Transport Layer Protocol Design over Flow-Switched Data Networks

Henna Priscilla Huang
B.S. Electrical Engineering
University of California, Davis (2009)

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 2012

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2012

Certified by .....
Vincent W.S. Chan
Joan and Irwin Jacobs Professor of Electrical Engineering and Computer Science
Thesis Supervisor

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

Transport Layer Protocol Design over Flow-Switched Data Networks

by

Henna Priscilla Huang

Submitted to the Department of Electrical Engineering and Computer Science on May 23, 2012 in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

Abstract

In this work, we explore transport layer protocol design for an optical flow-switched network. The objective of the protocol design is to guarantee the reliable delivery of data files over an all-optical end-to-end flow-switched network which is modeled as a burst-error channel. We observe that Transport Control Protocol (TCP) is not best suited for Optical Flow-Switching (OFS). Specifically, flow control and fair resource allocation through windowing in TCP are unnecessary in an OFS network. Moreover TCP has poor throughput and delay performance at high transfer rates due to window flow control and window closing with missing or dropped packets. In OFS, flows are scheduled and congestion control is performed by a scheduling algorithm. Thus, we focus on defining a more efficient transport protocol for optical flow-switched networks that is neither a modification of TCP nor derived from TCP.

The main contribution of this work is to optimize the throughput and delay performance of OFS using file segmentation and reassembly, forward error-correction (FEC), and frame retransmission. We analyze the throughput and delay performance of four example transport layer protocols: the Simple Transport Protocol (STP), the Simple Transport Protocol with Interleaving (STPI), the Transport Protocol with Framing (TPF) and the Transport Protocol with Framing and Interleaving (TPFI).

First, we show that a transport layer protocol without file segmentation and without interleaving and FEC (STP) results in poor throughput and delay performance and is not well suited for OFS. Instead, we found that interleaving across a large file (STPI) results in the best theoretical delay performance, though the large code lengths and interleaver sizes in this scheme will be hard to implement. Also, in the unlikely case that a file experiences an uncorrectable error, STPI requires extra network resources equal to that of an entire transaction for file retransmission and adds to the delay of the transaction significantly.

For the above reason, we propose the segmentation of a file into large frames combined with FEC, interleaving, and retransmission of erroneous frames (TPFI) as the protocol of choice for an OFS network. In TPFI, interleaving combined with FEC and frame retransmission allows a file to be segmented into large frames (>100 Mbits). In addition, TPFI also allows for fewer processing and file segmentation and reassembly overhead compared with a transport layer protocol that does not include interleaving and FEC (TPF).

Thesis Supervisor: Vincent W.S. Chan

Title: Joan and Irwin Jacobs Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to thank my research advisor, Professor Vincent Chan, without whose guidance, this thesis would not be possible. Professor Chan's dedication to his students, intuition, and generosity continue to be a great inspiration to me. Under his mentorship, I am not only gaining a firm theoretical foundation but also learning to think creatively and communicate effectively.

I would like to thank my high school computer science teacher, Mr. Simon, for introducing me to the field of engineering. I would also like to thank my professors at UC Davis: Professor Bevan Baas and Professor Kent Wilken for encouraging me to pursue my graduate studies.

A special thanks to Matt and Lei, for all of their help and support. I would like to thank Donna for her care for all the students in our group. Thank you to my fellow graduate students and friends at MIT: Tong, Shane, Andrew, Katherine, Manishika, Hoho, Joe, Rui, Niv and Ethan. Thank you to my housemates: Jean and Oksana. Thank you to all my friends from church for making me feel at home in Cambridge: Cynthia, Thomas, Jim, Judy, Sarah, Yanqing, Hannah, David, Kevin, Ray, Randall, Sarah, Ming, Josephine, Rebecca, Stephen, Stanley, Matt, Spencer, Kelly, Amy, Kate, Allison, and Maria.

Finally, I would like to thank my family: mom, dad, Sharon, Ruth, Sophie, and Daniel. I am thankful for their love and support all these years.

Table of Contents

List of Figures

List of Tables

List of Notation

q	Random bit error probability during a non-outage period
β_1	Rate of outages
β_2	Average rate at which the link returns to a non-outage state
$\overline{w}$	Expected number of flow wavelengths per fiber
$E[Y]$	Expected session duration per flow
R	Transmission data rate in bits per second
Y	Session duration
J	Interleaver depth
t_{start}	Flow start time
$A[\,]$	User A's transmitter availability
$B[\,]$	User B's receiver availability
R_A	User A's transmitter rate
R_B	User B's receiver rate
w	Assigned session wavelength
H_{CRC}	CRC overhead required to detect both random and burst errors in OFS
μ	Total message length (in bits) over which an error-correction code is applied
N	Length (in bits) of a code
U	Rate (in bits per transmission) of a (μ, N) code
C_q	Information capacity of a BSC with parameter q
K_q	The redundancy added to combat random errors over a BSC with parameter q
$Q(t)$	Channel crossover probability at time t
S_t	The current channel state at time t
ξ	Crossover probability of interleaved channel modeled as a BSC
C_ξ	Capacity of the interleaved channel with parameter ξ
K_ξ	The redundancy added to combat random errors over a BSC with parameter ξ
H_{FEC}	FEC overhead in bits
H_P	The length of the preamble
o	Probability that the preamble is found within a frame
H_O	Length of the OFS header in bits
γ	Total overhead of an OFS message.
λ	Rate of OFS session request arrivals
w_m	Number of wavelength channels available for flow traffic from a source MAN to a destination MAN
ρ	Network load
y_a	Minimum OFS session duration
y_b	Maximum OFS session duration
τ_s	Expected setup delay
ρ_{min}	Network load that results in a setup delay greater equal to the expected session duration
H	Sum of the OFS header length and CRC overhead
ψ	STP probability of an erroneous file
ϕ_m	STP probability of m failed transmissions
F	Transmission File size in bits

W	STP transmission size per session
ζ	Residual BER after FEC is applied to random errors
ϱ	Probability of an erroneous file due to random errors
v	Required number of transmissions for a file to be received without error at the destination
$\mathcal{G}$	STP expected throughput
$\mathcal{Y}$	STP expected delay of a session
$\mathcal{T}$	STP total expected delay
ψ_I	STPI probability of a decoding error
$\mathcal{G}_I$	STPI expected throughput
$\mathcal{Y}_I$	STPI expected delay of a session
$\mathcal{T}_I$	STPI total expected delay
p	TPF probability of an erroneous frame
θ	TPF probability of a failed initial transmission
D	Number of message bits per frame
L	TPF total number of bits per frame
n	Number of frames that a file is segmented into
Δ	Maximum fraction of retransmission frames to the total number of frames in a file per session
δ	Additional time per session allowed for frame retransmissions
τ_p	Round trip propagation delay
η	TPF expected throughput
D^*	TPF message length at the optimal frame length
L^*	TPF optimal frame length
η^*	TPF maximum expected throughput
ε	Relaxation away from the optimum throughput or delay
D_ε	TPF message length at the practical frame length
η_ε	TPF expected throughput ε away from the optimal expected throughput
L_ε	TPF practical frame length
T	TPF total delay
$E[T]$	TPF expected total delay to send a file
$p_{k\backslash x}$	probability that k outstanding frames remain after session termination out of x initial frames sent
$E[T^*]$	TPF optimal expected delay
Δ^*	TPF fraction of retransmission frames to the total number of frames in a file per session that results in the minimum total delay
$E[X_1]$	Expected number of outstanding erroneous frames after the initial session
$E[T_\varepsilon]$	TPF expected delay ε away from the optimal expected delay
p_I	TPFI probability of an erroneous frame
θ_I	TPFI probability of a failed initial transmission
L_I	TPFI total number of bits per frame
η_I	TPFI expected throughput
${D_I}^*$	TPFI message length at the optimal frame length
${L_I}^*$	TPFI optimal frame length
${\eta_I}^*$	TPFI maximum expected throughput
η_ε^I	TPFI expected throughput ε away from the optimal expected throughput

D_{ε}^{I}	TPFI message length at the practical frame length
L_{ε}^{I}	TPFI practical frame length
T_I	TPFI total delay
$E[T_I]$	TPFI expected total delay
$E[T_I{}^*]$	TPFI optimum expected delay
$\Delta_I{}^*$	TPFI fraction of retransmission frames to the total number of frames in a file per session that results in the minimum total delay
$E[T_{\varepsilon}^{I}]$	TPFI expected delay ε away from the optimal expected delay
p_{Γ}	TPFI probability of an erroneous frame (FEC corrects for Γ burst errors)
θ_{Γ}	TPFI probability of a failed initial transmission (FEC corrects for Γ burst errors)
η_{Γ}	TPFI expected throughput (FEC corrects for Γ burst errors)
η_{Γ}^{*}	TPFI optimal throughput (FEC corrects for Γ burst errors)
$D_{\Gamma}{}^{*}$	TPFI message length at the optimal frame length (FEC corrects for Γ burst errors)
$L_{\Gamma}{}^{*}$	TPFI optimal frame length (FEC corrects for Γ burst errors)
$\eta_{\varepsilon}^{\Gamma}$	TPFI expected throughput ε away from the optimal expected throughput (FEC corrects for Γ burst errors)
$L_{\varepsilon}{}^{\Gamma}$	TPFI practical frame length (FEC corrects for Γ burst errors)
T_{Γ}	TPFI total delay (FEC corrects for Γ burst errors)
$E[T_{\Gamma}]$	TPFI expected total delay (FEC corrects for Γ burst errors)
$E[T_{\varepsilon}^{\Gamma}]$	TPFI delay ε away from the optimal expected delay (FEC corrects for Γ burst errors)

Chapter 1

1 Introduction

In this thesis, we consider flow-switched networks as an enabler for large file transfers with high data rate optical fiber transmission. Optical flow switching (OFS) has promise in reducing cost and lower energy consumption for large data file transmissions [1] [2]. Current forecasts of Internet data show an exponential growth in traffic [3]. High data rate applications such as High-Definition (HD) video streaming, 3-D imaging, cloud computing, and data center transmissions will result in increased data rate demand per user. Internet video currently accounts for over one-third of all consumer Internet traffic and 3-D and HD video is projected to account for 46 percent of consumer Internet video traffic by 2014 [3] [4].

As the demand for large volume data transfer continues to grow in emerging commercial and business applications, optical flow-switched networks can offer reduced cost per bit compared to traditional packet and circuit-switched network architectures. However, the dynamic nature of flow-switched networks introduces new challenges, such as transient power excursions resulting in burst errors. In our work, we address the need to design and analyze a new transport layer protocol for flow-switched networks. We show that traditional transport protocols suffer in both throughput and delay performance in large delay-bandwidth product networks and bursty loss environments. The goal of our transport layer design is to deal with random and "on-off" or burst-error channels for large file transfers.

1.1 Traditional and Flow-Switched Architectures

As networks continue to scale with increasing data demands, legacy data networks face challenges to increase throughput and at the same time reduce both cost and energy consumption. In Internet Protocol (IP) packet-switched networks, users do not reserve network resources before data transmission. Instead, users contend for network resources by sending packets into the network with no prior coordination and use a transport layer protocol such as Transport Control Protocol (TCP) to throttle the transmission rate. Advantages of the IP architecture include low management overhead

associated with sharing network resources. However, large data files are segmented into many small packets where packets are usually switched independently using electronic packet header processing and switching across an electronic fabric at the router. This turns out to be throughput efficient but not cost and energy efficient for large file transfer [5] [6].

Packet switching is an unscheduled service. Other examples of unscheduled architectures include optical burst-switching and the ALOHA multiple access network [7]. In burst switching, large transactions use random access to gain access to the network, resulting in low utilization and waste of precious network resources due to collisions. In the event of many collisions, network resources will spend a large fraction of time with retransmitted data. The pure form of OBS uses a random access scheme similar to ALOHA which has the merit of being simple and requiring little coordination, however, the significant amount of retransmissions increases energy consumption.

Scheduled flow architectures allow for high network utilization in exchange for more complex scheduling overhead. In scheduled architectures, users transmit data according to a schedule that is computed on-line. Users either take advantage of a scheduler or a path reservation mechanism to allocate network resources. For each transaction, users have dedicated resources, and data is sent through the same path in the network for a prearranged duration [8]. In traditional long duration (not usually per-session) circuit-switched networks, however, reserved resources are wasted when users are idle and do not have data to send. In circuit-switched and time-division multiple access (TDMA) networks, network topology and path computation is a slow process. Therefore, once a connection is reserved, the network topology changes on times scales of hours, days, months, and even years.

An OFS network is a per-transaction dynamically scheduled optical flow-switched network. Similar to traditional circuit-switched networks, users reserve resources in the form of sessions in OFS. Users begin and end data transmission according to the time allocated by a scheduler. Data paths are established before data transmission and network resources are released once sessions terminate. Sessions are not reoccurring; users must request a new session for each data transaction. OFS allows for high network utilization in addition to data rate guarantees.

Optical flow switching has the promise to dramatically reducing the cost per bit compared with packet architectures [2]. While the majority of this thesis focuses on developing the transport layer

protocol for OFS, flow-switched architecture are not limited to the optical data plane. With further research, flow-switching can be used over satellite and wireless communication channels to enable high data rate transmission.

1.2 Shortcoming of Traditional Transport Protocols

The transport layer is an end user peer process. The objectives of a transport layer protocol are to provide congestion control, bandwidth matching between transmitter and receiver, and the reliable delivery of data. Currently, TCP is the dominant transport layer protocol in practice. TCP is most often associated with the IP network architecture in the form of the TCP/IP protocol stack. Due to the massive scale of the Internet, the bulk of transport layer protocol work has been to modify TCP. However, the need for a disruptive change in transport protocols becomes apparent as data rates and data volume per transaction continues to increase. Adopting incremental changes to TCP alone is insufficient to combat the drastic performance degradation when using TCP in high delay-bandwidth product networks.

Although TCP provides reliable delivery of packets, it has serious performance problems generally for large bandwidth-delay product networks and those with high contiguous packet losses. The three defining characteristics of TCP are Additive Increase Multiplicative Decrease (AIMD), Slow Start, and Fast Retransmit/Fast Recovery. AIMD and Slow Start use packet loss as an indication of congestion and react to loss due to errors by decreasing a user's congestion window size to one packet. In addition, Fast Retransmit/Fast Recovery is sensitive to out of order (OOO) packets, a common event in networks that experience link outages. In the event of an outage, TCP misinterprets duplicate acknowledgements (ACKs) as an indication of congestion. By assuming congestion in the network, TCP's window closing and timeout mechanisms adversely reduce a user's data rate and throughput performance. A severe outage can cause a TCP timeout and close a user's window to one packet in flight [9]. Packets lost due to an outage as opposed to those lost due to network congestion should trigger different reactions by the transport layer.

Furthermore, as networks continue to grow in scale and data rate, TCP underutilizes links with high delay-bandwidth products (e.g. satellite and optical links) even without packet losses. In these networks, available data rates can be on the order of gigabits per second, and data traverses thousands

to tens of thousands of kilometers from sender to destination. In TCP, users enter the network through slow start mode to avoid overloading the network. This policy is detrimental to achieving high transmission rates for high delay-bandwidth networks [10]. In theory, depending on a user's round trip time (RTT), it may take hours or even days before a user's congestion window can grow to the maximum window size even if it is uncapped as proposed. A user must wait many RTTs before a significant increase in transmission data rate.

Fairness is another performance area where TCP lacks as a transport layer protocol for flow-switched networks. On a high delay-bandwidth link, TCP provides an unfair rate advantage to a user with a shorter RTT compared to one with a larger RTT. Users with shorter RTTs can increase their congestion window sizes much faster than users that must wait a longer RTT before increasing their window sizes. Thus, users with shorter RTTs are able to claim a larger proportion of the link bandwidth than users with longer RTTs. In [9], we see that average link utilization can be very low with TCP as a transport layer protocol for large delay-bandwidth product channels.

We focus on defining a more efficient transport protocol for flow-switched networks that is neither a modification of TCP nor derived from TCP. In the definition of our protocol, we assume a scheduled optical flow-switched network architecture. In this work, we provide specific examples of the transport layer protocol for OFS. The property of the physical layer considered in this thesis has aspects that are peculiar to the optical network and some aspects that are similar to satellite and wireless networks. However, since some of the parameters and approximations used here are special for optical networks, further research is required to design the transport layer of flow-switching over wireless networks and satellite networks.

1.3 Transport Layer Objectives

In flow-switched networks, flow transmission occurs in two stages: session setup and data transmission. In the first stage, session setup, users communicate with a scheduler in the control plane to reserve network resources. User requests are blocked by the scheduler if network resources are unavailable or heavily loaded. Users agree upon the transmission rate during session setup through the scheduler. The traffic generated for communication between users and the scheduler is irregular and small compared with flow data. Therefore, messages generated in the reservation stage of data

transmission are sent over a TCP/IP electronic packet switching network or similar electronic network architecture.

In the second stage of optical flow-switching, data transmission, users transmit data all-optically end-to-end until completion. The objectives of the transport layer protocol are to provide congestion control, bandwidth matching, and the reliable delivery of flow data. In flow-switching, however, congestion control and bandwidth matching are accomplished through scheduling and blocking at the ingress queue of the network during the session setup phase. Additional network capacity is provided through network reconfiguration performed on the order of minutes, not on a per-transaction basis. Thus, congestion control is performed via admission control at the network entry points where all queueing occurs and is unnecessary within the network. We assume that intermediary switches in the network do not buffer data. Therefore, the transport layer protocol does not need to continually perform the tasks of rate matching and flow control during each session.

The end-to-end reliable delivery of flow data in the presence of both random errors and burst errors is the focus of this work and is described in Chapter 3. Our general transport protocol incorporates error-detection, error-correction, and a retransmission protocol. Due to the dynamic nature of flow-switching, session durations are finite and do not last indefinitely. If a session expires before reliable flow data transmission is completed, it may be necessary to request a new session to send outstanding data. Therefore, the two stages of flow-switching data transactions may be cyclic.

