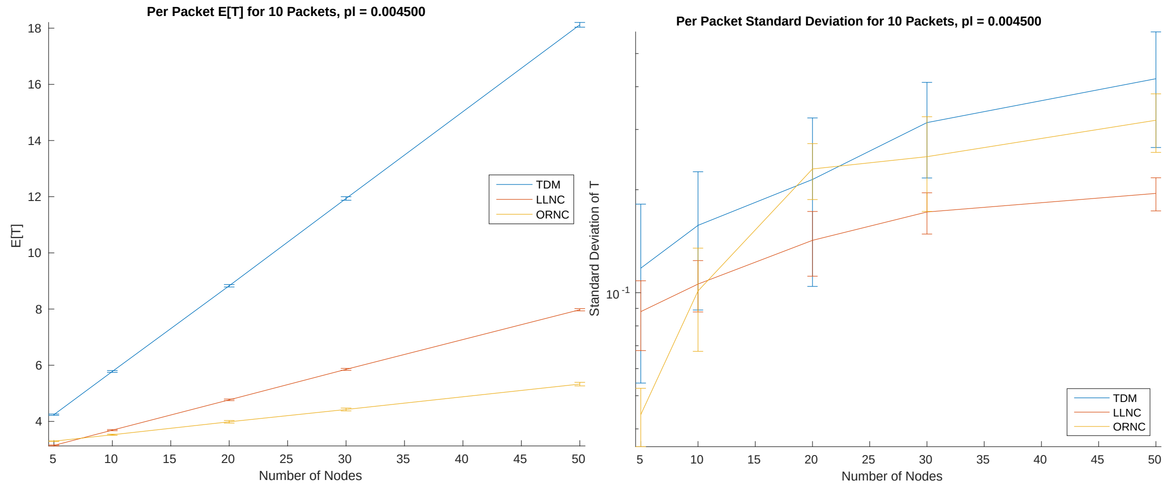

Figure 3-7: Per packet expected completion time and standard deviation versus number of nodes for end-to-end coding using TDM and network coding without and with opportunistic routing. The generation size is 10 packets, and the loss probability on a single link is 0.0045. When opportunistic routing is used, the loss probabilities for one, two, and three hops are 0.0045, 0.0065, and 0.0779 respectively.

Per packet expected completion time and standard deviation versus generation size for multiple packet coded schemes are shown in figure 3-6. Per packet expected completion time and standard deviation versus number of nodes is shown in figure 3-7, and versus probability of success in figure 3-8.

When opportunistic routing is used, the completion time of the network coding protocol is usually lower than when opportunistic routing is not used. The exception is when the network is very short (in this case, when there are five nodes). Since $\delta = 3$, a transmission by any node will interfere with the transmission of any other, thus only one transmission can occur in the network at any time. When $\delta = 1$, this problem does not occur.

With opportunistic routing, the variance of the completion time is usually higher than when opportunistic routing is not used. This is due to the variance introduced when sending the packet a variable number of hops and the corresponding wait time of the protocol. This trend is not seen when $N = 5$, and is due to the same reason that the completion time is higher.

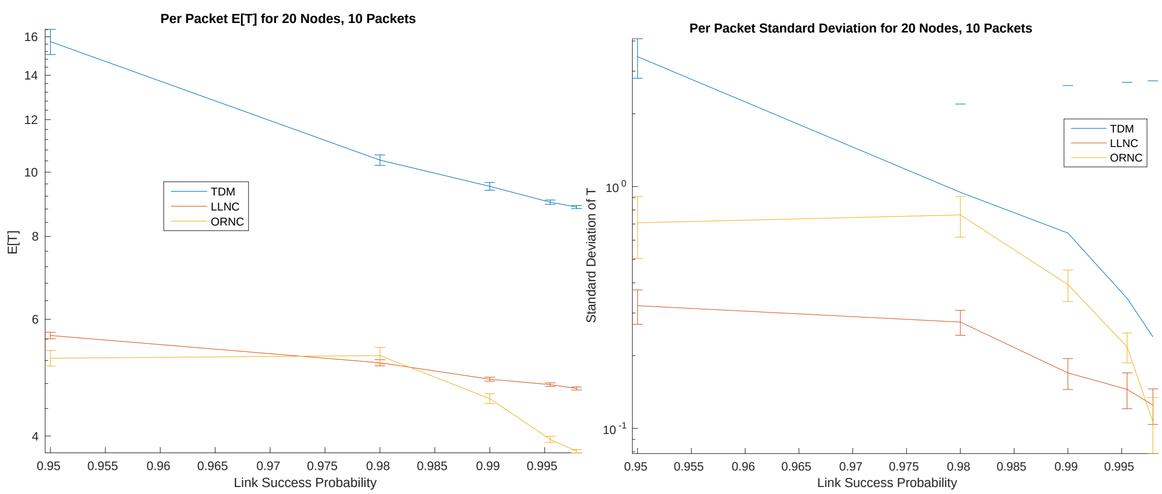

Figure 3-8: Per packet expected completion time and standard deviation versus probability of success on a single link for end-to-end coding using TDM and network coding without and with opportunistic routing. The generation size is 10 packets, and there are 20 nodes.

3.4 Comparison of Simulated and Analytical Results

Comparisons of analytical and simulated results are shown in figures 3-9 through 3-24. For plots of the mean, the error bars mark twice the standard error of the mean in both directions, i.e.,

$$\pm\frac{2\sigma}{\sqrt{n}}$$

where σ is the sample standard deviation and $n = 100$. For plots of the standard deviation, the error bars mark one standard error of the variance.

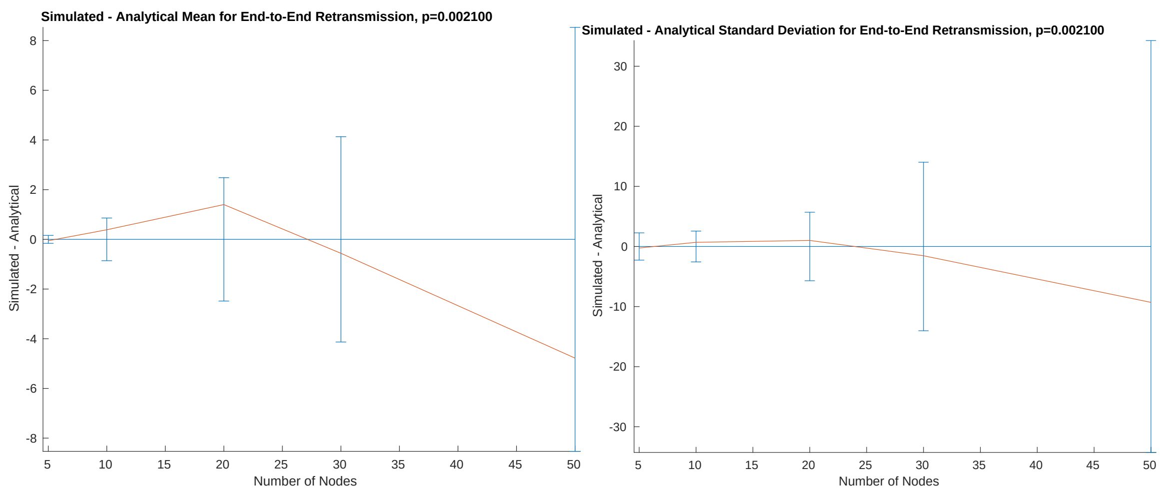

Figure 3-9: Comparison of analytical and simulated mean and standard deviation of completion time for end-to-end retransmission. The loss probability on a single link is 0.0021.

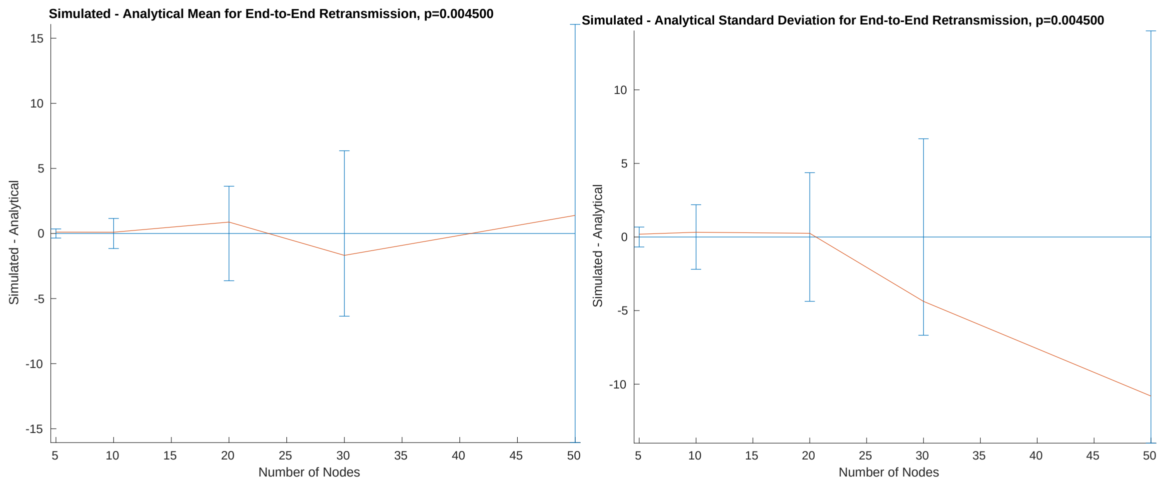

Figure 3-10: Comparison of analytical and simulated mean and standard deviation of completion time for end-to-end retransmission. The loss probability on a single link is 0.0045.

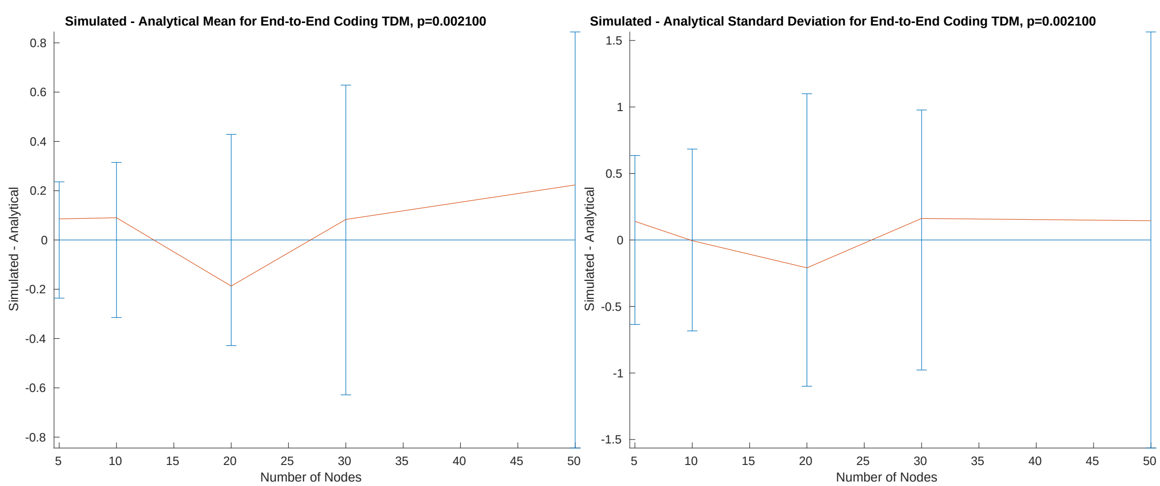

Figure 3-11: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using TDM. There are 10 packets, and the loss probability on a single link is 0.0021.