1.3.1 Metrics

In addition to end-to-end reliable delivery, we use two metrics to analyze the performance of a transport layer protocol: throughput and delay. Throughput is defined as the ratio of the useful data rate delivered to the users to the total amount of capacity. Delay is defined as the expected transaction time. The goal of an efficient transport layer design is to maximize throughput and minimize delay but usually cannot be done simultaneously.

The optimization of throughput and optimization of delay are in general not equivalent and require very different operating points for the network. At the optimal (high) throughput of an OFS network, the probability of an available flow session immediately is low and thus either the queueing delay or the blocking probability at the ingress of the network is high. Hence, the expected time to

transaction completion suffers due to long queueing delay and/or added setup time due to repeated attempts to access the network upon blocking.

Conversely, to minimize delay, the maximum network loading and maximum achievable throughput is limited. To optimize delay, network utilization should be lower, extra capacity is added once the aggregate average channel loading passes a set threshold. Thus, throughput performance suffers when delay is minimized.

Providing extra capacity and monitoring network loading to minimize delay increases network capital and operating expenditures. While some users value delay performance, other users may tolerate increased delay for lower cost. In this thesis, we present performance optimizations over throughput and delay independently rather than as a joint optimization over both metrics. In doing so, we allow users and network operators to define network loading and operating conditions. The reader is referred to [11] to see how low setup and queueing delay can be achieved in high load networks at the expense of more effort in setting up flows via probing multiple paths simultaneously.

1.4 Thesis Organization

Chapter 2 introduces the physical layer model, assumptions and error mechanisms. Error-detection and error-correction codes will be used just as in current fiber links. We will also describe the OFS network architecture and physical link outage parameters. Chapter 3 introduces the OFS transport layer protocol. We will describe in detail the two stages of a flow session: session setup and data transmission.

In Chapter 4 and 5, we find the optimal throughput and delay of four example transport layer protocols: Simple Transport Protocol, Simple Transport Protocol with Interleaving, Transport Protocol with Framing, and Transport Protocol with Framing and Interleaving. Chapter 6 concludes our work on the transport layer protocol design for optical flow-switching, where we compare the delay and throughput performance of the four transport layer protocols described in this work.

Chapter 2

2 OFS Network Architecture and Physical Layer

2.1 Network Architecture

In this section, we present a summary of the OFS network architecture presented in [5] [6] [12]. OFS is an all-optical, end-to-end transport service. Users request a dedicated, end-to-end session for the duration of a file transfer. Collisions due to contention are avoided due to scheduling. When a session terminates, network resources are immediately relinquished to other users [12].

The OFS architecture is expected to serve large transactions, where the network management burden and cost of end-user equipment required to set up an end-to-end, all-optical connection required to serve small transactions outweighs the benefits of OFS. Instead, small transactions should be serviced by electronic packet-switched networks. OFS exploits wavelength division multiplexing (WDM) with optical amplification and switching in the Wide Area Network (WAN). The smallest granularity of bandwidth that can be reserved is a wavelength [5]. Session durations are assumed to be on the order of hundreds of milliseconds or longer [12]

Scheduling messages are sent over an electronic control plane. Data files are sent over an all-optical data plane. In the data plane, all queueing of data occurs at the end users and electronic routers are replaced with bufferless optical cross connects (OXC) in the core network. Network reconfiguration (lightpath reconfiguration) is performed over durations of many sessions (on the order of seconds or minutes or even longer), resulting in a quasi-static network topology reacting only to trends but not per flow requests [5].

An example of the OFS network physical architecture is shown in Figure 2-1 for two users. The ingress scheduler resides at the gateway between the ingress MAN (Metropolitan Area Network) and WAN. The egress scheduler resides at the gateway between the egress MAN and WAN [12]. A MAN node is connected to one or more Distribution Networks (DNs) [6].

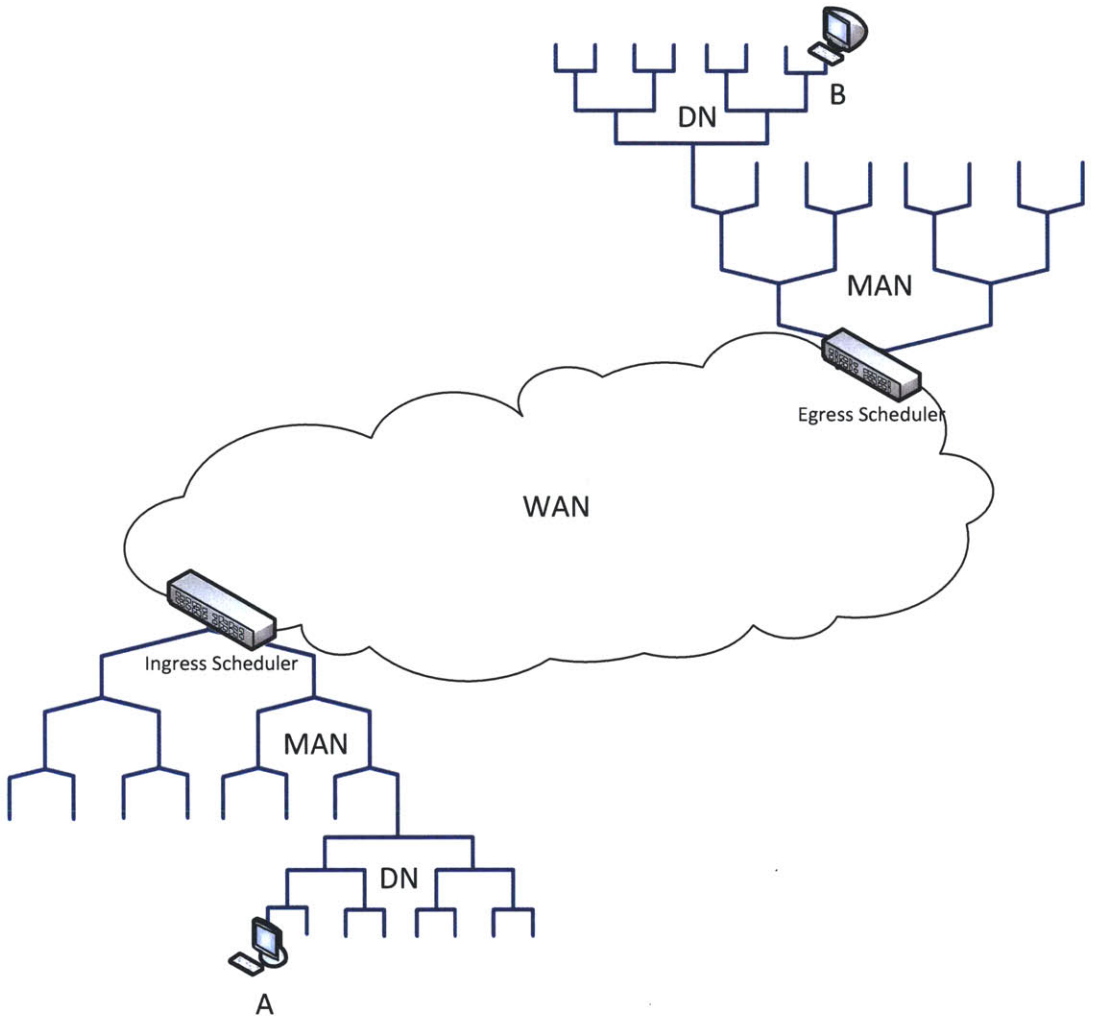


Figure 2-1: OFS Network Physical Architecture.

2.2 Error Mechanisms

We consider two physical layer error mechanisms in OFS: random errors and burst errors. In this work, our focus is on combating burst errors. We assume that bits corrupted due to random errors are independent of previous and future corrupted bits. In our protocol, we assume forward error-correction (FEC) is added to correct for random, independent and identically distributed (IID) errors. The required strength of the error-correcting code depends on the frequency of occurrence of random errors in the physical layer.

In flow-switched optical fiber links, there is the possibility of burst errors that disrupts transmission as a result of transient power excursions [13]. In OFS, a user's traffic occupies an entire wavelength division multiplexed (WDM) channel and dynamically enters and exits the network on a per-transaction time scale. However, currently, erbium doped fiber amplifiers (EDFAs) are limited in their ability to support fast dynamic loading conditions at each wavelength level [14].

The transient problem has been studied in EDFA based systems [13] [14]. When the power in one or more WDM channel abruptly changes, a power excursion is generated on one or more channels. Transients persist until network control elements are able to restore channels to their target powers. A transient event results from power coupling between different WDM channels via nonlinear or optical intensity-dependent components or devices [13].

In amplifiers operated to maintain constant power, an increase in the power of one channel requires a decrease in the power of another channel [14]. Power coupling occurs when the number of channels loading the amplifier changes, resulting in changes to the output power per channel. In amplifiers operated to maintain constant gain on the total power, residual wavelength dependent gain ripple and tilt causes a non-constant gain on the signal power of each channel [13]. Power coupling occurs due to both linear and non-linear wavelength dependent gain ripple and tilt [14]. In current systems, the channel transmission bit error ratio is a function of the channel power [13]. Power variations translates into performance degradations [14].

Power transients in optical fiber flow-switched networks may also result due to network reconfigurations and fiber breaks or other such faults [13]. Network reconfigurations accommodate changing capacity demands in flow-switched optical networks by adding or subtracting wavelengths in the network. However, in the OFS architecture we are considering, network reconfigurations respond to average traffic changes over many sessions rather than per-session traffic changes. Thus, the majority of power transients seen during data transmission are a result of dynamic session additions and terminations rather than network reconfigurations.

In addition to transient power excursions, steady-state power excursions arising from new wavelength additions in a WDM system has been studied in [15]. Steady-state power excursions are power deviations that persist after transient control response decays [16]. Quenching steady-state

power excursions requires additional research in network control systems and is outside the scope of this work.

2.3 Physical Layer Model

We represent the link state during a transient power excursion as an outage state. In this work, correlated bit errors experienced during an outage is referred to as a burst error. The rate at which a link returns to a non-outage state is dependent on the rate at which network control elements are able to restore channels to their target powers [13]. For file transfers on the order of tens to hundreds of gigabits and the number of wavelengths per fiber on the order of ten to one hundred wavelengths, we expect many outage periods to occur during a single session. We interchange the terms "burst error" and "outage period" in this work as outage periods yield burst errors during transmission.

In our model, we assume that during a non-outage period, the physical link experiences a low level (comparable to current long duration optical circuits) of IID bit errors. During an outage period, we assume that the physical link experiences burst errors at high bit error probably, $\sim 1/2$. We model the physical link using a two-state Markov Process as shown in Figure 2-2. The Markov model describes a channel with memory [17]. We assume that random bit errors during a non-outage period occur with probability q. Additionally, we assume that data sent during a link outage has a bit error probability of $1/2$.

State 1 corresponds to a non-outage period with IID errors. β_1 denotes the aggregate average session arrival and termination rate at which users transmit on a channel. When a new user begins transmission, the link goes from the non-outage state to the outage state due to power transients as discussed in Section 2.2. Similar effects occur during the turn-off transient. In our model, we assume sessions arrive according to a Poisson process. Let β_1 be the rate of outages. The duration between outage periods is exponentially distributed with mean $1/\beta_1$. State 2 corresponds to an outage period with burst errors. β_2 denotes the average rate at which the link returns to a non-outage state. This parameter depends on the delay of network control elements. As an approximation we also assume the reverse process has exponential waiting time with mean rate β_2.

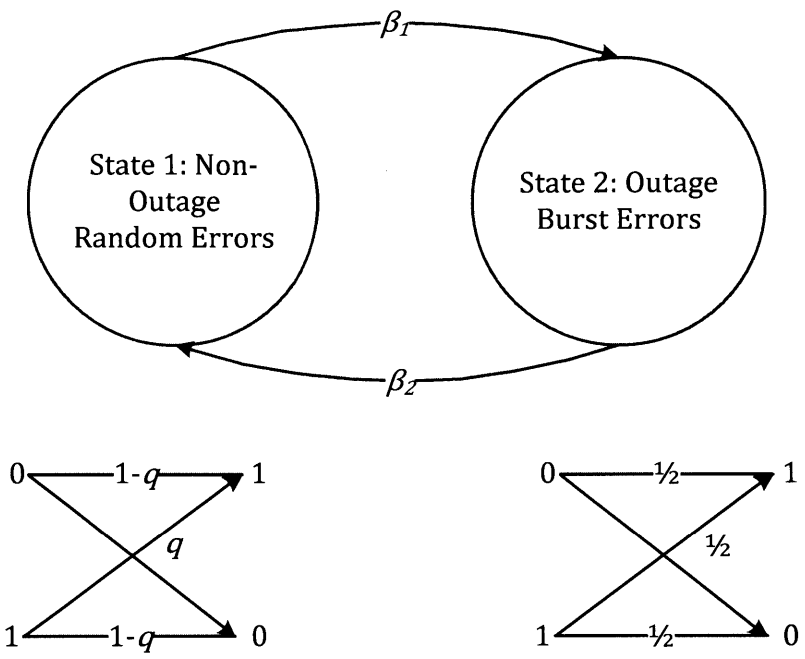


Figure 2-2: Two-state Markov process.

In our analysis, we assume sessions to be spaced contiguously. Therefore, we consider a session completion and new session addition on a wavelength as a single transient event. For lightly loaded networks, the turn-off transient should also be counted as a transient event. However, that case should be automatically taken care of if the protocol works in the high loading regime. Let $\bar{w}$ be the expected number of flow wavelengths per fiber, let Y be the session duration, and let $E[Y]$ be the expected session duration per flow. The expected outage rate is equal to the expected rate of new session arrivals where $\beta_1 = \bar{w}/E[Y]$. We have assumed 100% loading for simplicity. Thus, the statistical model considered here is the worst case scenario that almost never happens even for moderately high loads and the performance results are bounds on what would happen in realistic loading conditions. Let R be the transmission data rate in bits per second. In Table 2-1, we list the expected range of values for β_1, β_2, R, and Y for OFS.

	β_1 Rate from non-outage to outage period	β_2 Rate from outage to non-outage period	R Transmission Rate	Y Session duration
OFS	$1 - 200/s$	$10^3 - 10^6/s$	10 Gbps – 100 Gbps	≥ 100 ms

Table 2-1: OFS link parameters.

2.4 Codes and Interleaving

We will use both error-detection and error-correction codes to mitigate IID errors and burst errors in flows. We use cyclic redundancy check (CRC) codes in OFS data traffic for error detection. A CRC code is based on long division, where the "divisor" is called a generator polynomial. The generator polynomial defines the CRC code. To form a CRC codeword, redundant bits are added to a data string so that the resulting codeword is "divisible" by the generator polynomial [18].

At the receiver, the coded message is divided by the generator polynomial. If the remainder is nonzero, the message contains an error and the message is discarded. If the remainder is zero, the message either does not contain an error or contains an undetectable error. A codeword generated by any polynomial with more than one term detects all single errors. A codeword generated by a polynomial of degree x detects any burst error of length x or less [19].

In OFS, we propose that error-correction be performed through both forward error-correction (FEC) and an Automatic Repeat-reQuest (ARQ) protocol. Information theoretic techniques show there exist codes with an arbitrarily small probability of error for long block lengths [20]. Since all flows have finite lengths, there will always be some residual errors. ARQ is used to correct for these infrequent errors. We propose the use of turbo codes and Low-Density Parity Check (LDPC) codes as practical FEC codes for optical flow-switching. These codes are used today in high end fiber communication systems.

Figure 2-3 shows the sequence of error-detection and error-correction at the transmitter and receiver. At the transmitter, a CRC is appended, followed by an error-correcting code. FEC is applied over both the message and the CRC. The encoded message is sent over a burst-error channel (a channel with both random and correlated errors). At the receiver, the reverse process is performed. First, error-correction is performed, followed by error-detection. If an error is detected, the message is discarded and retransmission is requested through a retransmission protocol.

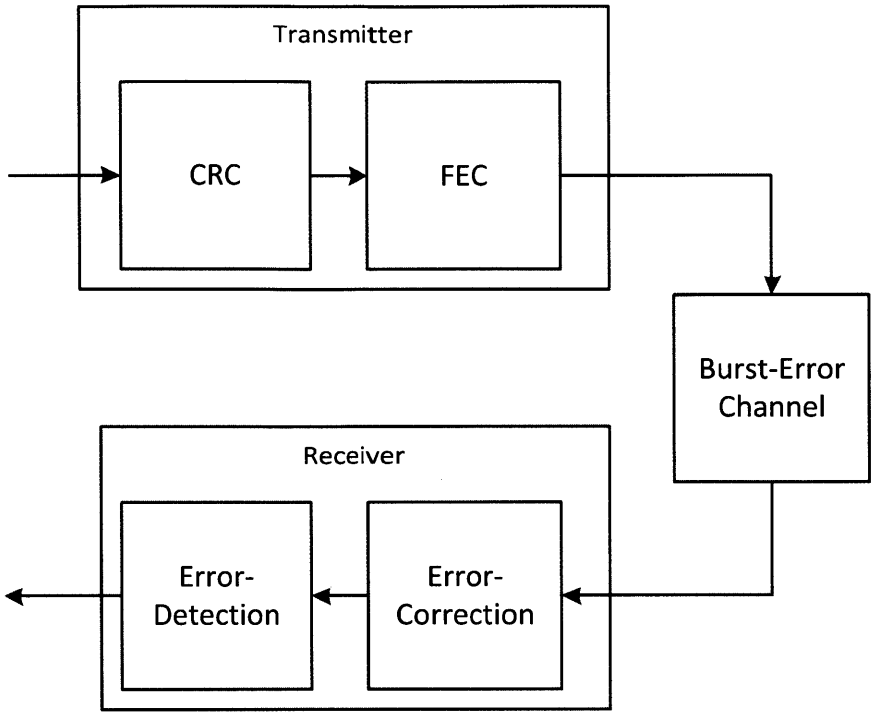


Figure 2-3: Block diagram of message transmission error control.

The performance of codes designed to operate on memoryless channels degrades in channels with memory. Channels with memory can be converted into essentially memoryless channels with the use of interleaving. With interleaving, symbols in a given codeword are transmitted at widely spaced intervals of time (usually ~10x the burst duration). If the channel memory fades with increasing time separation, interleaving allows the channel noise affecting successive symbols in a codeword to be essentially independent [21]. The cost of interleaving includes buffer storage and additional transmission delay. Interleaving can be impractical if the channel memory is very long compared to the symbol transmission time [22]. The distance between successive symbols after interleaving is defined as the interleaver depth. Symbols within the same codeword should be separated by a distance larger than the expected outage duration.

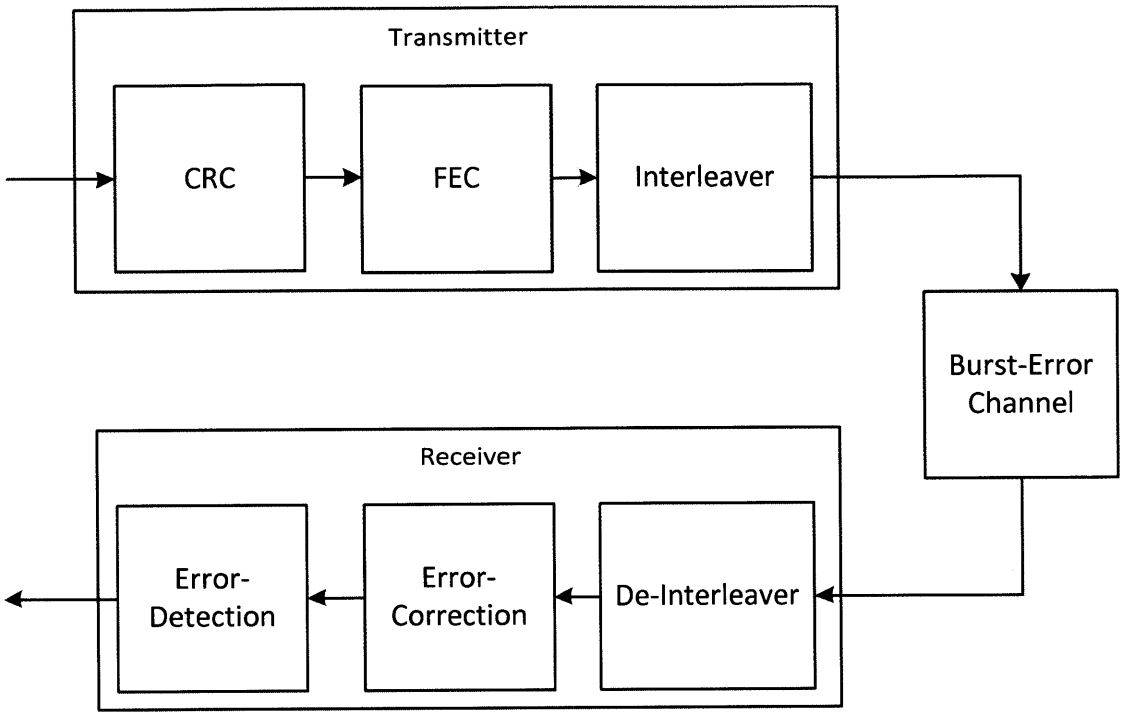


Figure 2-4: Block diagram of message transmission with interleaving.

Let J be the depth of the interleaver. J should be greater than the expected outage duration. We assume that $J > R/\beta_2$. The optimal distance between symbols is beyond the scope of this work but is generally ≥ 10x the burst length.

Figure 2-4 shows the sequence of error-detection and error-correction at the transmitter and receiver with interleaving. Depending on the interleaver depth and outage coherence time, interleaving can enable an error-correcting code to correct for both random and burst errors. Figure 2-5 is an example of a rectangular de-interleaver over a message of length V. During the de-interleaving procedure, data bits are written from left to right and read from top to bottom [18].

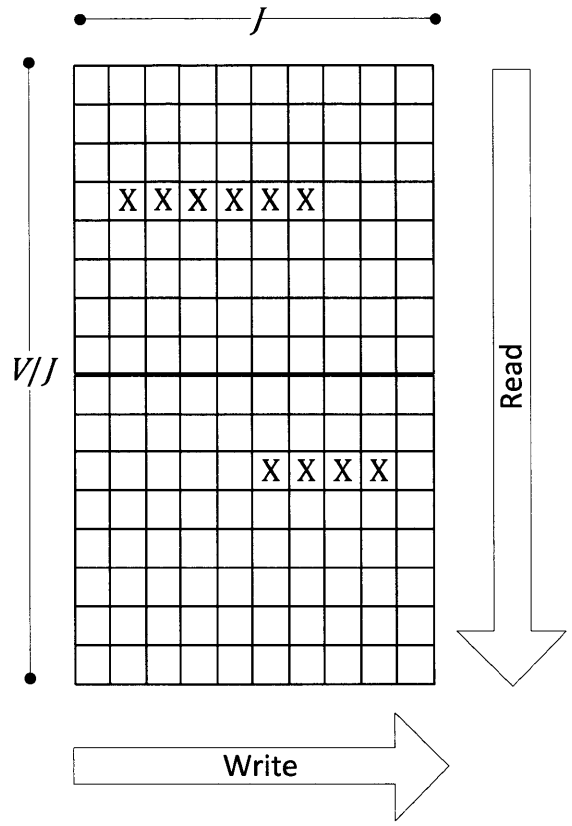


Figure 2-5: Example of a rectangular de-interleaver. Squares with an "X" correspond to burst errors during an outage period.

Chapter 3

3 OFS Transport Layer Design

3.1 Transport Layer Protocol for OFS

TCP is not best suited for OFS [12]. TCP limits a sender's rate of window increase and takes a long time to achieve the full rate of a wavelength channel even if the upper limit of window size is removed. In addition, TCP has the wrong reaction to outages due to its congestion control mechanisms severely reducing throughput [9]. In OFS, flows are scheduled and congestion control is performed by a scheduling algorithm. Thus, flow control and fair resource allocation through windowing in TCP are unnecessary in the OFS transport layer protocol and should be eliminated due to its detrimental effects on throughput [12]. The OFS transport layer protocol only needs to provide end-to-end reliable delivery of data. In our protocol, flow transmission occurs in two stages: session setup and data transmission.

3.1.1 Session Setup

We describe a scalable scheduling algorithm for inter-MAN OFS communication based on the algorithms presented in [5] [6] [12]. The scheduling algorithm presented in this work is for the basic OFS service. An ultrafast service with one round-trip time for scheduling and low blocking probability is also available and can be found in [11] [12]. This ultrafast service has a different session setup mechanism than the one described here but the same transport layer protocol can be used for that service as well.

At the MAN-WAN interface, we assume the absence of wavelength conversion capability and that wavelength continuity between WAN and MAN wavelength channels is respected. In the reference architecture described in [6] we see that a computationally tractable scheduling algorithm requires that for each wavelength channel provisioned for inter-MAN OFS communication in the WAN, there exists a dedicated wavelength channel in both source and destination MANs. Thus, we assume a dedicated wavelength channel exists from the edge of the source DN to the edge of the destination DN that passes through the ingress MAN, the WAN, and the egress MAN [6].

We also assume, via quasi-static wavelength assignment and routing, a one-to-one correspondence between OFS wavelengths at each DN to its parent MAN and a continuous connection to the WAN [6] [12]. We have assumed a quasi-static WAN logical topology with inter-MAN connections changing in the time scale of average traffic shifts on the order of many flows. Thus, we assume that in the problem of the design of the transport layer protocol, wavelength channels provisioned for end-to-end user communication can be considered to be static.

Control plane messages are sent over a guaranteed datagram transport layer protocol such as TCP. Users are not allowed to change scheduler policies nor directly usurp resources. Preventing users from direct access to control plane algorithms improves security and also prevents unfair resource allocation [12]. We require that users be allowed to schedule only one session at a time.

A scheduling request originates from the source end-user residing in the ingress DN within the ingress MAN. The flow generated at the source user is destined for an end-user residing in the egress DN within the egress MAN. The source requests a session by sending a scheduling request to the ingress scheduler. Within the scheduling request, the source provides the length of the flow, its transmitter availability, and its transmitter rate. At the ingress scheduler we assume a first-in first-out (FIFO) queue for every possible MAN destination, although any queueing discipline can be used with this transport protocol. The queueing delay of the FIFO queue is given in this thesis. When other types of queues are used, their delay performance can be used in the performance expressions given in Chapter 4 and 5. When the request arrives at the ingress scheduler, it is placed at the end of the queue associated with the egress MAN [6]. The length of the scheduling queue is finite. An overflowed queue results in blocked requests [12].

Once the request reaches the head of the queue, the ingress scheduler requests for the destination end-user's receiver availability and receiver rate. The destination node communicates with the ingress scheduler through the egress scheduler. If the destination end-user is unavailable, the request is blocked. The ingress scheduler assigns the flow to an available wavelength channel [6]. The ingress scheduler informs the source and destination end-users of the session wavelength, start time, rate, and duration. Once the source and destination receive and acknowledge the session parameters, the request departs from the scheduling queue and an all-optical end-to-end path is reserved for the

entire session duration. The new session is added to a flow sequence that is kept by the ingress scheduler.

We assume that all clocks at the end-users and at the schedulers are synchronized. We summarize the scheduling algorithm for inter-MAN OFS communication in the following algorithm description. This is a version of a scalable algorithm but not necessarily optimum. Let w be the assigned wavelength channel, let t_{start} be the flow start time, and let y be the reserved session duration.

- User A initiates a session by establishing a TCP/IP connection with user B announcing his intentions and upon acknowledgment by user B, sends a scheduling request to the ingress scheduler. Within the scheduling request, User A sends the length of the flow, its transmitter availability, and its transmitter rate, $\{y, A[\,], R_A\}$.
- When the request arrives at the ingress scheduler, it is placed at the end of the queue associated with the egress MAN of the destination.
- Once the request reaches the head of the queue, the ingress scheduler requests for User B's receiver availability and receiver rate, $\{B[\,], R_B\}$.
- The ingress scheduler assigns the flow to the first available wavelength channel.
- The ingress scheduler informs User A and User B of the session wavelength, start time, rate, and duration, $\{w, t_{start}, R, y\}$.
- User A and User B receive and acknowledge $\{w, t_{start}, R, y\}$ to the ingress scheduler.
- The request departs from the scheduling queue and the ingress scheduler adds the new session to the flow sequence.
- User A waits until t_{start} to begin data transmission. At t_{start}, User A tunes its transmitter to w and User B tunes its receiver to w.
- A TCP connection between User A and B would have been started for ARQ purposes on the flow before data transmission begins.

Figure 3-1: Summary of the scheduling algorithm for OFS.

3.1.2 Data Transmission

During session setup, network resources are reserved from the source end-user to the destination end-user starting at time t_{start}. The channel remains contention-free throughout the session until time $t_{start} + y$. The source begins data transmission at time t_{start} and transmits at the full data rate and on the wavelength channel assigned by the scheduler. At time $t_{start} + y$, the session terminates and network resources are immediately relinquished to other users [12]. At time $t_{start} + y$, if the flow is received without error at the destination, the transmission is complete. Otherwise, an additional session is requested for data retransmission. Although not explored in our work, users may also request for retransmission over IP rather than request for reflow if the frame that needs retransmission is small.

We propose four example transport layer protocols that guarantee the reliable delivery of data over a burst-error channel. The delay and throughput of each protocol in found in Chapter 4 and 5. The transport layer protocols considered in our work are:

1. the Simple Transport Protocol (STP),
2. the Simple Transport Protocol with Interleaving (STPI),
3. the Transport Protocol with Framing (TPF),
4. the Transport Protocol with Framing and Interleaving (TPFI).

In STP, we use a CRC for error-detection. In addition, an error-correction code is applied to correct for random errors. In STP, the entire file is transmitted as one large frame. Therefore, STP allows for low framing overhead and no segmentation and reassembly overhead. At the destination once the file is received, the receiver performs error-correction followed by error-detection. If the CRC detects an error, the received file is discarded and the destination requests that the file be retransmitted.

We observe that as the file size increases, the probability that a file experiences an outage period also increases. STP can be used if the expected interval between outages is known to be much longer than the session duration. Failed transmissions and the need for retransmissions lead to a decrease in throughput and delay performance.

In STPI, a file is transmitted from source to destination, without segmentation as in STP. In STPI, interleaving is also performed at the channel input and de-interleaving is performed at the channel output. With interleaving, we assume symbols in a given codeword are transmitted at widely spaced intervals of time compared to an outage duration. Interleaving allows the channel noise affecting successive symbols in a codeword to be essentially independent [21]. FEC is then designed to correct for both random errors and burst errors.

In TPF, a large file is segmented into smaller frames at the transmitter and reassembled at the receiver. TPF provides users the option to reserve additional time per session for frame retransmissions. In Chapter 5, we find the optimal delay, optimal throughput, optimal additional session reservation time, and optimal frame size over the channel statistics. If the optimal frame size is equal to the transmission file size, TPF is equivalent to STP.

In TPF, the transmitter appends a CRC to each frame for error detection. An error-correction code is also applied to each frame to correct for random errors. At the session start time, the source begins transmitting frames sequentially. After the transmitter has finished transmitting all data frames once, the transmitter processes and sends retransmission frames. At the receiver, if an error is detected in a received frame, it is discarded and the destination requests that the erroneous frame be retransmitted via ARQ using the reverse electronic packet-switched channel which can be over a TCP/IP network. Although unlikely in an OFS network, we assume frames can be lost due to network errors. We assume that if the sequence number of a received frame is larger than the next expected sequence number, intermediary frames have been lost in the network and the destination requests that the missing frames be retransmitted. If a received frame is error-free and is the next expected frame, the user passes the frame up to the application. All other correct frames not in sequence will be placed in a buffer and wait for retransmission of missing intermediate frames before passing frames to the application layer in the correct sequence. The session terminates with two possible results: with outstanding erroneous or missing frames or with all frames received without error. If the session terminates with outstanding frames, the source sends a session request for data retransmission. A new flow is requested and outstanding frames are transmitted.

In TPFI, interleaving is performed at the channel input and de-interleaving is performed at the channel output. We assume that if interleaving is performed at the channel input, FEC can be used to

correct for both random errors and burst errors. If the length of a frame is large enough, we assume that interleaving is performed across a single frame. This has the advantage of requesting frame retransmission once the frame is processed and deemed to have errors instead of having to wait until the whole file is received. If the frame size is too small, interleaving across multiple frames but not the entire file is a possible variant of this algorithm. In Chapter 5 we find the optimal delay, optimal throughput, optimal additional session reservation time, and optimal frame length.

3.2 Algorithm Flow Chart

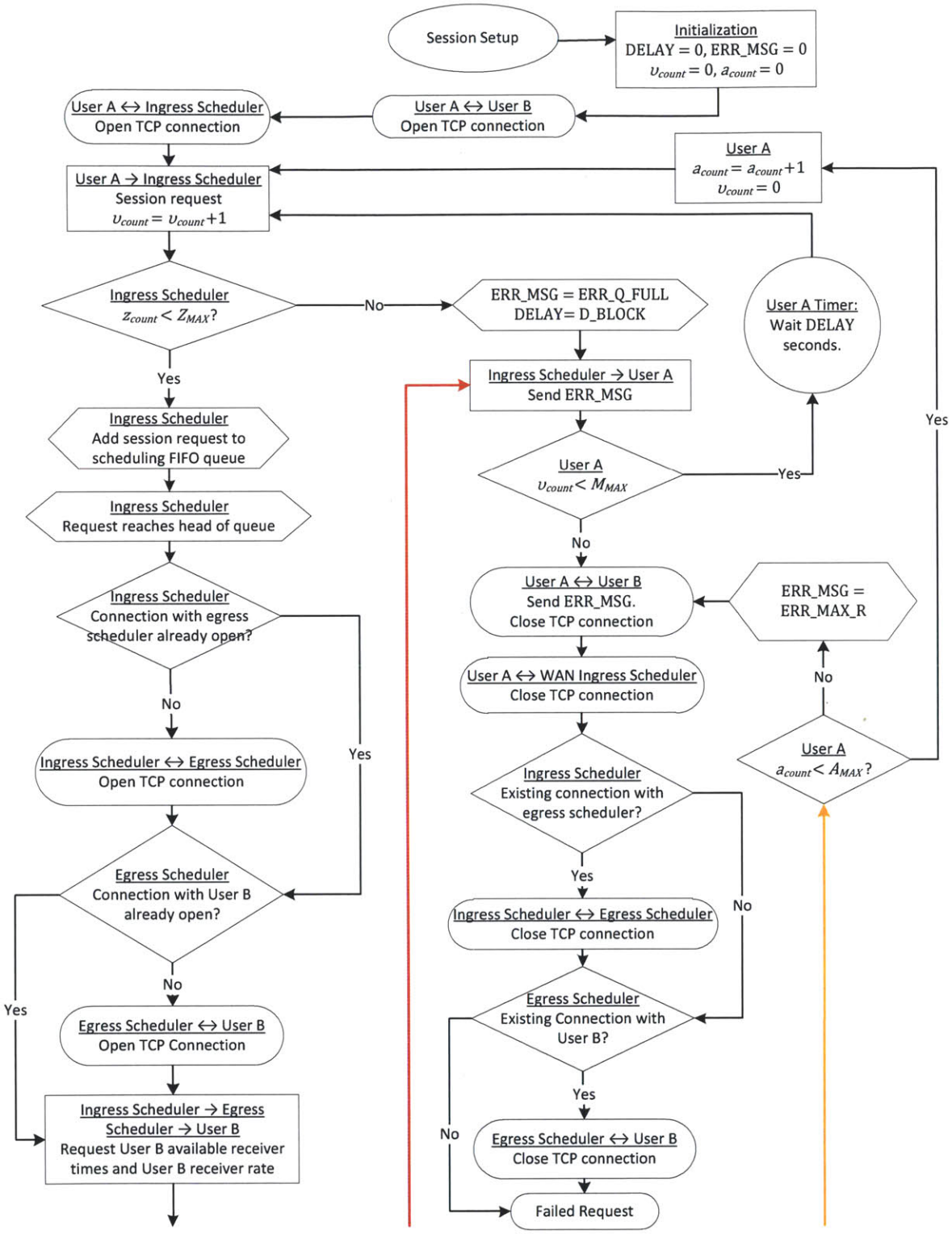
In Figure 3-2, we provide a flow chart of the transport layer protocol for inter-MAN OFS communication. In Table 3-1 we list timers, counters, and errors messages of the OFS transport layer protocol.