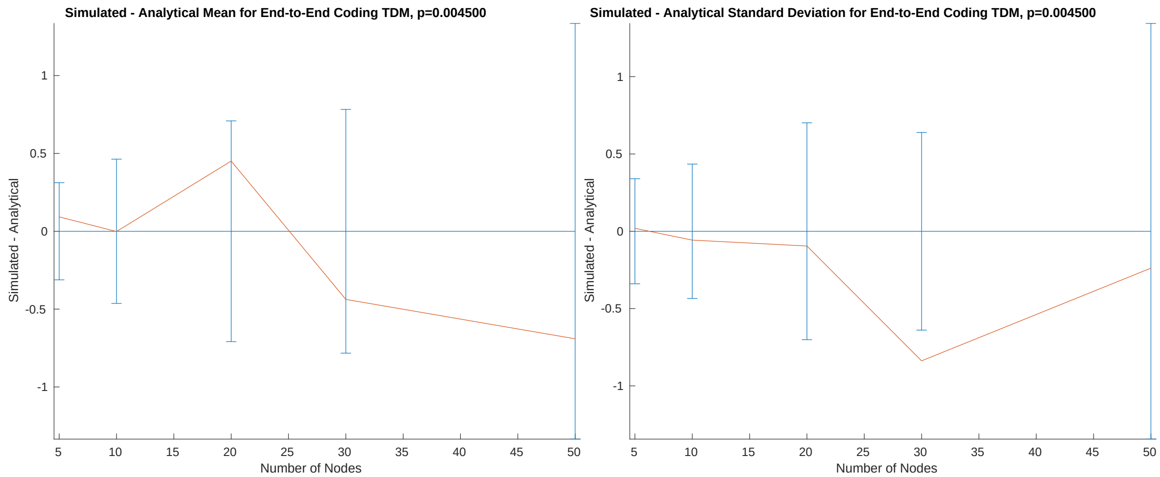

Figure 3-12: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using TDM. There are 10 packets, and the loss probability on a single link is 0.0045.

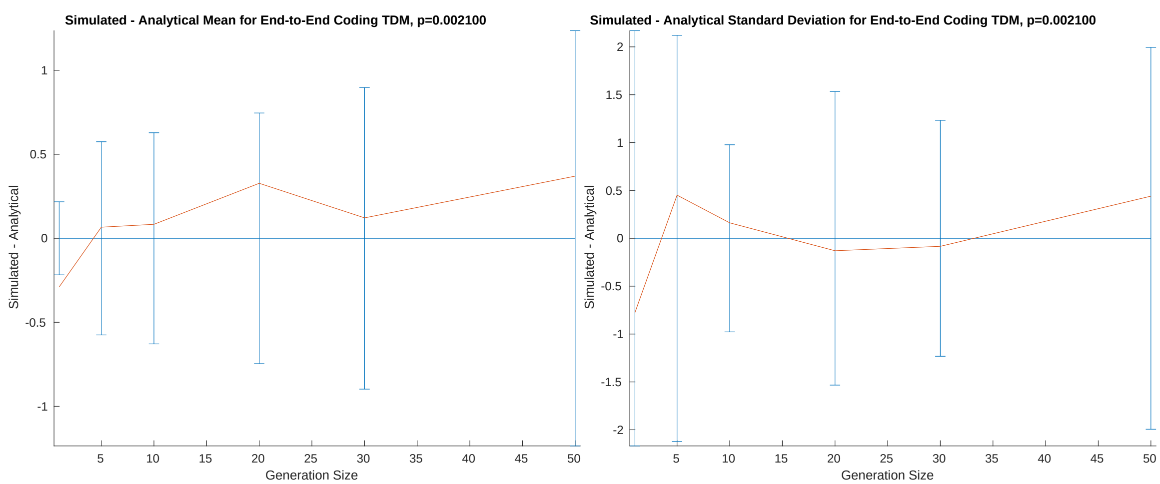

Figure 3-13: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using TDM. There are 30 nodes, and the loss probability on a single link is 0.0021.

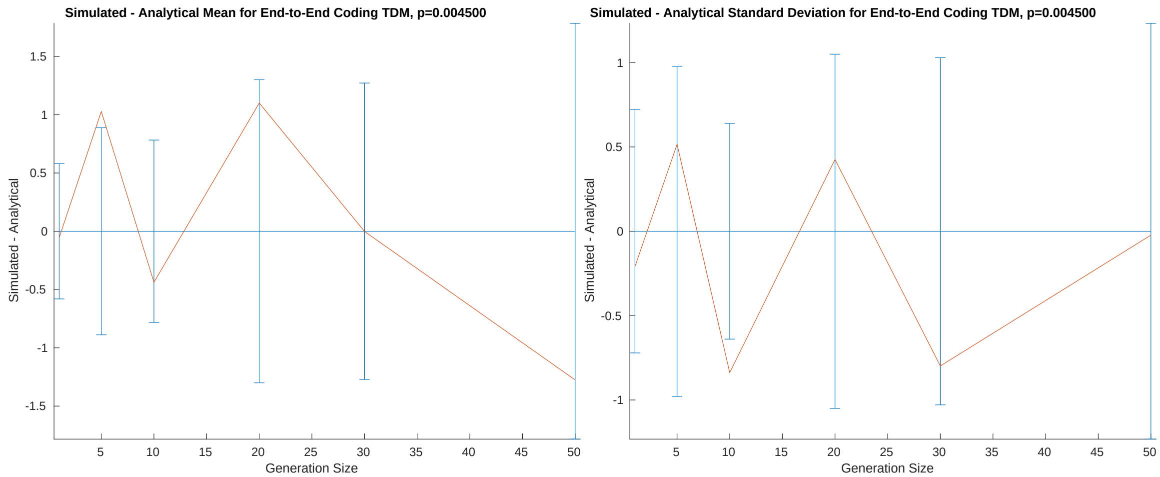

Figure 3-14: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using TDM. There are 30 nodes, and the loss probability on a single link is 0.0045.

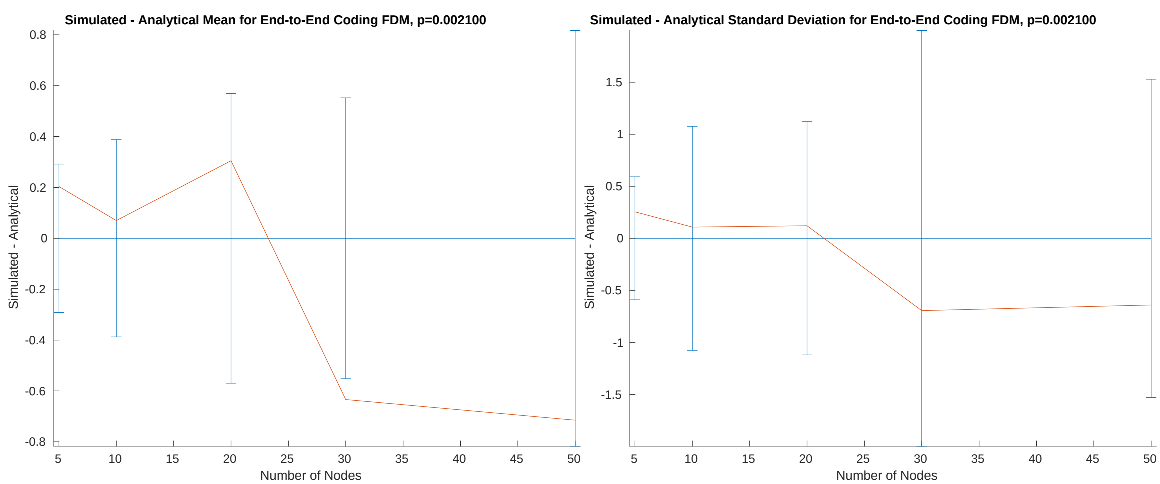

Figure 3-15: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using FDM. There are 10 packets, and the loss probability on a single link is 0.0021.

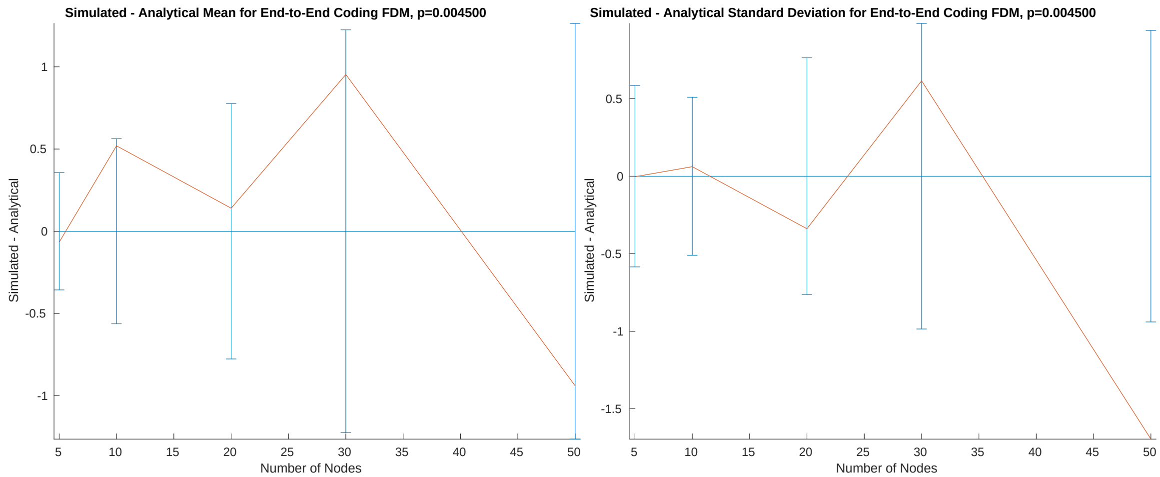

Figure 3-16: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using FDM. There are 10 packets, and the loss probability on a single link is 0.0045.

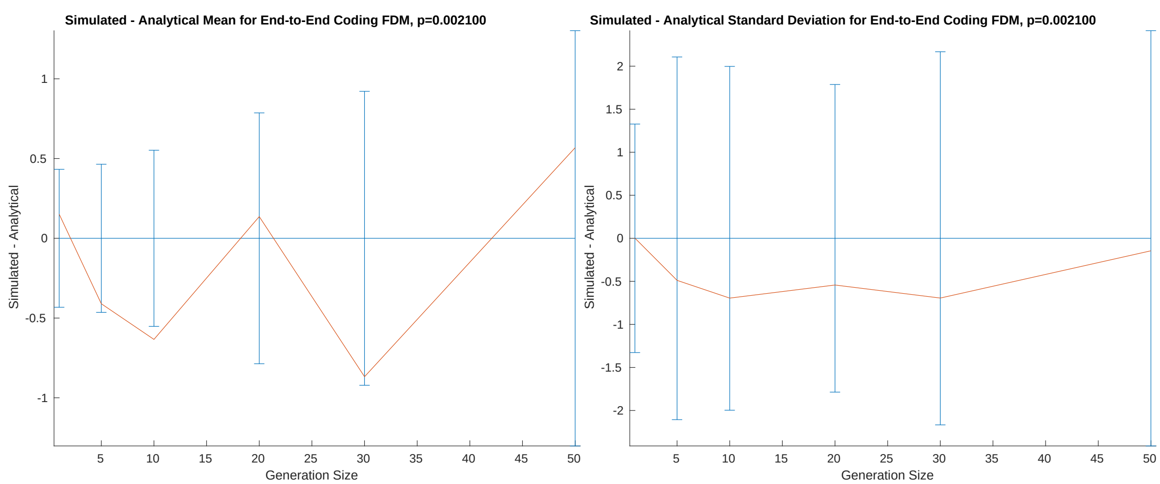

Figure 3-17: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using FDM. There are 30 nodes, and the loss probability on a single link is 0.0021.