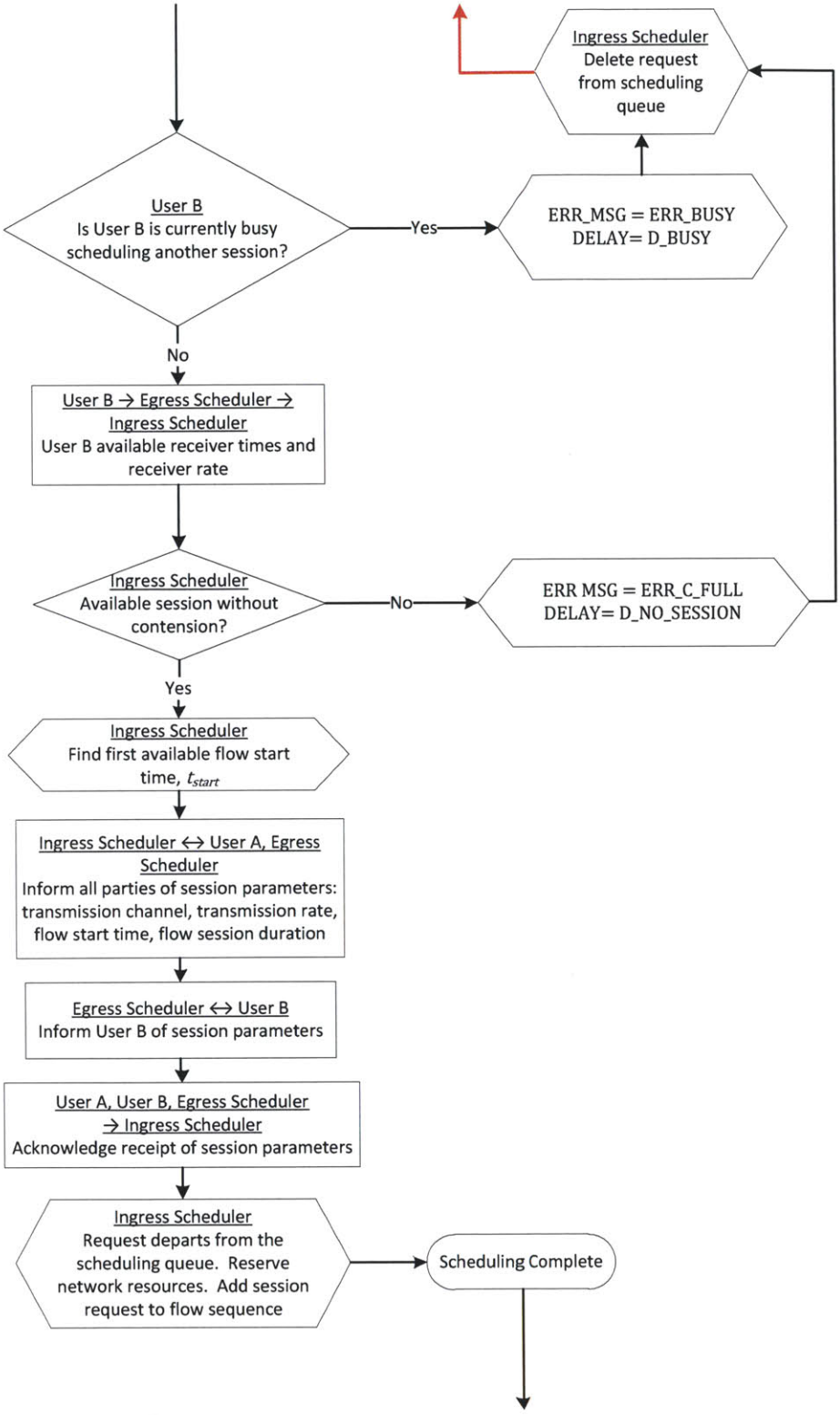
Timers (DELAY)	
D_BLOCK	The delay a user must wait before reattempting a scheduling request after the request was blocked at the scheduling queue.
D_BUSY	The delay a user must wait before reattempting a scheduling request after the request was blocked because User B was busy.
D_NO_SESSION	The delay a user must wait before reattempting a scheduling request after the request was dropped due to unavailable sessions.

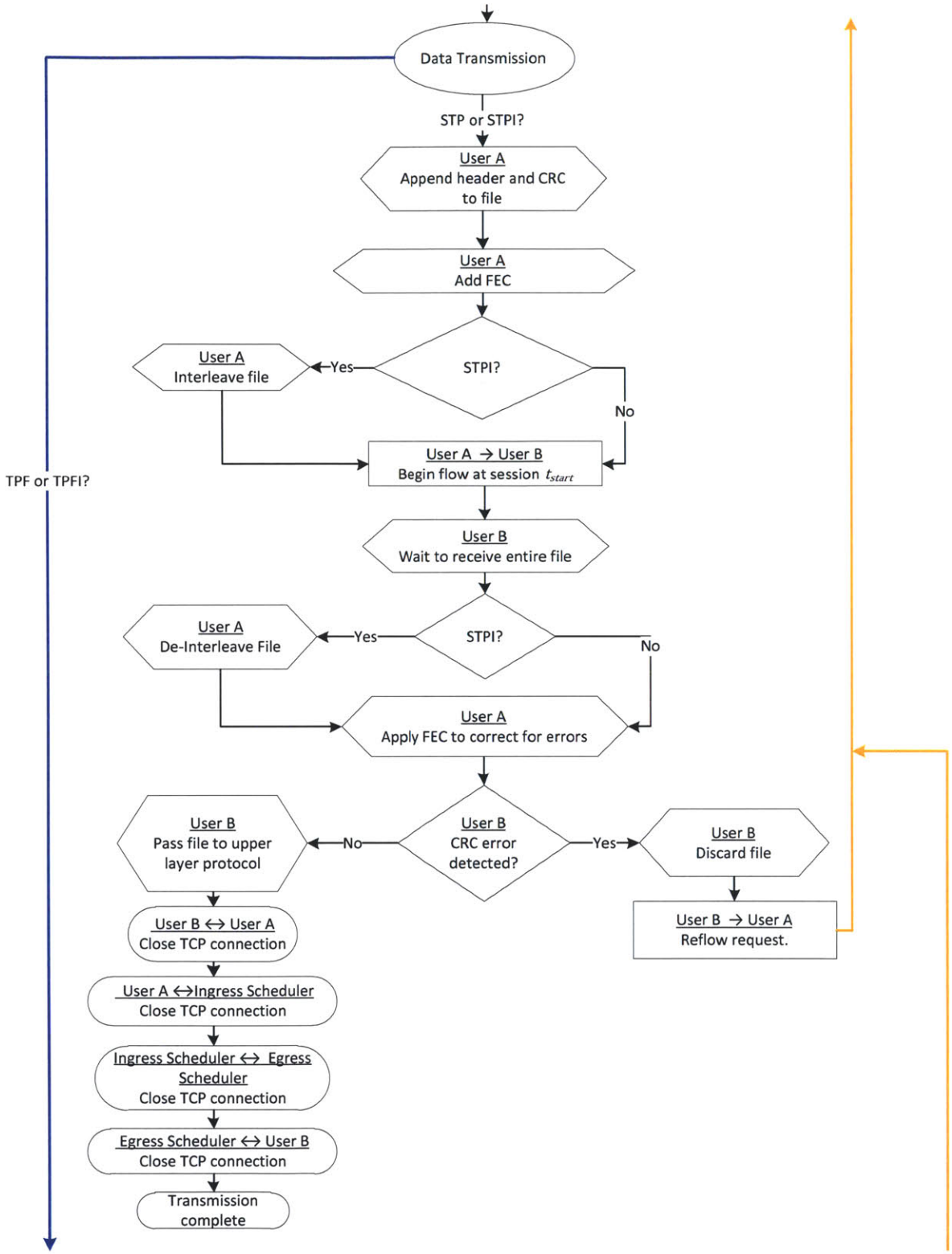
Counters and Limits	
v_{count}	The number of scheduling attempts including the first scheduling attempt.
M_{MAX}	The maximum number of scheduling attempts.
a_{count}	The number of retransmission requests.
A_{MAX}	The maximum number of retransmission requests.
z_{count}	The number of requests currently in the scheduling queue.
Z_{MAX}	The maximum number of requests in the scheduling queue.

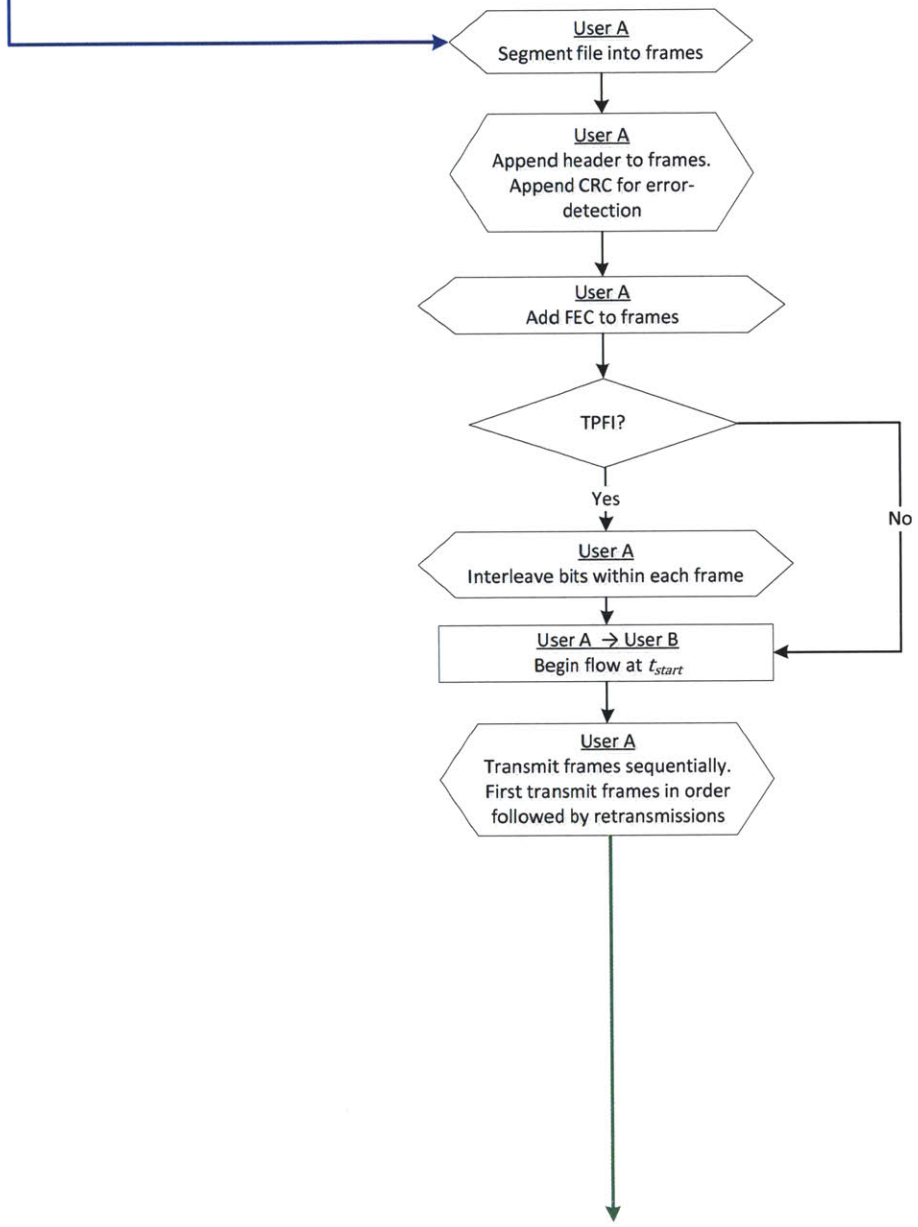
Error Messages (ERR_MSG)	
ERR_Q_FULL	An overflowed scheduling queue.
ERR_BUSY	User B is not available.
ERR_C_FULL	No open sessions available.
ERR_MAX_R	The maximum number of retransmission attempts has been reached.

Table 3-1: Algorithm Timers, Counters, and Error Messages.


Session Setup
Initialization
DELAY = 0, ERR_MSG = 0
$v_{count} = 0$, $a_{count} = 0$
User A ↔ User B
Open TCP connection
User A ↔ Ingress Scheduler
Open TCP connection
User A
$a_{count} = a_{count} + 1$
$v_{count} = 0$
User A → Ingress Scheduler
Session request
$v_{count} = v_{count} + 1$
Ingress Scheduler
$z_{count} < Z_{MAX}$?
No
ERR_MSG = ERR_Q_FULL
DELAY= D_BLOCK
User A Timer:
Wait DELAY
seconds.
Yes
Ingress Scheduler → User A
Send ERR_MSG
Ingress Scheduler
Add session request to
scheduling FIFO queue
User A
$v_{count} < M_{MAX}$
Yes
Yes
Ingress Scheduler
Request reaches head of queue
No
Ingress Scheduler
Connection with egress
scheduler already open?
User A ↔ User B
Send ERR_MSG.
Close TCP connection
ERR_MSG =
ERR_MAX_R
User A ↔ WAN Ingress Scheduler
Close TCP connection
No
No
Ingress Scheduler ↔ Egress Scheduler
Open TCP connection
Yes
User A
$a_{count} < A_{MAX}$?
Ingress Scheduler
Existing connection with
egress scheduler?
Egress Scheduler
Connection with User B
already open?
Yes
No
Ingress Scheduler ↔ Egress Scheduler
Close TCP connection
No
Yes
Egress Scheduler ↔ User B
Open TCP Connection
Egress Scheduler
Existing Connection with
User B?
Ingress Scheduler → Egress
Scheduler → User B
Request User B available receiver
times and User B receiver rate
Yes
No
Egress Scheduler ↔ User B
Close TCP connection
Failed Request


Ingress Scheduler
Delete request from scheduling queue
User B
Is User B is currently busy scheduling another session?
Yes
ERR_MSG = ERR_BUSY
DELAY= D_BUSY
No
User B → Egress Scheduler → Ingress Scheduler
User B available receiver times and receiver rate
Ingress Scheduler
Available session without contension?
No
ERR MSG = ERR_C_FULL
DELAY= D_NO_SESSION
Yes
Ingress Scheduler
Find first available flow start time, t_{start}
Ingress Scheduler ↔ User A, Egress Scheduler
Inform all parties of session parameters: transmission channel, transmission rate, flow start time, flow session duration
Egress Scheduler ↔ User B
Inform User B of session parameters
User A, User B, Egress Scheduler → Ingress Scheduler
Acknowledge receipt of session parameters
Ingress Scheduler
Request departs from the scheduling queue. Reserve network resources. Add session request to flow sequence
Scheduling Complete


Data Transmission
STP or STPI?
User A
Append header and CRC to file
User A
Add FEC
STPI?
Yes
User A
Interleave file
No
User A → User B
Begin flow at session t_{start}
User B
Wait to receive entire file
STPI?
Yes
User A
De-Interleave File
No
User A
Apply FEC to correct for errors
User B
CRC error detected?
No
User B
Pass file to upper layer protocol
Yes
User B
Discard file
User B → User A
Reflow request.
User B ↔ User A
Close TCP connection
User A ↔Ingress Scheduler
Close TCP connection
Ingress Scheduler ↔ Egress Scheduler
Close TCP connection
Egress Scheduler ↔ User B
Close TCP connection
Transmission complete
TPF or TPFI?


User A
Segment file into frames
User A
Append header to frames.
Append CRC for error-detection
User A
Add FEC to frames
TPFI?
Yes
No
User A
Interleave bits within each frame
User A → User B
Begin flow at t_{start}
User A
Transmit frames sequentially.
First transmit frames in order
followed by retransmissions

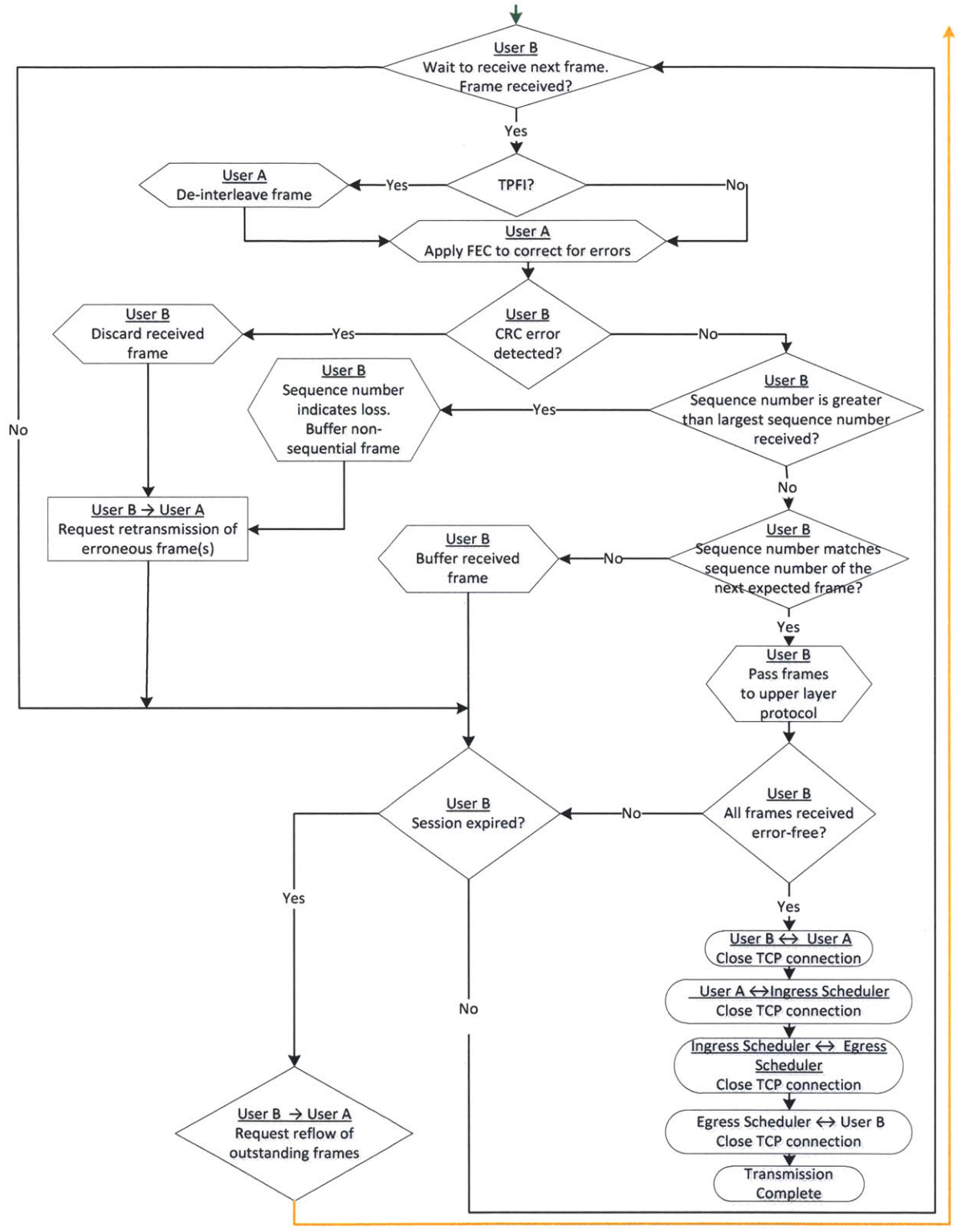
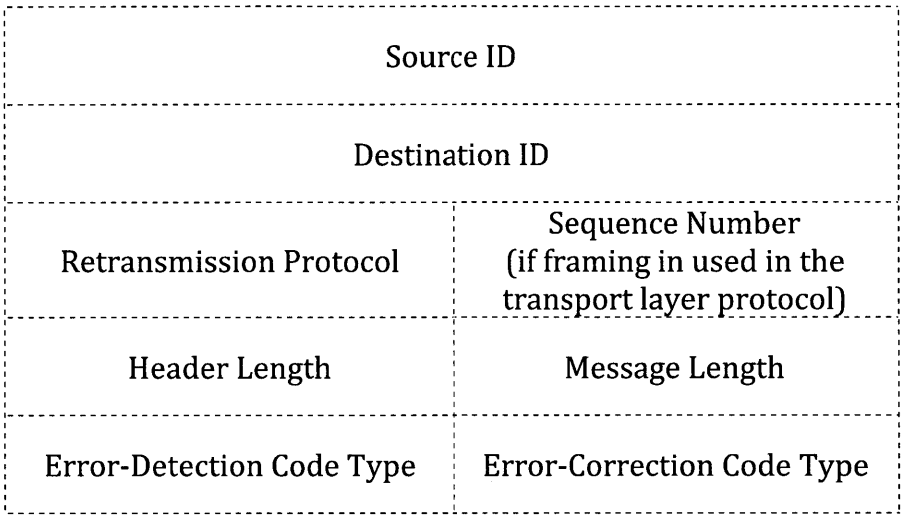


Figure 3-2: OFS Transport Layer Protocol Flow Chart.

3.3 Overhead



Parameter	Description
Source ID:	Source Identifier.
Destination ID:	Destination Identifier.
Retransmission Protocol:	Retransmission Protocol type.
Sequence Number:	If framing is used in the transport layer protocol, this field is used in the ordering of frames. If framing is not used, this field is set to 0.
Header Length:	Header length.
Message Length:	Length of the message.
Error-Detection Code Type:	Error-Detection code type applied at the transmitter (e.g., CRC, checksum).
Error-Correction Code Type:	Error-Correction code type applied at the transmitter (e.g., LDPC, turbo codes).

Figure 3-3: OFS Header.

In this section, we describe the header, error-detection, error-correction overhead, and preamble of the general transport layer protocol. In OFS, the flow header is a fixed overhead appended to every

message and does not vary with the message length. In Figure 3-3, we illustrate a possible design for the OFS header.

In error-detection codes, the transmitter introduces redundancy, which the receiver uses to detect the presence of an error [18]. In Section 2.4, we use a CRC for error-detection in OFS. We saw that a codeword generated by any polynomial with more than one term detects all single errors. Let H_{CRC} be the overhead required to detect both random and burst errors in OFS. A codeword generated by a polynomial of degree H_{CRC} detects any burst of length b where $b \leq H_{CRC}$. The fraction of undetected burst errors is [19]

$$\begin{cases} 2^{-H_{CRC}}, & b = H_{CRC} + 1, \\ 2^{-(H_{CRC}-1)}, & b > H_{CRC} + 1. \end{cases} \tag{3.1}$$

We will use a large enough CRC (> 100 bits) so the probability of an undetected error is negligible and will not be a factor in the reliability of the data transfer.

We use information theoretic techniques to approximate the additional overhead of including an error-correction code. During data transmission, the transmitter adds redundancy to combat errors in the channel [3]. Let μ be the total length (in bits) over which an error-correction code is applied. In our protocol, μ is the sum of the message, OFS header, and CRC overhead. Let N be the length (in bits) of the code. We define U as the rate (in bits per transmission) of a (μ, N) code where $U = \mu/N$. For sufficiently large block lengths, rates U less than the capacity/signaling-rate of the channel yield arbitrarily small error probabilities [20].

We assume that if interleaving is not performed at the channel input, the FEC can only correct for random errors experienced during a non-outage period. We model the channel during a non-outage state as a binary symmetric channel (BSC) with cross over probability q. When the channel has a binary input, the channel input is equal to the channel output with probability $1-q$. Channel capacity is the maximum rate at which a message can be transmitted and reconstructed with an arbitrarily low probability of error with coding. Let C_q be the information capacity of a BSC with parameter q where [20]

$$C_q = 1 - h(q) \tag{3.2}$$

and

$$h(z) = -z\log_2(z) - (1-z)\log_2(1-z).$$

Let K_q be the redundancy added to combat random errors in the channel:

$$\begin{aligned} K_q &= N - \mu \\ &= \mu\left(\frac{1-U}{U}\right). \end{aligned} \tag{3.3}$$

The coding theorem states that for a discrete memoryless channel, all rates below capacity are achievable and for every rate $U<C_q$, there exists a code where the maximum probability of error approaches zero with increasing N [20]. Thus, we can upper bound the code rate by the capacity of the channel, $U<C_q$. We can use this upper bound to lower bound K_q.

$$K_q > \mu\left(\frac{1-C_q}{C_q}\right). \tag{3.4}$$

In OFS, file transfers are on the order of tens to hundreds of gigabits and the practical frame length (to be discussed in Chapter 5) is greater than a Megabit. Therefore, we note that in OFS, both the length of a message μ and the length of a block code N are large. We assume that in OFS, N is large enough such that for every rate $U<C_q$, there exists a code where the maximum probability of error is low. We approximate K_q as follows:

$$K_q \cong \mu\left(\frac{1-C_q}{C_q}\right). \tag{3.5}$$

We assume that if interleaving is performed at the channel input, FEC can be used to correct for both random errors and burst errors. We refer to the two state channel model developed in Section 2.3. Let $Q(t)$ denote the random channel symbol crossover probability at time t and let S_t denote the current channel state at time t, then we have:

$$Q(t) = \begin{cases} q, & S_t = \text{non} - \text{outage}, \\ \frac{1}{2}, & S_t = \text{outage}. \end{cases} \tag{3.6}$$

We assume that if the interleaver depth is long enough, the interleaved channel (the cascade of interleaver, channel, and de-interleaver) may be considered memoryless [23]. We model the interleaved channel as a BSC with crossover probability ξ where

$$\begin{aligned}\xi &= E[Q(t)] \\ &= \frac{\beta_2}{\beta_1+\beta_2}q + \frac{\beta_1}{\beta_1+\beta_2}\frac{1}{2}.\end{aligned} \tag{3.7}$$

Let C_ξ be capacity of the interleaved channel with parameter ξ where

$$C_\xi = 1 + \xi\log_2(\xi) + (1-\xi)\log_2(1-\xi). \tag{3.8}$$

The capacity of the channel where channel state information is available at the receiver is considered in [23], however, is beyond the scope of this work. Using the same argument as in (3.5), we approximate the additional overhead of including an error-correction code. Let K_ξ be the redundancy added to combat both random and burst errors in the channel where

$$K_\xi \cong \mu\left(\frac{1-C_\xi}{C_\xi}\right). \tag{3.9}$$

Let H_{FEC} be the FEC overhead in bits. The error-correction overhead is

$$H_{FEC} = \begin{cases} K_q, & \text{FEC corrects only radom errors,} \\ K_\xi, & \text{FEC corrects radom and burst errors.} \end{cases} \tag{3.10}$$

In OFS, a preamble is added for synchronization and indicates the start of a frame [18]. Let H_P be the length of the preamble. Let o be the probability that the preamble is found within a frame where

$$o \le (\mu + H_{FEC})2^{-H_P}. \tag{3.11}$$

To guarantee that the probability of a preamble collision is less than o requires a preamble overhead of

$$H_P \geq \log_2\left(\frac{\mu + H_{FEC}}{o}\right). \tag{3.12}$$

In our protocol, we will use a large enough preamble (> 64 bits) so that the probability of a collision is negligible and will not be a factor in the reliability of data transmission.

Figure 3-4 illustrates an OFS message plus overhead. Let H_O be the length of the OFS header in bits. We note that both H_O and H_{CRC} do not vary with the message size. Let B be the length of the message in bits.

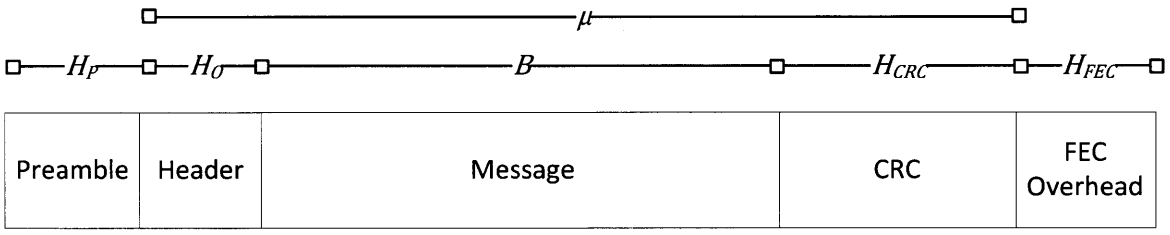


Figure 3-4: OFS Message Framing.

In OFS, we propose that FEC be performed over the header, message, and CRC overhead. The total overhead of an OFS message is equal to the sum of the header length, CRC overhead, error-correction code overhead, and preamble length. Let γ denote the total overhead of an OFS message. We can approximate γ as follows:

$$\begin{aligned} \gamma &\cong H_P + H_O + H_{CRC} + (H_O + H_{CRC} + B) \\ &= H_P + \frac{H_O + H_{CRC}}{C} + B\left(\frac{1 - C}{C}\right) \end{aligned} \tag{3.13}$$

where

$$C = \begin{cases} C_q, & \text{FEC corrects for only radom errors,} \\ C_\xi, & \text{FEC corrects for radom and burst errors.} \end{cases}$$

3.4 Setup Delay

In this section we want to show that the setup delay is at least equal and mostly likely much larger than the transmission and propagation delay. We define the setup delay as the delay from when the source sends a scheduling request to the ingress scheduler to the time the source begins data transmission. Thus, this includes the contention period and the time spent in the sequence holder waiting for transmission. In this section, we lower bound the setup delay as the queueing delay of a request before it emerges the flow sequence and transmission begins right after, assuming the request is not blocked. When the network is very lightly loaded, the setup delay is small. However, when the network is moderate to heavily loaded, users must wait for scheduled sessions to finish before beginning a flow and we will show that the setup delay is a significant fraction of the total delay.

We assume that OFS session requests arrive according to a Poisson process with rate λ [6]. Let Y be the duration of a session. Session durations are modeled as identical and independently distributed random variables with probability distribution $f_Y(y)$ [6]. Let $E[Y]$ be the mean of Y and let $E[Y^2]$ be the second moment of Y.

Let w_m be the number of wavelength channels available for flow traffic from a source MAN to a destination MAN. We approximate the queue at the ingress scheduler for a source-destination pair as w_m M/G/1 queueing system operating under a normalized traffic load of $1/w_m$ [6].

We summarize the results found in [24] for the setup delay under two distribution assumptions for Y: a truncated heavy-tailed distribution and a truncated exponential distribution in Table 3-2. Let ρ be the network load where $\rho = \lambda E[Y]$. Let y_a be the minimum OFS session duration and let y_b be the maximum OFS session duration [24]. We assume that sessions of duration less than y_a seconds do not qualify for OFS service. In [24], they assumed the maximum queue length is unbounded and thus there is not blocking. The results are good approximations of the realistic situation where the flow sequence holder has a maximum length of several sessions and the blocking probability is very small, e.g. $< 10^{-2}$.

Heavy-Tail Distribution
$\alpha \in (0,1)$
$f_Y(y) = \begin{cases} \dfrac{\alpha}{{y_a}^{-\alpha} - {y_b}^{-\alpha}} \dfrac{1}{y^{\alpha+1}}, & for\ y \in [y_a, y_b], \\ 0, & otherwise. \end{cases}$
$E[Y] = \dfrac{\alpha}{{y_a}^{-\alpha} - {y_b}^{-\alpha}} \dfrac{{y_b}^{1-\alpha} - {y_a}^{1-\alpha}}{1-\alpha}$
$E[Y^2] = \dfrac{\alpha}{{y_a}^{-\alpha} - {y_b}^{-\alpha}} \dfrac{{y_b}^{2-\alpha} - {y_a}^{2-\alpha}}{2-\alpha}$

Exponential Distribution
$\gamma \in (0, \infty)$
$f_Y(y) = \begin{cases} \dfrac{\gamma}{e^{-\gamma y_a} - e^{-\gamma y_b}} e^{-\gamma y}, & for\ y \in [y_a, y_b], \\ 0, & otherwise. \end{cases}$
$E[Y] = \dfrac{1}{\gamma} + \dfrac{y_a e^{-\gamma y_a} - y_b e^{-\gamma y_b}}{e^{-\gamma y_a} - e^{-\gamma y_b}}$
$E[Y^2] = \dfrac{2}{\gamma^2} + \dfrac{2}{\gamma} \dfrac{y_a e^{-\gamma y_a} - y_b e^{-\gamma y_b}}{e^{-\gamma y_a} - e^{-\gamma y_b}} + \dfrac{{y_a}^2 e^{-\gamma y_a} - {y_b}^2 e^{-\gamma y_b}}{e^{-\gamma y_a} - e^{-\gamma y_b}}$

Table 3-2: Distribution, mean, and second moment results found in [24].

For stability, the condition $\lambda < 1/E[Y]$ must hold. Let τ_s be the expected setup delay. The setup delay is given by Pollaczek-Khinchin formula for the M/G/1 queue where [24]

$$\tau_s \geq \frac{\rho}{2(1-\rho)} \frac{E[Y^2]}{E[Y]}. \tag{3.14}$$

We plot the lower bound for the expected delay against ρ in Figure 3-5 under the assumption of a heavy-tail distribution of Y and under the assumption of an exponential distribution of Y. $E[Y]$ is the expected transmission delay of a message plus overhead. The setup delay exceeds the transmission delay when

$$\rho \geq \frac{2E[Y]^2}{E[Y^2] + 2E[Y]^2}. \tag{3.15}$$

Let ρ_{min} be the value of ρ that satisfies (3.15) with equality. Using the inequality $E[Y^2] \geq E[Y]^2$, we find an upper bound for ρ_{min} where $\rho_{min} \leq 2/3$. In this work, we assume that at moderate to high network loading the queueing delay is large enough such that the setup delay is a significant fraction of the total delay and that is the region where the user is most concerned about delay. In Chapter 5 we will adopt the strategy that the user will try to avoid multiple requests for new flow sessions due to the significant setup delay. Thus, the protocol assumes at most one request for reflow that only occurs with rather low probability.