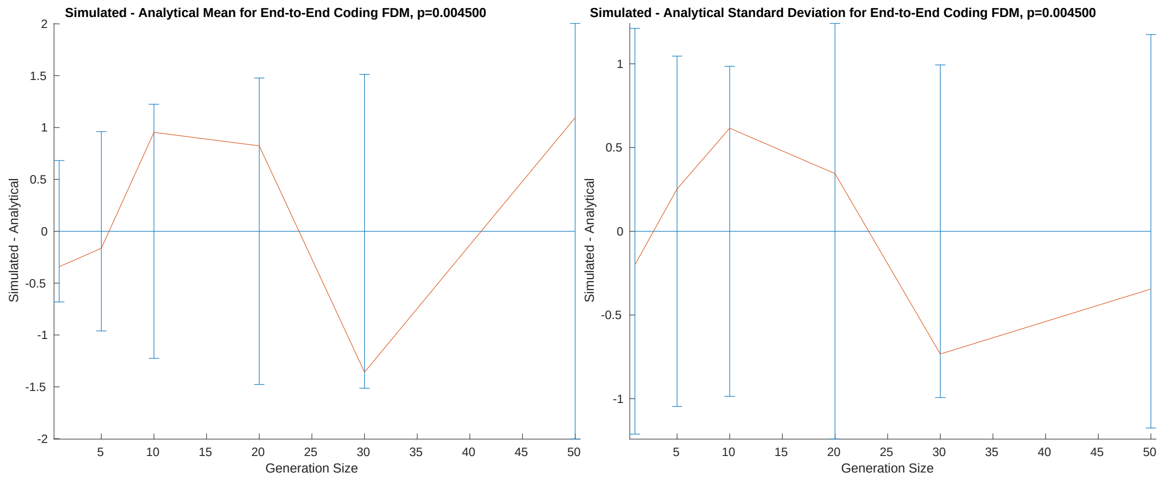

Figure 3-18: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using FDM. There are 30 nodes, and the loss probability on a single link is 0.0045.

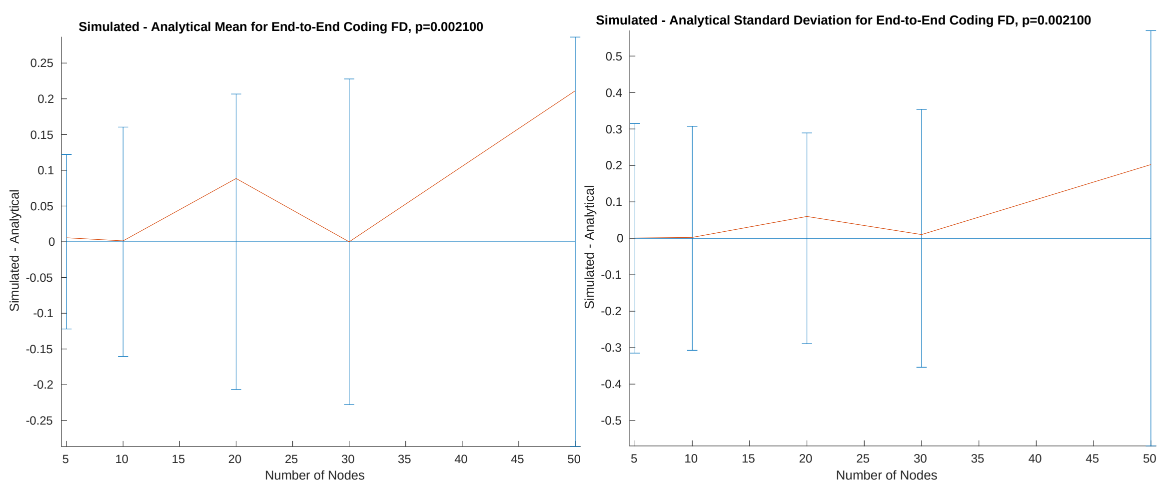

Figure 3-19: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using a full duplex channel. There are 10 packets, and the loss probability on a single link is 0.0021.

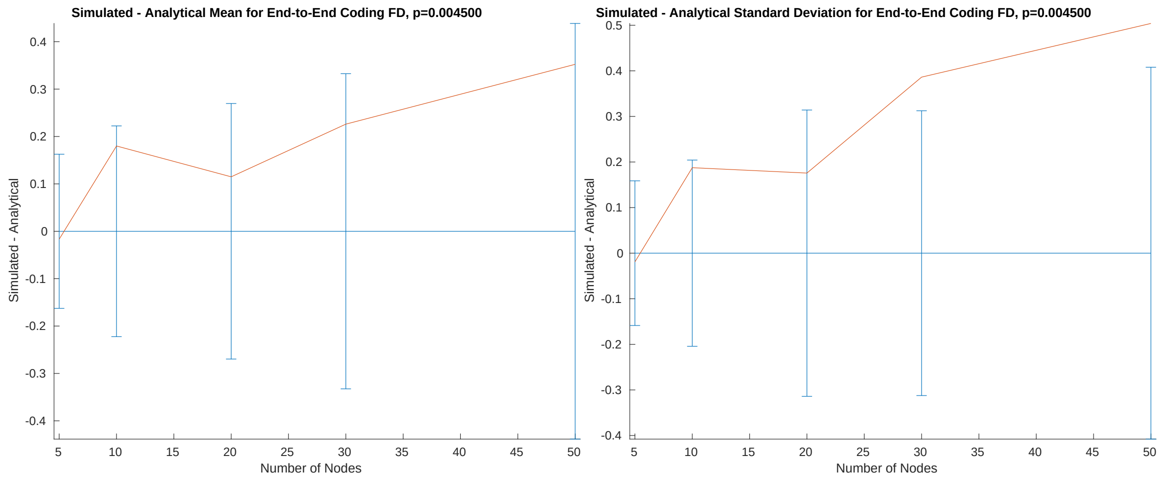

Figure 3-20: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for end-to-end coding using a full duplex channel. There are 10 packets, and the loss probability on a single link is 0.0045.

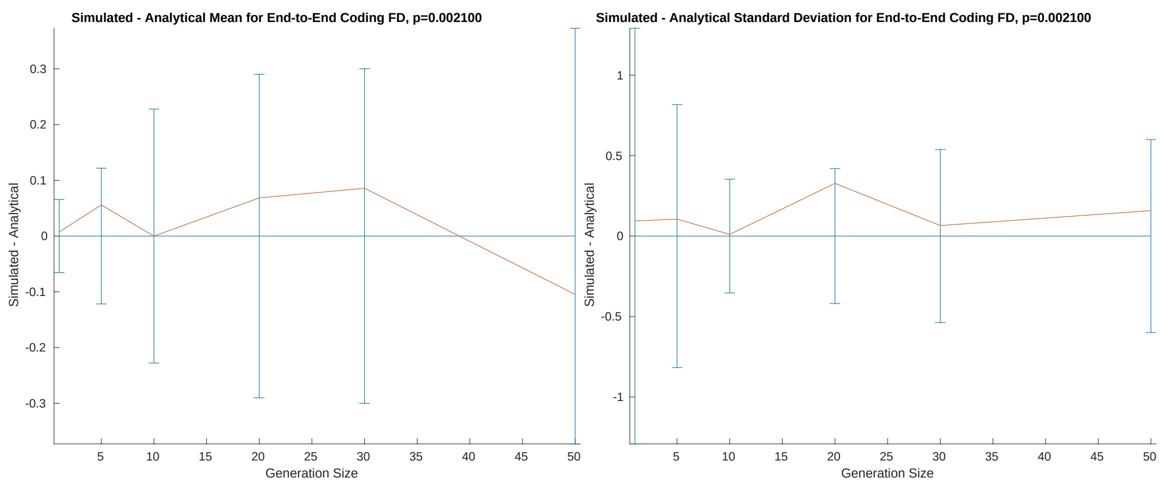

Figure 3-21: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using a full duplex channel. There are 30 nodes, and the loss probability on a single link is 0.0021.

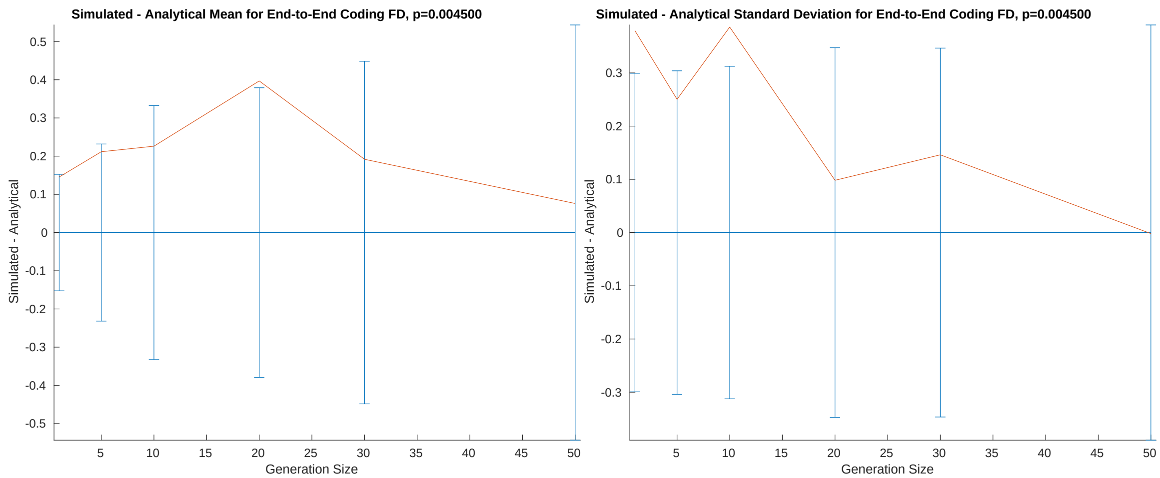

Figure 3-22: Comparison of analytical and simulated mean and standard deviation of completion time against generation size for end-to-end coding using a full duplex channel. There are 30 nodes, and the loss probability on a single link is 0.0045.

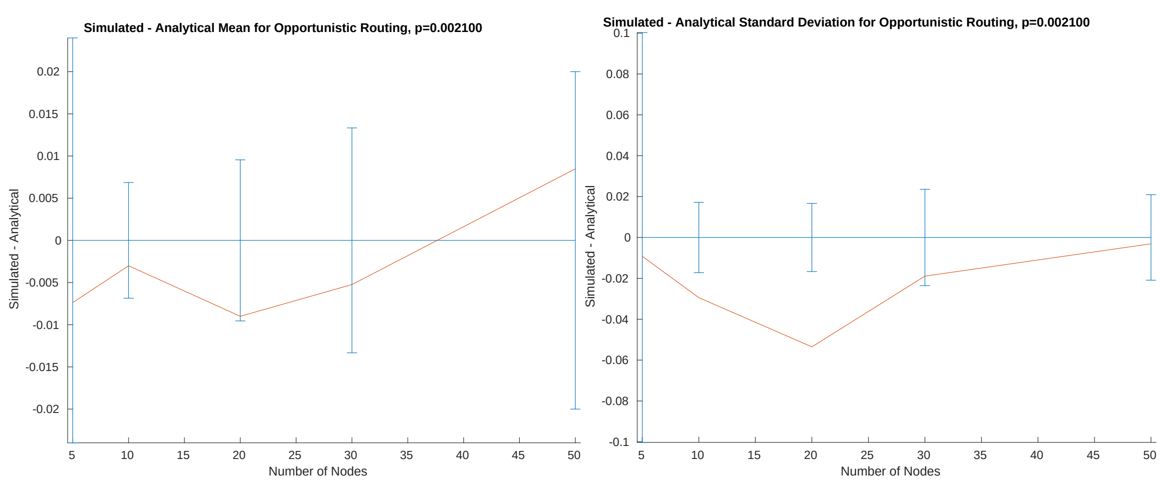

Figure 3-23: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for single packet opportunistic routing. The loss probabilities for one, two, and three hops are 0.0021, 0.0025, and 0.0045 respectively.