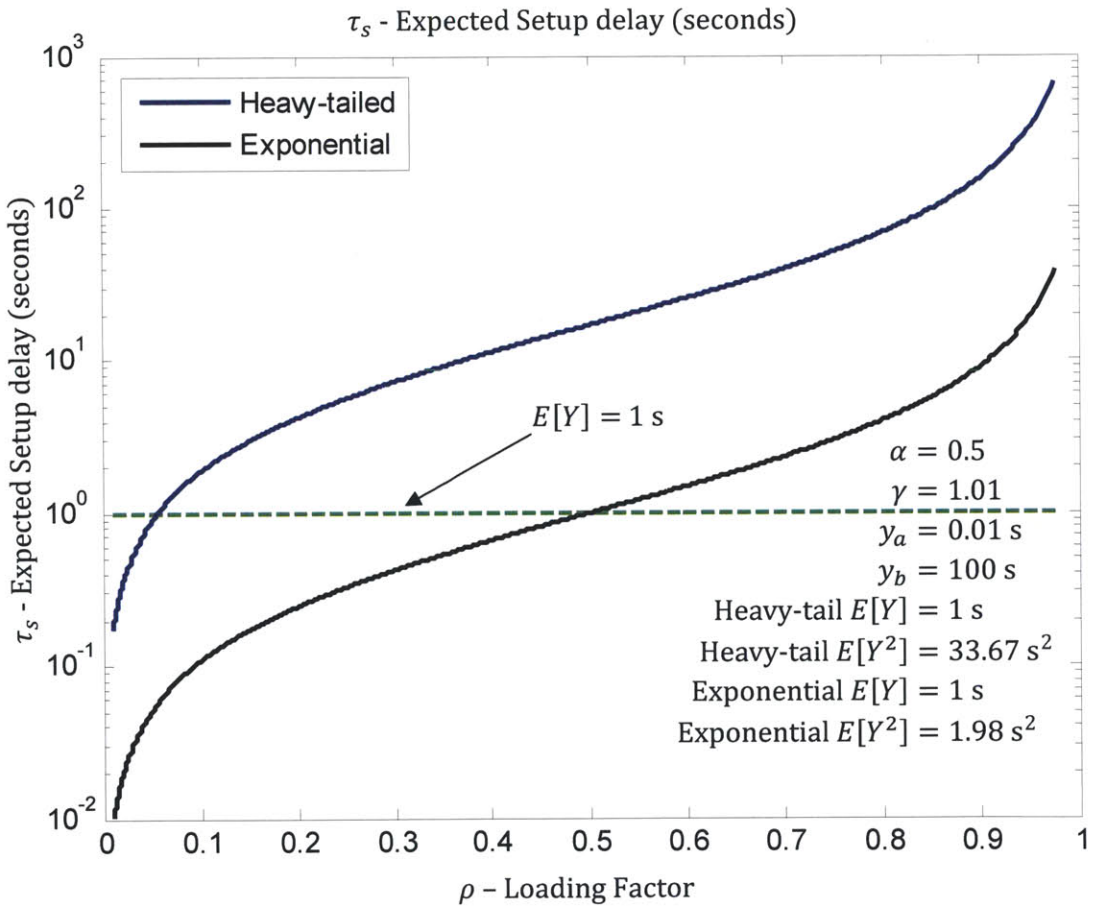


Figure 3-5: Lower bound for the expected setup delay versus network loading.

Chapter 4

4 OFS Transport Layer Protocol Performance Analysis – the Simple Transport Protocol (STP) and the Simple Transport Protocol with Interleaving (STPI)

4.1 Simple Transport Protocol (STP)

In this section, we find the throughput and delay performance of the Simple Transport Protocol (STP). In STP, an entire file is transmitted as one large frame. If an error is detected in a received file, the file is discarded and needs to be retransmitted. If retransmission is necessary, the source requests a new session from the scheduler and reflows the file. We assume that interleaving is not performed at the channel input and FEC can only correct for random errors. In STP, error-detection and retransmission are used to correct for burst errors [21].

Let C_q be the channel capacity defined in Chapter 3. Let H be the sum of the OFS header length and CRC overhead where

$$H = H_O + H_{CRC}. \tag{4.1}$$

4.1.1 Error Probabilities

In this section, we define two error probabilities: ψ, the probability of an erroneous file and ϕ_m, the probability of m failed transmissions. Let F be the file size in bits. The transmission size per session, W, is equal to the sum of the file size and overhead where

$$W = \frac{F + H}{C_q} + H_P. \tag{4.2}$$

We assume that FEC applied to random errors results in a very low residual BER of 10^{-12} or lower [18]. Let ζ be the residual BER. Let ϱ be the probability of an erroneous file due to random errors where

$$\varrho = 1 - (1 - \zeta)^W. \tag{4.3}$$

In this work, our focus is on combating burst errors, and we thus assume ϱ to be negligible compared with the probability of an erroneous file due to burst errors.

We assume that if an outage period occurs during data transmission, a file encounters an uncorrectable error. The probability of an erroneous file is equal to the probability of one or more burst errors during transmission. Let S_0 be the channel state at the start of data transmission. The probability of an erroneous file is:

$$\begin{aligned}\psi &= P(\text{outage}|S_0 = \text{non-outage})P(S_0 = \text{non-outage}) \\ &\quad + P(\text{outage}|S_0 = \text{outage})P(S_0 = \text{outage}) \\ &= \left(1 - e^{-\beta_1 \frac{W}{R}}\right)\frac{\beta_2}{\beta_1 + \beta_2} + (1)\frac{\beta_1}{\beta_1 + \beta_2} \\ &= 1 - \frac{\beta_2}{\beta_1 + \beta_2} e^{-\beta_1 \frac{W}{R}}.\end{aligned} \tag{4.4}$$

The probability of an erroneous file is plotted versus the file length in Figure 4-1. We see that as the file size increases, so does the probability of an erroneous file. The probability of m failed transmissions is

$$m \in [0, \infty)$$

$$\begin{aligned}\phi_m &= \psi^m \\ &= \left(1 - \frac{\beta_2}{\beta_1 + \beta_2} e^{-\frac{\beta_1 W}{R}}\right)^m.\end{aligned} \tag{4.5}$$

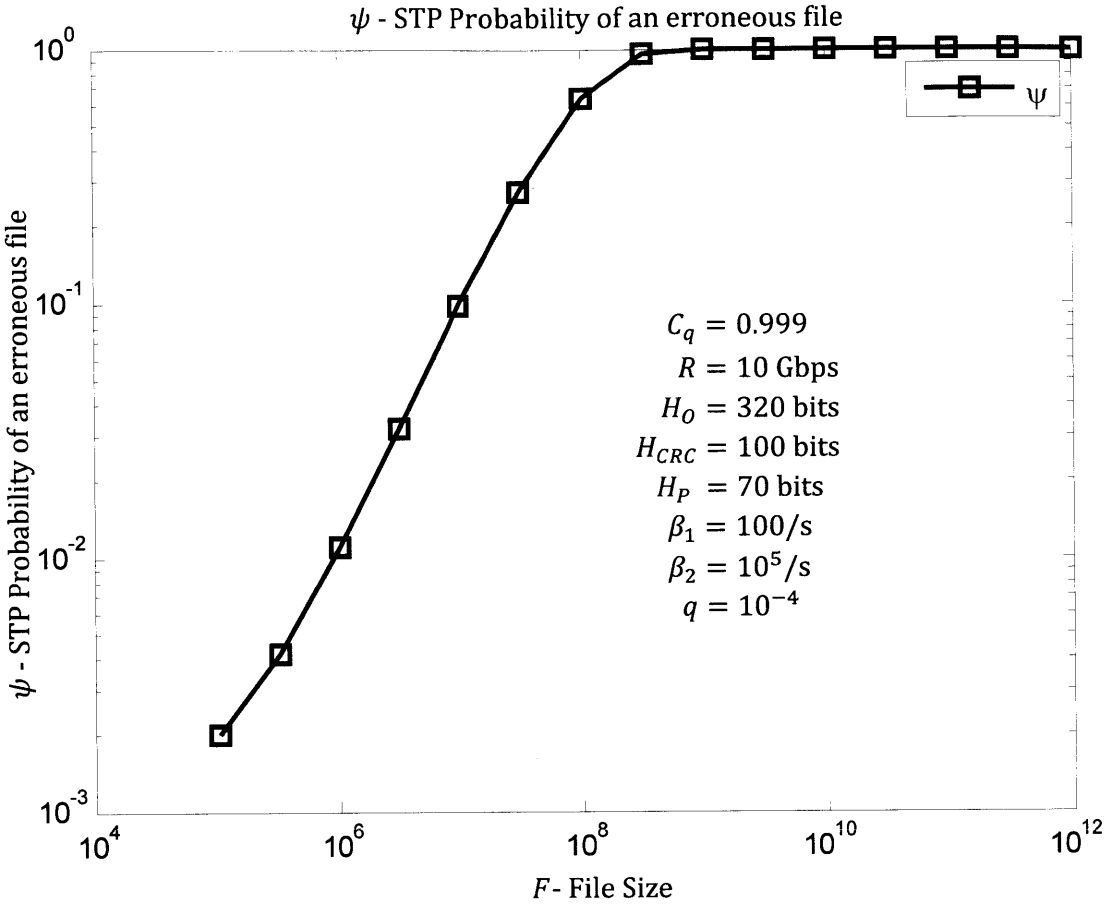


Figure 4-1: STP probability of an erroneous file vs. file length for one transmission.

4.1.2 Throughput

Let v be the required number of transmissions for a file to be received without error at the destination.

Let $\mathcal{G}$ be the expected throughput of STP where [25]

$$\begin{aligned}\mathcal{G} &= \frac{F}{W}\frac{1}{E[v]} \\ &= \frac{F}{W}\frac{1}{(1-\psi)\sum_{i=0}^{\infty}(i+1)\psi^i} \\ &= \frac{F}{W}(1-\psi).\end{aligned} \tag{4.6}$$

We substitute the expression in (4.2) into (4.6) to find the expected throughput as a function of the file size where

$$\mathcal{G} = \frac{FC_q}{F + H + H_P C_q} \frac{\beta_2}{\beta_1 + \beta_2} e^{-\beta_1 \frac{F+H+H_P C_q}{RC_q}}. \tag{4.7}$$

The expected throughput is shown in Figure 4-2. We notice that with increasing file sizes, the expected throughput of STP decreases.

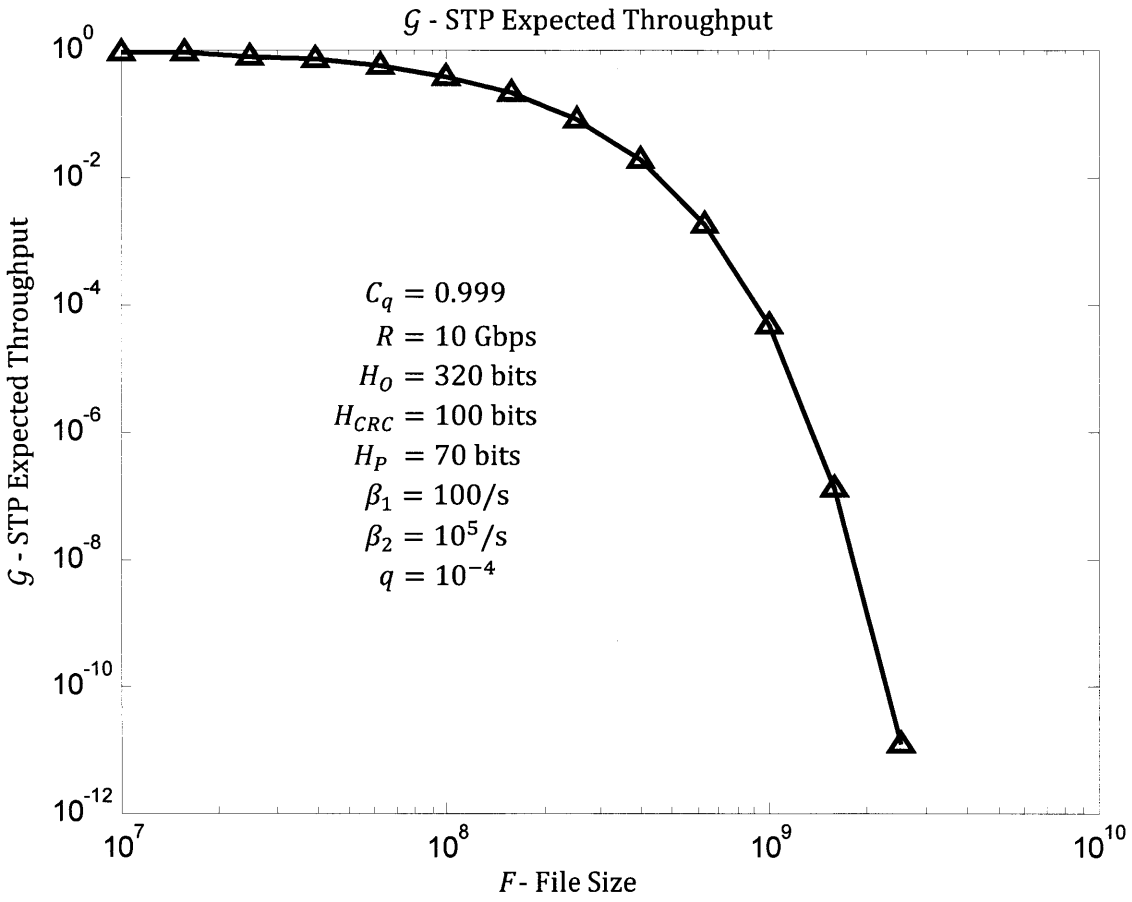


Figure 4-2: STP Expected Throughput

4.1.3 Delay

The delay of a session is equal to the sum of the setup delay and the session duration. The expected setup delay τ_s was found in Chapter 3. Let $\mathcal{Y}$ be the expected delay of a session where

$$\mathcal{Y} = \tau_s + \frac{W}{R}. \tag{4.8}$$

The total delay of a transaction is equal to the sum of the session delay plus retransmission delays. We assume that the propagation delay of sessions and session requests are small compared with the session delay and the setup delay. Thus, we neglect the propagation delay in our delay analysis [5]. A retransmitted file is the same size as the initial file. The probability that a retransmitted file experiences an uncorrectable error is also the same as for the initial file. Let $\mathcal{T}$ be the total expected delay where

$$\begin{aligned} E[\mathcal{T}] &= \sum_{m=0}^{\infty} \mathcal{Y}\,\psi^m \\ &= \frac{\mathcal{Y}}{1-\psi} \\ &= \left(\tau_s + \frac{F+H}{RC_q} + \frac{H_P}{R}\right)\left(1+\frac{\beta_1}{\beta_2}\right)e^{\beta_1\frac{F+H+H_PC_q}{RC_q}}. \end{aligned} \tag{4.9}$$

We plot the total expected delay in Figure 4-3. We see that with increasing file sizes, delay performance suffers. Thus, if the code does not correct for any burst errors, the longer the file size the more likely that a burst error will occur and both the throughput and delay performance are bad. This suggests we need a coding scheme that can correct for burst errors and we will treat one example in the next section (STPI).

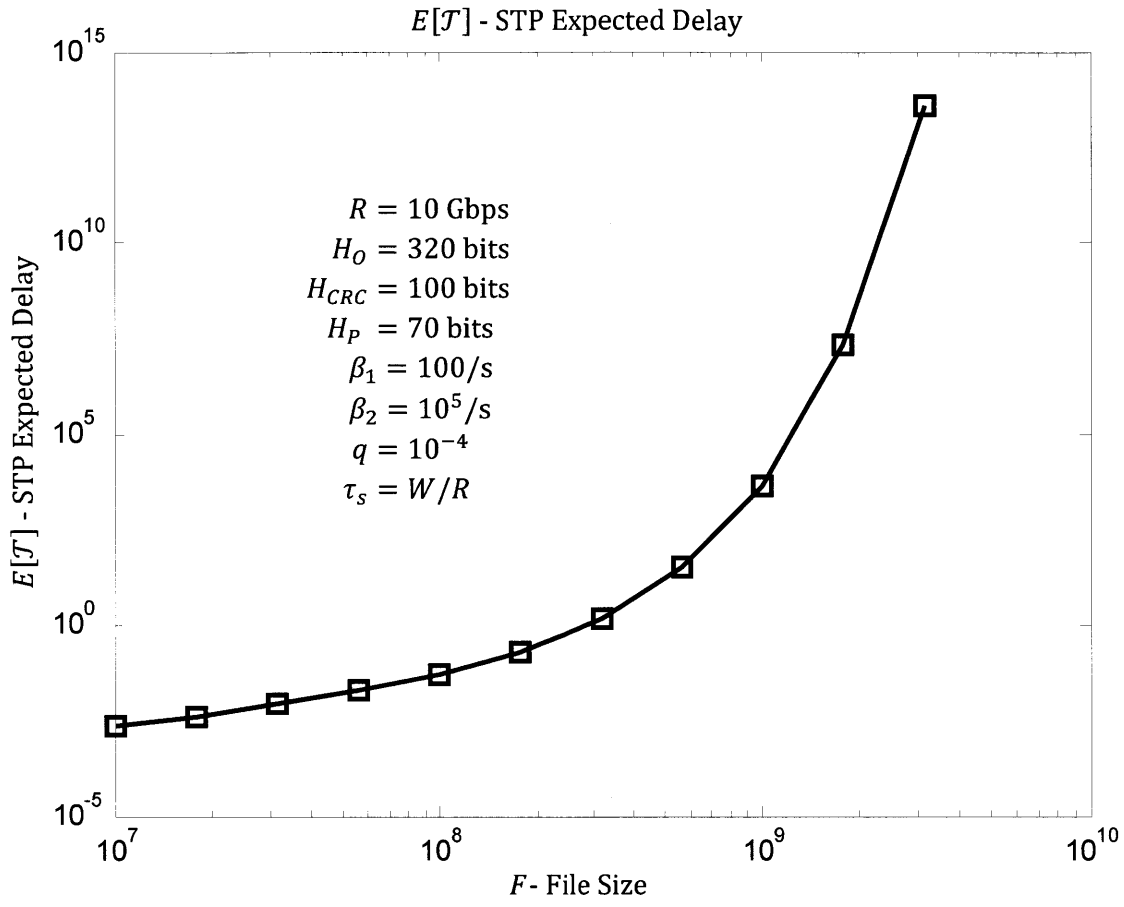


Figure 4-3: Expected STP Delay versus file size.

4.2 Simple Transport Protocol with Interleaving (STPI)

In this section, we find the throughput and delay performance of the Simple Transport Protocol with Interleaving (STPI). In STPI, an entire file is transmitted as one large frame. If an error is detected in a received file, the file is discarded and needs to be retransmitted. If retransmission is necessary, the source requests a new session from the scheduler and reflows the file. In STPI, we assume that interleaving is performed at the channel input. Retransmission combined with FEC is used to correct for both random and burst errors [21].