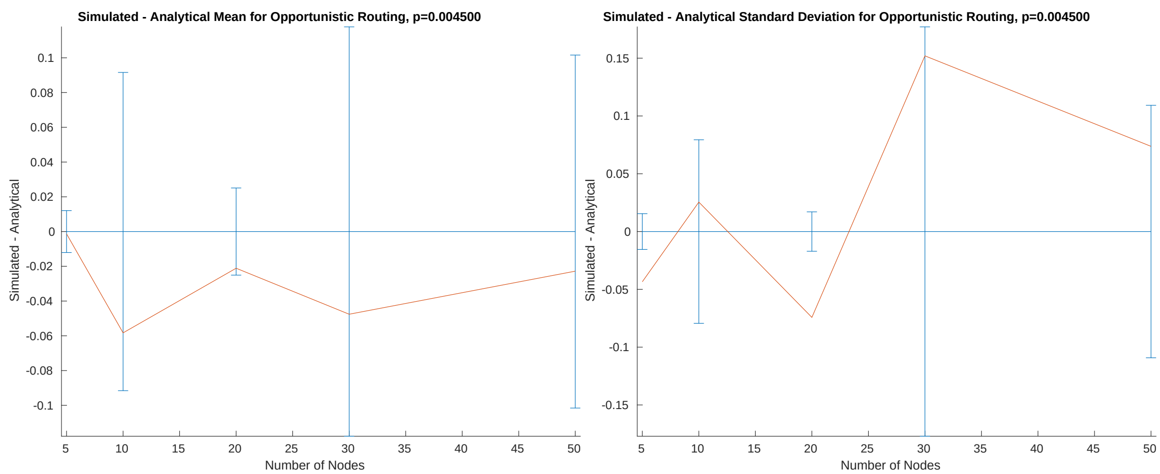

Figure 3-24: Comparison of analytical and simulated mean and standard deviation of completion time against number of nodes for single packet opportunistic routing. The loss probabilities for one, two, and three hops are 0.0045, 0.0065, and 0.0779 respectively.

Chapter 4

Conclusion

This thesis presented several protocols for communication on a wireless linear mesh network, and compared their performance using the mean and variance of the completion time as metrics.

When sending a single packet, the mean completion time, $E[T]$, for TDM is about 50% higher than end-to-end retransmission because of the construction of the duplexing protocol. With TDM, the acknowledgment encounters a delay of one time slot at each node. The standard deviation for TDM is about 10% that of end-to-end retransmission, since many duplicate packets are sent before the acknowledgment is received. $E[T]$ for link-by-link retransmission is roughly 40% that of end-to-end retransmission, and the variance is about 25% that of TDM. Link-by-link retransmission provides a clear benefit since packet losses can be recovered from without having to retransmit from node 1.

When end-to-end coding is used, it is somewhat surprising that full duplexing does not perform much better, if at all, than TDM, given that a full duplex channel is strictly better than a half duplex channel. (Recall that the costs of using a full duplex channel are not considered.) This results from a protocol construction which takes advantage of the full duplex channel: TDM uses a three time slot cycle, and the full duplex protocol uses a four time slot cycle. This allows two transmissions per cycle (as opposed to one for TDM), but leads to a larger delay when sending the acknowledgment. When the generation size is small or the number of nodes is large, the extra time slot of delay is significant. Full duplexing usually shows a smaller standard deviation than TDM, but the mean completion time is only better than TDM when the generation size is large (approximately greater than 30) or the number of nodes is small (approximately less than 15). Within the scope of the simulations in this thesis, at best, full duplexing shows a 20% reduction in $E[T]$. As $M \to \infty$, the reduction in $E[T]$ is only 33%. Although full duplexing provides a benefit in some cases, in most cases considered here, the limiting factor is the broadcast nature of the channel and packet collisions, not the lack of full duplexing. Furthermore, the costs may not be justified given the limited benefit, especially when compared to link-by-link retransmission with network coding.

Also somewhat surprising, is that FDM performs worse than TDM. Again, the limiting factor is the broadcast nature of the channel leading to packet collisions. Since the models are nearly identical, differing only in the cost vector, $\vec{\xi}$, the factor which leads to the difference in results is the α parameter—the increase in packet duration due to the halving of transmit bandwidth—used in FDM. The value of α accounts for the decreased transmission rate, but also the decreased noise power due to the reduction in transmit bandwidth. The value for α was set to 1.74 for the simulations. The critical value which produces results identical to TDM is $\alpha = 1.5$, which may be difficult to achieve when the channel bandwidth is the limiting factor in the maximum achievable rate of communication. If, however, FDM can be achieved with little or no reduction in the transmission rate, both the mean and standard deviation of the completion time will be lower than with TDM. Another advantage of FDM, is that traffic is not necessarily prioritized in one direction. When using TDM, a packet traveling upstream will encounter a delay of one time slot at each node because of the duplexing scheme. With FDM, this delay is not present because multi-packet reception is possible.

In general, when end-to-end coding is used, TDM usually performs similarly or better than both FDM and full duplexing. Although FDM and full duplexing have limited benefits in some cases, the added cost may not be justified.

Link-by-link retransmission with network coding, both with and without OR, has a lower mean and standard deviation of the completion time than end-to-end coding using TDM, FDM, or full duplexing. When OR is not used, the mean using network coding is far lower than that using end-to-end coding, with the benefit increasing with the number of nodes and probability of error. The standard deviation is also far lower, and increases less drastically as the probability of error or the number of nodes increases. Both of these results are a direct consequence of being able to recode and transmit new dofs in the network. It should also be noted that link-by-link retransmission with network coding requires no special hardware, although it may require extra computational resources—neither frequency division nor full duplexing is used. Thus the cost of implementing FDM or full duplexing may not be justified when link-by-link retransmission with network coding can be used with no extra hardware cost.

The importance of the increasing gap between the completion times of link-by-link retransmission with network coding and end-to-end coding should not be understated. Often, the packet erasure channel is provided by an underlying physical layer code which uses forward error correction. Since network coding is more robust to losses than end-to-end coding, the physical layer may be able transmit at a higher rate, increasing the probability of packet loss, while maintaining or improving $E[T]$.

When OR is used, the standard deviation of the completion time is almost always higher than when it is not used. The mean completion time, however, admits mixed results. When the probability of error on a single link is relatively low ($< 1\%$), then OR shows a significant reduction in completion time, and the benefit diminishes as the error probability increases. When the probability of link error is higher, and δ is correspondingly reduced, the mean completion time is on par with link-by-link retransmission without OR. The benefit of OR is largely affected by the time required to detect a packet, or equivalently, T_0, which is the main source of overhead in the protocol. The choice of T_0 must also account for any randomness in propagation delay and packet detection, possibly increasing the overhead of OR. If $T_0 \ll d$ (where d is the packet duration), then the time required to determine the forwarding node is negligible and the reduction in $E[T]$ is more pronounced than when OR is not used.