4.2.1 Error Probabilities

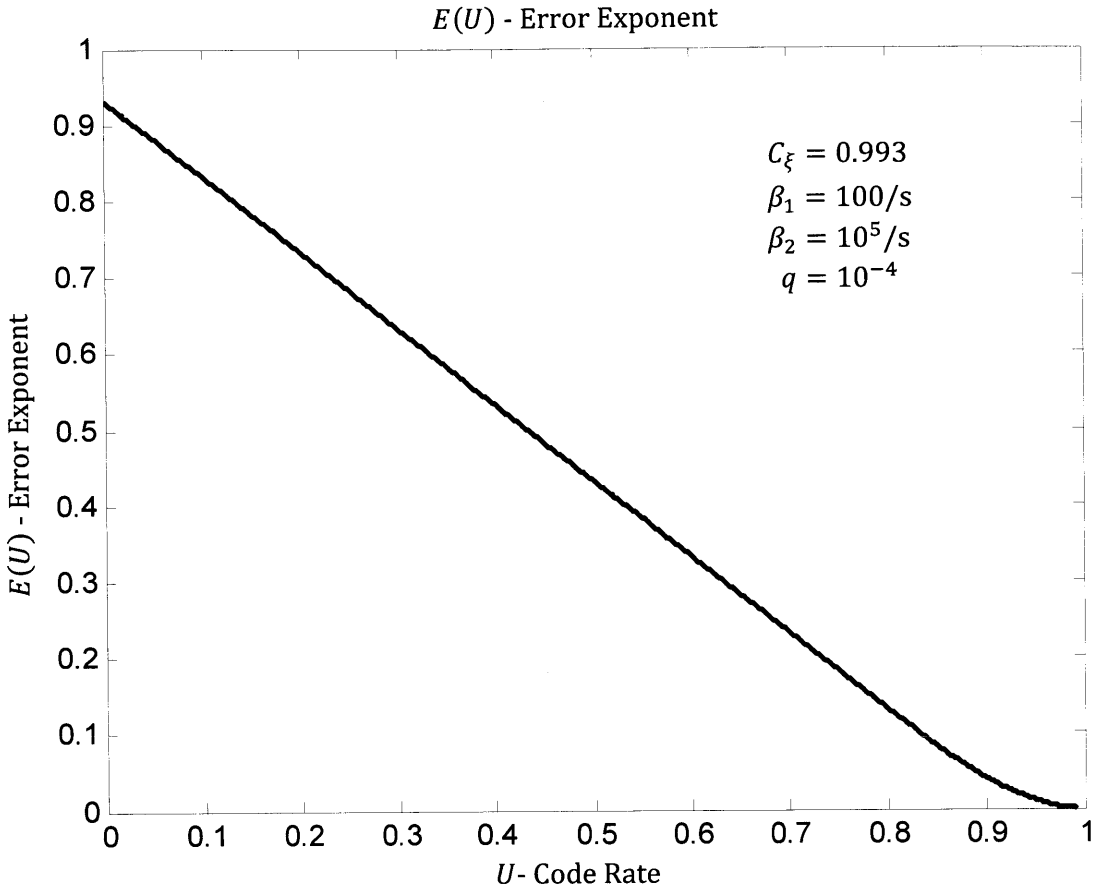


Figure 4-4 Error exponent vs. code rate

In STPI, a file requires retransmission if a decoding error occurs. In this section, we find an upper bound on the average decoding error probability of a file. We assume that the burst channel can be converted into an essentially memoryless channel with the use of interleaving [21]. Let U be the code rate and ξ be the error probability defined in Chapter 3. Let N be the length of the file plus overhead, where

$$N = \frac{F + H}{U} + H_P. \tag{4.10}$$

Let ψ_I be the probability of a decoding error, where [21]

$$\psi_I \leq 2^{-NE(U)} \tag{4.11}$$

and

$$\begin{aligned} 0 \leq \omega \leq 1 \\ E(U) = \max_{\omega}(E_0(\omega) - \omega U) \\ E_0(\omega) = \omega - (1+\omega)\log_2\left(\xi^{\frac{1}{1+\omega}} + (1-\xi)^{\frac{1}{1+\omega}}\right). \end{aligned} \tag{4.12}$$

$E(U)$ is sometimes called the reliability function and is the tightest exponential bound as a function of the frame size. As a function of U, $E(U)$ can be expressed in parametric form. Let π be related to ω through [21]

$$\pi = \frac{\xi^{\frac{1}{1+\omega}}}{\xi^{\frac{1}{1+\omega}} + (1-\xi)^{\frac{1}{1+\omega}}}. \tag{4.13}$$

$$\text{For } U \geq 1 - h\left(\frac{\sqrt{\xi}}{\sqrt{\xi} + \sqrt{1-\xi}}\right): \tag{4.14}$$

$$\begin{aligned} U &= 1 - h(\pi) \\ E(U) &= -\pi \log_2 \xi - (1-\pi)\log_2(1-\xi) - h(\pi). \end{aligned}$$

$$\text{For } U < 1 - h\left(\frac{\sqrt{\xi}}{\sqrt{\xi} + \sqrt{1-\xi}}\right): \tag{4.15}$$

$$E(U) = 1 - 2\log_2\left(\sqrt{\xi} + \sqrt{1-\xi}\right) - R.$$

We plot the error exponent, $E(U)$ versus the code rate in Figure 4-4. If the code rate is less than the capacity of the channel, by choosing an appropriate code, the error probability approaches 0 with increasing file sizes [21].

4.2.2 Throughput

Let $\mathcal{G}_I$ be the expected throughput. The derivation of $\mathcal{G}_I$ follows from the derivation of $\mathcal{G}$ in (4.6) where

$$\begin{aligned} \mathcal{G}_I &= \frac{F}{N}(1-\psi_I) \\ &= \frac{FU}{F+H+H_PU}(1-\psi_I). \end{aligned} \tag{4.16}$$

We note that the error probability ψ_I and code rate U should satisfy:

$$\begin{aligned} 0 \le \psi_I \le 1 \\ 0 \le U < C_\xi. \end{aligned} \tag{4.17}$$

We find upper and lower bounds for the expected STPI throughput. A lower bound is found by substituting (4.11) into (4.16) and maximizing over U. An upper bound is found in three steps. First, in the numerator, we upper bound U by the channel capacity C_ξ. Second, in the denominator, we assume that $F \gg H_P$. Third, we notice that the error probability approaches 0 with increasing file sizes. Thus we lower bound ψ_I with 0. The STPI throughput bounds are plotted in Figure 4-5. The expected throughput is bounded by

$$\max_{0 \le U < C_\xi} \frac{FU}{F+H+H_PU}\left(1-2^{-NE(U)}\right) \le \mathcal{G}_I < \frac{FC_\xi}{F+H}. \tag{4.18}$$

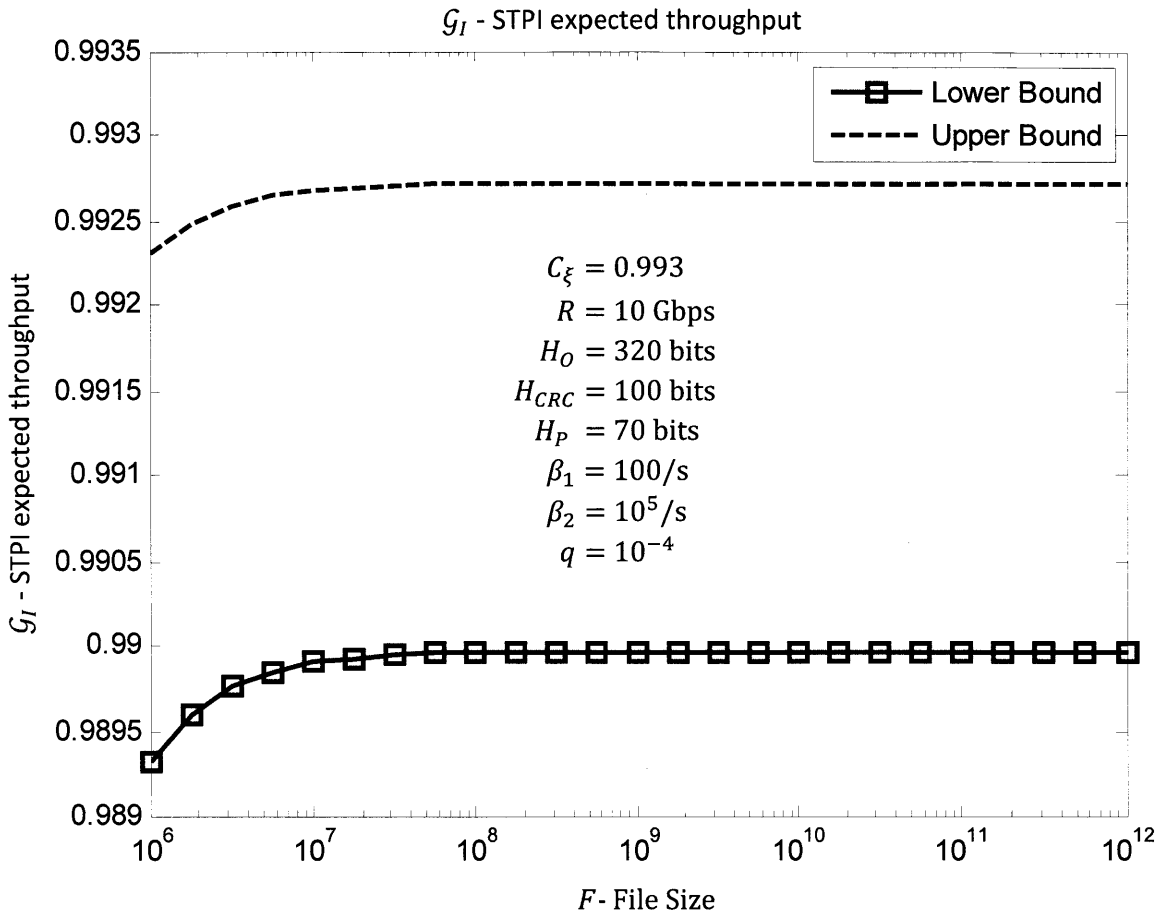


Figure 4-5: STPI expected throughput upper and lower bounds. The lower bound is from the reliability function and should be an excellent approximation.

4.2.3 Delay

Let $\mathcal{Y}_I$ be the expected delay of a STPI session where

$$\mathcal{Y}_I = \tau_s + \frac{N}{R}. \tag{4.19}$$

The total delay is equal to the sum of the initial session delay plus retransmission delays.

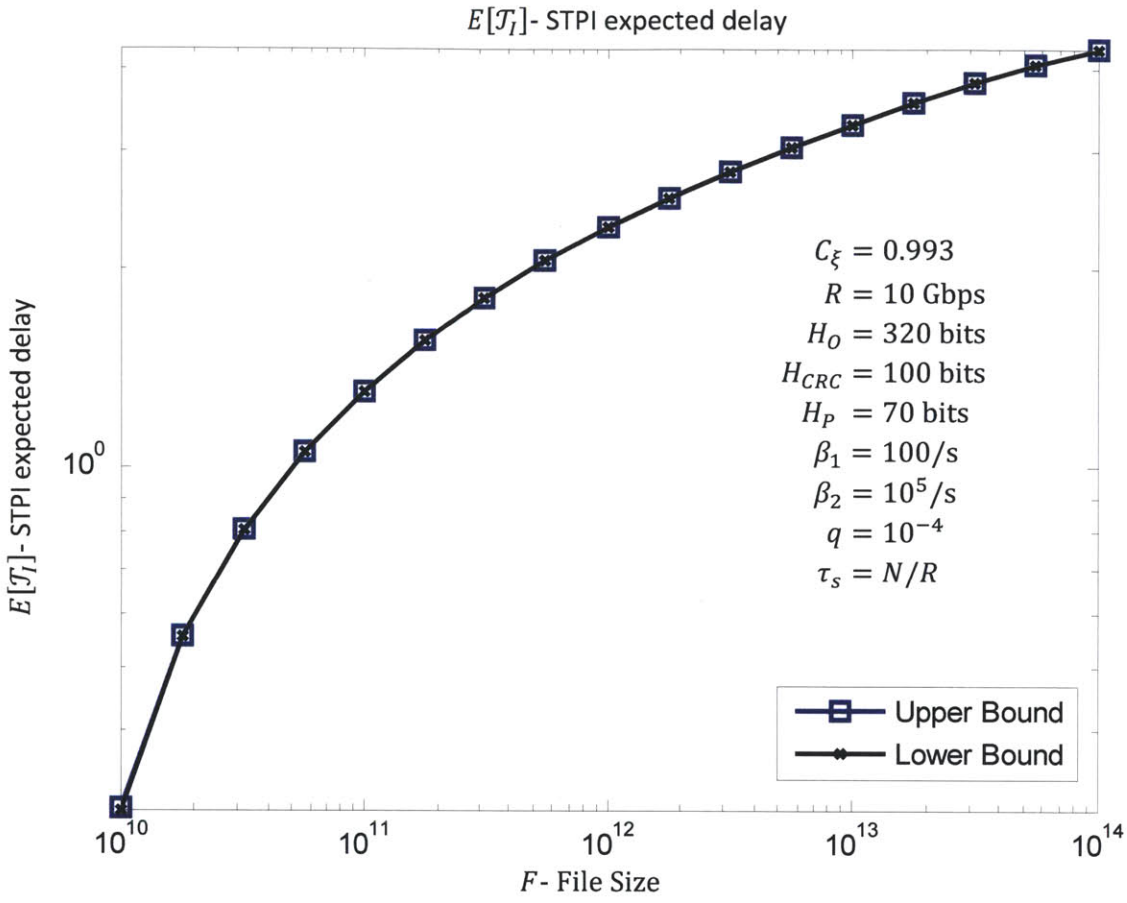


Figure 4-6: STPI expected delay upper and lower bounds.

Let $\mathcal{T}_I$ be the total delay where

$$\begin{aligned} E[\mathcal{T}_I] &= \sum_{m=0}^{\infty} \mathcal{Y}_I \psi_I{}^m \\ &= \frac{\mathcal{Y}_I}{1-\psi_I} \\ &= \frac{\tau_s + \frac{F+H}{UR} + \frac{H_P}{R}}{1-\psi_I}. \end{aligned} \tag{4.20}$$

We find upper and lower bounds for the expected STPI delay. An upper bound is found by substituting (4.11) into (4.20) and minimizing over U. A lower bound is found by upper bounding U with C_ξ and lower bounding ψ_I with 0. The upper bound comes from the reliability function and should

be exponentially tight and provide an excellent approximation of $E[\mathcal{T}_I]$. We plot the STPI delay bounds in Figure 4-6 where

$$\tau_s + \frac{F+H}{C_\xi R} + \frac{H_P}{R} < E[\mathcal{T}_I] \leq \min_{0 \leq U < C_\xi} \frac{\tau_s + \frac{F+H}{UR} + \frac{H_P}{R}}{1 - 2^{-NE(U)}}. \tag{4.21}$$

The upper and the lower bounds are very close to each other and any one can be used as the approximation for the delay. Except for light loading, the setup time is at least one or more session transmission times and dominates the delay.

Chapter 5

5 OFS Transport Layer Protocol Performance Analysis – the Transport Protocol with Framing (TPF) and the Transport Protocol with Framing and Interleaving (TPFI)

5.1 Transport Protocol with Framing (TPF)

In this section, we find the throughput and delay performance of the Transport Protocol with Framing (TPF). In TPF, a large file is segmented into frames at the source and reassembled at the destination. We assume that interleaving is not performed at the transmitter and FEC is only used to correct for random errors but not the burst errors. ARQ allows frames that are received with errors to be retransmitted. Only erroneous frames are retransmitted.

5.1.1 Error Probabilities

We refer to the first attempt to send a file as the "initial" transmission. Subsequent sessions are used for frame retransmissions. In this section, we define two error probabilities: p, the probability of an erroneous frame and θ, the probability of a failed initial transmission. Let D be the number of message bits per frame. Let H and H_p be the overhead defined in Chapter 3. Let C_q be the channel capacity defined in Chapter 3. Let L be the total number of bits per frame (message plus overhead) where

$$L = \frac{D + H}{C_q} + H_P. \tag{5.1}$$

We assume a frame that experiences a burst error due to an outage period cannot be corrected with FEC and is received in error. The probability that a frame is received in error is:

$$p = 1 - P(0 \text{ bursts} | S_0 = \text{ non–outage}) P(S_0 = \text{ non–outage})$$
$$= 1 - \frac{\beta_2}{\beta_1 + \beta_2} e^{-\frac{\beta_1 L}{R}}. \tag{5.2}$$

In TPF, a file is segmented into n sequential frames, where $n = \lceil F/D \rceil$. Practically, the n^{th} frame may be smaller than the other $n - 1$ frames. In our analysis, however, we assume that all n frames are of the same size. A successful transmission results if all n frames are received without error at the destination. The source sends all n frames in order. The probability that a file is received in error is equal to the probability that one or more frames are received in error. Thus, the probability of a failed initial transmission is

$$\theta = \sum_{k=0}^{n-1} \binom{n}{k} (1-p)^k p^{n-k} \tag{5.3}$$
$$= 1 - (1-p)^n.$$

As shown in Chapter 3, we assume in the operating region of interests that network loading is large enough such that the setup delay is a significant fraction of the total delay. If a file is received in error, the source requests a new session from the scheduler to retransmit outstanding frames. If retransmission is necessary, a large setup delay associated with acquiring a new session may lead to a significant increase in the total delay to transmit a file.