It was noted during simulation involving OR, that when δ was increased by one and the probability of packet loss to the farthest node was high (>0.45), that the expected completion time actually increased. However, one may wish to include more distant nodes because of the possibility of energy savings. Characterization of the energy use/completion time trade-off, as well as the more general problem of determining which nodes to consider for forwarding, remains a topic for future work.

The energy usage of each of these protocols also remains a topic for future work. The use of OR reduces the total number of transmissions needed, especially when δ is high, thus saving energy. However, it is not clear if a similar energy savings can be achieved by reducing the transmit power so a node can only feasibly communicate with its immediate neighbors. This will depend on the channel.

Bibliography

[1] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] C.R. Berger, Shengli Zhou, Yonggang Wen, P. Willett, and K. Pattipati. Optimizing Joint Erasure- and Error-Correction Coding for Wireless Packet Transmissions. *IEEE Transactions on Wireless Communications*, 7(11):4586–4595, November 2008.

[3] Sanjit Biswas and Robert Morris. Opportunistic Routing in Multi-hop Wireless Networks. *SIGCOMM Comput. Commun. Rev.*, 34(1):69–74, January 2004.

[4] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, pages 169–180, New York, NY, USA, 2007. ACM.

[5] T.A. Courtade and R.D. Wesel. Optimal Allocation of Redundancy Between Packet-Level Erasure Coding and Physical-Layer Channel Coding in Fading Channels. *IEEE Transactions on Communications*, 59(8):2101–2109, August 2011.

[6] Daniel Lucani, Frank Fitzek, Muriel Médard, and Milica Stojanovic. Network Coding For Data Dissemination: It Is Not What You Know, But What Your Neighbors Don't Know.

[7] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed. On the Delay and Throughput Gains of Coding in Unreliable Networks.

[8] C. Fragouli, D. Lun, M. Medard, and P. Pakzad. On Feedback for Network Coding. In *41st Annual Conference on Information Sciences and Systems, 2007. CISS '07*, pages 248–252, March 2007.

[9] J. Cloud, D. Leith, and M. Medard. A coded generalization of selective repeat ARQ. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2155–2163, April 2015.

[10] Emrah Kılıç and Pantelimon Stanica. The inverse of banded matrices. *Journal of Computational and Applied Mathematics*, 237(1):126–135, January 2013.

[11] MinJi Kim, Thierry Klein, Emina Soljanin, Joao Barros, and Muriel Medard. Modeling Network Coded TCP: Analysis of Throughput and Energy Cost. *arXiv:1208.3212 [cs]*, August 2012. arXiv: 1208.3212.

[12] D. Koutsonikolas, Chih-Chun Wang, and Y.C. Hu. Efficient Network-Coding-Based Opportunistic Routing Through Cumulative Coded Acknowledgments. *IEEE/ACM Transactions on Networking*, 19(5):1368–1381, October 2011.

[13] Daniel E. Lucani, Muriel Médard, and Milica Stojanovic. Network Coding Schemes for Underwater Networks: The Benefits of Implicit Acknowledgement. In *In Proc. WUWnet '07*, pages 25–32.

[14] D.E. Lucani, M. Medard, and M. Stojanovic. On coding for delay - New approaches based on network coding in networks with large latency. In *Information Theory and Applications Workshop, 2009*, pages 191–200, February 2009.

[15] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter. An Analysis of Finite-Memory Random Linear Coding on Packet Streams. In *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, April 2006.

[16] Desmond S. Lun, Muriel Medard, Ralf Koetter, and Michelle Effros. On Coding for Reliable Communication over Packet Networks. *Physical Communication*, 1(1):3–20, March 2008. arXiv: cs/0510070.

[17] D.S. Lun, M. Medard, and R. Koetter. Network Coding for Efficient Wireless Unicast. In *2006 International Zurich Seminar on Communications*, pages 74–77, 2006.

[18] Neil A. Weiss. *A Course In Probability*. Pearson Addison-Wesley, Boston.

[19] A. ParandehGheibi, M. Medard, A. Ozdaglar, and S. Shakkottai. Avoiding Interruptions: A QoE Reliability Function for Streaming Media Applications. *IEEE Journal on Selected Areas in Communications*, 29(5):1064–1074, May 2011.

[20] Robert G. Bartle and Donald R. Sherbert. *Introduction to Real Analysis*. Wiley, Hoboken, NJ, 4th edition, 2011.

[21] M. E. Rusli, R. Harris, and A. Punchihewa. Performance analysis of implicit acknowledgment coordination scheme for opportunistic routing in wireless sensor networks. In *2012 International Symposium on Telecommunication Technologies (ISTT)*, pages 131–136, November 2012.

[22] B. Siqueira, D. Passos, D.C. Muchaluat-Saade, and C. Albuquerque. LIBR: ID-based routing for linear Wireless Mesh Networks. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 461–466, January 2015.

[23] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros. Network Coding Meets TCP: Theory and Implementation. *Proceedings of the IEEE*, 99(3):490–512, March 2011.

[24] S. Teerapittayanon, K. Fouli, M. Medard, M.-J. Montpetit, X. Shi, I. Seskar, and A. Gosain. Network Coding as a WiMAX Link Reliability Mechanism. *arXiv:1208.4766 [cs]*, August 2012. arXiv: 1208.4766.

[25] Surat Teerapittayanon, Kerim Fouli, Muriel Médard, Marie-José Montpetit, Xiaomeng Shi, Ivan Seskar, and Abhimanyu Gosain. Network Coding as a WiMAX Link Reliability Mechanism: An Experimental Demonstration. In Boris Bellalta, Alexey Vinel, Magnus Jonsson, Jaume Barcelo, Roman Maslennikov, Periklis Chatzimisios, and David Malone, editors, *Multiple Access Communications*, number 7642 in Lecture Notes in Computer Science, pages 75–78. Springer Berlin Heidelberg, November 2012. DOI: 10.1007/978-3-642-34976-8_7.