In TPF, users have the option to reserve additional time per session to allow for frame retransmissions. In this mode, the transmitter sends all n frames sequentially followed by frame retransmissions. Users request retransmissions immediately after an erroneous frame is detected at the receiver. The forward link is assumed to be idle while waiting for the last retransmission request. We assume intermediary retransmission requests are sent on the reverse link while the forward link is not idle. Thus, we only consider the roundtrip propagation delay of the last outstanding frame to contribute to the total delay. Let Δ be the maximum fraction of retransmission frames to the total number of frames sent in a session. Let δ be additional time per session allowed for frame retransmissions where

$$\delta = \frac{\lfloor n\Delta \rfloor}{R}. \tag{5.4}$$

Let τ_p be the round trip propagation delay. To account for the roundtrip propagation delay of the last outstanding frame, we assume that the total additional time reserved is $\delta + \tau_p$. The probability of a failed initial transmission is

$$\theta = \sum_{k=0}^{n-1} \binom{\lfloor n(1+\Delta) \rfloor}{k} (1-p)^k p^{\lfloor n(1+\Delta) \rfloor - k}. \tag{5.5}$$

The probability of a failed initial transmission in (5.3) is a degenerate case of (5.5) for Δ equal to zero. The sum in (5.5) is the cumulative distribution function (CDF) of a binomial distribution with mean

$$\mu = \lfloor n(1+\Delta) \rfloor (1-p) \tag{5.6}$$

and variance

$$\sigma^2 = \lfloor n(1+\Delta) \rfloor (1-p)p. \tag{5.7}$$

We can relax the integer constraint on n and $n\Delta$ and approximate the sum in (5.5) with the normal CDF where

$$\begin{aligned} \theta &\cong \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{n-1} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ &= \Phi\left(\frac{n-1-\mu}{\sigma}\right) \\ &= \Phi^{\mathrm{c}}\left(\frac{\mu-(n-1)}{\sigma}\right). \end{aligned} \tag{5.8}$$

Furthermore, (5.8) can be bounded by [26]

$$\begin{aligned} \theta &> 1 - \frac{1}{2} e^{-\frac{(n-1-\mu)^2}{2\sigma^2}}, \quad for\, n-1 > \mu, \\ \theta &< \frac{1}{2} e^{-\frac{(\mu-n+1)^2}{2\sigma^2}}, \quad for\, n-1 \leq \mu. \end{aligned} \tag{5.9}$$

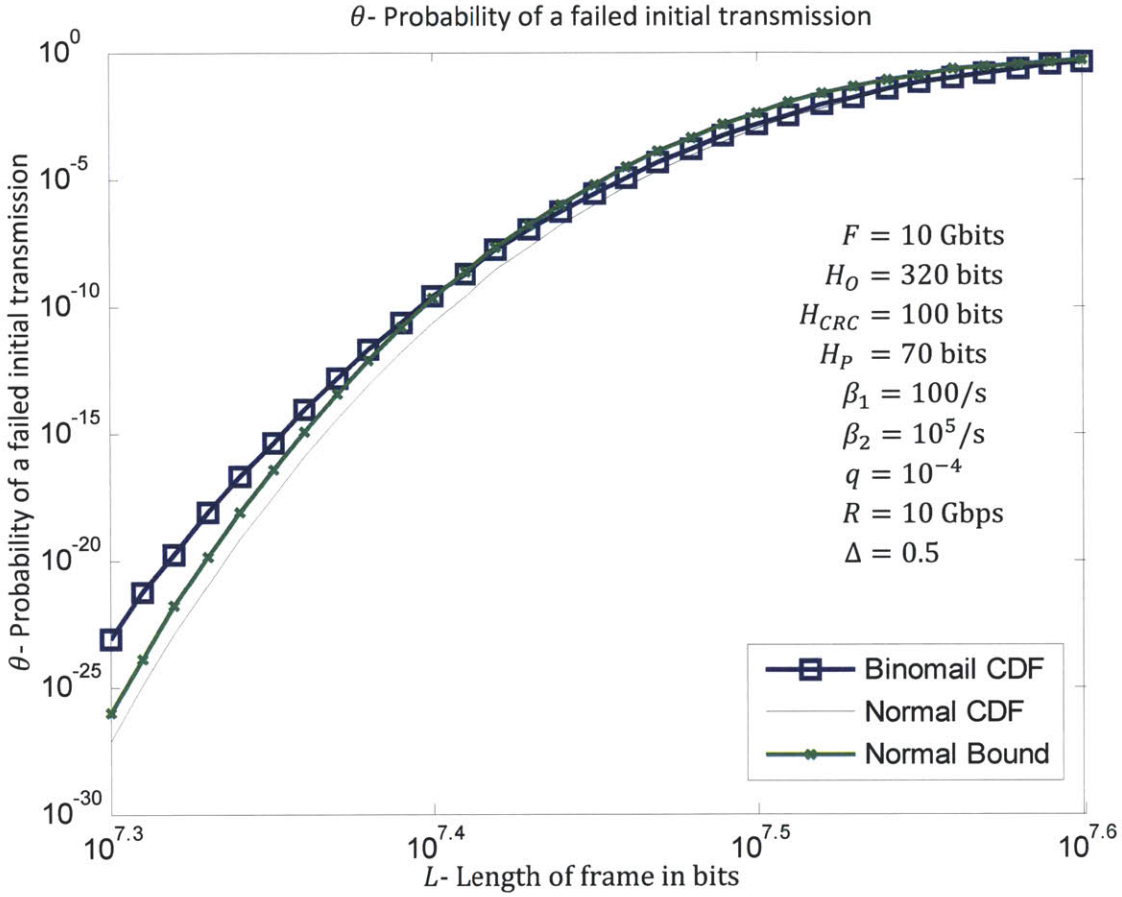


Figure 5-1: TPF probability of a failed initial transmission vs. frame length. "Binomial CDF" corresponds to the probability of a failed initial transmission found in (5.5). "Normal CDF" corresponds to the approximation for the probability of a failed initial transmission found in (5.8). "Normal CDF" corresponds to the approximation for the probability of a failed initial transmission found in (5.9).

We plot the probability of a failed initial transmission in Figure 5-1.

5.1.2 Performance Optimization – Throughput

We assume that the probability of an erroneous frame is independent of future and previous transmitted frames. We model the channel as a binary erasure channel (BEC) and consider the transmission of frames (rather than bits). The capacity of the binary erasure channel is $1 - p$ [20]. An erasure corresponds to a frame received in error. Thus, we assume that frames are lost with IID probability p. Let η be the expected throughput where

$$\eta = (1-p)\frac{D}{L} \tag{5.10}$$
$$= \frac{\beta_2}{\beta_1+\beta_2} e^{-\frac{\beta_1 L}{R}} \frac{(L-H_P)C_q - H}{L}.$$

Equation (5.10) shows that the expected throughput is a function of the overhead, outage parameters, transmission rate, and frame length. The frame overhead, outage parameters and transmission rate are network or link dependant parameters. The frame length, however, is an adjustable parameter that can be set by the OFS transport layer. In the next section, we find the optimal frame length that maximizes the expected throughput.

5.1.2.1 Optimal Frame Length

We find the optimal frame length that achieves the maximum expected throughput. As the frame length decreases, the frame overhead becomes a significant fraction of the total frame length. For very small frames, throughput performance suffers as the link is mostly sending frame overhead. As the frame length increases, the probability that a frame experiences an uncorrectable burst error also increases. For very large frames, throughput performance suffers as the link is occupied mostly for sending erroneous and retransmission frames. The optimal frame length lies in the region where the probability of an erroneous frame is low and the message length is much larger than the overhead length. We substitute (5.1) into (5.10) to find the expected throughput as a function of the message size where

$$\eta = \frac{DC_q}{D+H+C_qH_P}\frac{\beta_2}{\beta_1+\beta_2} e^{-\beta_1\frac{D+H+H_PC_q}{C_qR}}. \tag{5.11}$$

We take the derivative of η with respect to D to find the optimal message length where

$$\frac{\partial \eta}{\partial D} = \frac{\beta_2}{\beta_1+\beta_2} e^{-\beta_1\frac{D+H+H_PC_q}{C_qR}} \left(\frac{C_q(H+H_PC_q)R - \beta_1 D(D+H+H_PC_q)}{R(D+H+H_PC_q)^2} \right) = 0. \tag{5.12}$$

We solve (5.12) for the optimal message length. Let D^* be the length of the message at the optimal frame length and L^* be the optimal frame length where

$$D^* = \sqrt{(H + C_q H_P)\left(\frac{H + C_q H_P}{4} + \frac{C_q R}{\beta_1}\right)} - \frac{H + C_q H_P}{2}$$
$$L^* = \frac{1}{C_q}\sqrt{(H + C_q H_P)\left(\frac{H + C_q H_P}{4} + \frac{C_q R}{\beta_1}\right)} + \frac{1}{2}\left(H_P + \frac{H}{C_q}\right). \quad (5.13)$$

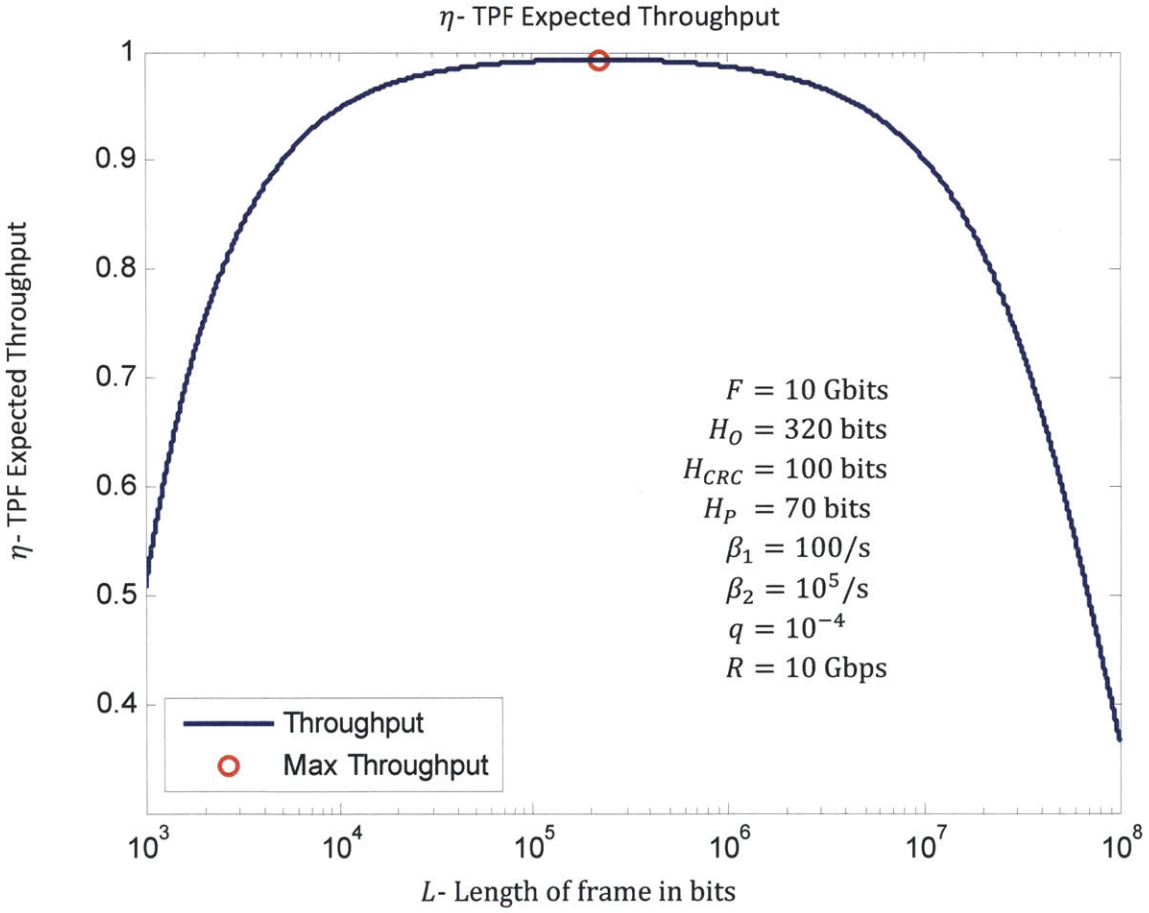


Figure 5-2: TPF throughput as a function of frame length. The red circle indicates the maximum throughput.

Let η^* be the maximum expected throughput where

$$\eta^* = \frac{\mathrm{D}^*}{L^*}\frac{\beta_2}{\beta_1 + \beta_2} e^{-\frac{\beta_1 L^*}{R}}. \quad (5.14)$$

We plot the TPF expected throughput as a function of frame length in Figure 5-2.

5.1.2.2 Practical Frame Length

Users who wish to avoid large segmentation and reassembly overhead can choose to partition a file into frames of length larger than the optimal frame length. We notice that the derivative of the throughput expression near the maximum is small. Thus, users can choose a larger frame size and experience a small throughput performance penalty in exchange for less segmentation and reassembly overhead. In this section, we find the practical frame length, the frame length that results in an expected throughput ε fraction away from the optimal expected throughput.

We assume that for frame lengths greater than the optimal frame length (including the practical frame length), the message length is much larger than the overhead size. Let D_ε be the length of the message at the practical frame length. Thus, we assume that $D_\varepsilon \gg H$ and $D_\varepsilon \gg H_P$. We can approximate the expected throughput at the practical frame length as

$$\eta \cong \frac{C_q \beta_2}{\beta_1 + \beta_2} e^{-\beta_1 \frac{D_\varepsilon + H + H_P C_q}{C_q R}}. \tag{5.15}$$

Let η_ε be the expected throughput ε away from the optimal expected throughput. At the practical frame length,

$$\eta_\varepsilon = (1 - \varepsilon)\eta^*. \tag{5.16}$$

We substitute (5.15) into (5.16) and solve for the practical frame length where

$$\frac{C_q \beta_2}{\beta_1 + \beta_2} e^{-\beta_1 \frac{D_\varepsilon + H + H_P C_q}{C_q R}} \cong (1 - \varepsilon)\eta^*. \tag{5.17}$$

Let L_ε be the practical frame length. Solving, (5.17) we find the practical message length and practical frame length where

$$D_\varepsilon \cong \frac{C_q R}{\beta_1} ln\left(\frac{\frac{\beta_2 C_q}{\beta_1+\beta_2}}{(1-\varepsilon)\eta^*}\right) - C_q H_P - H$$
$$L_\varepsilon \cong \frac{R}{\beta_1} ln\left(\frac{\frac{\beta_2 C_q}{\beta_1+\beta_2}}{(1-\varepsilon)\eta^*}\right). \tag{5.18}$$

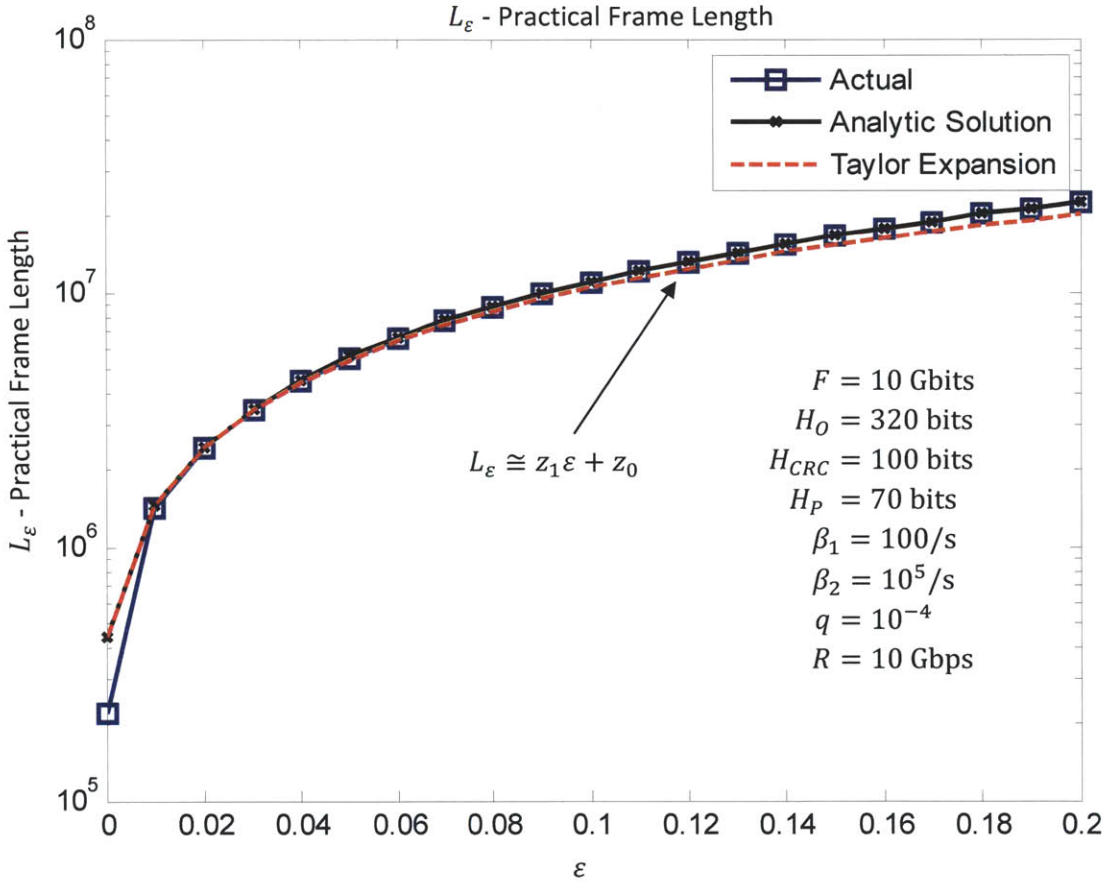


Figure 5-3: TPF practical frame length vs. ε . "Actual" corresponds to numerically solving for the practical frame length in (5.16). "Analytic Solution" corresponds to the expression for the practical frame length in (5.18). "Taylor Expansion" corresponds to the expression for the practical frame length in (5.20).

Using the first term for the Taylor series expansion $\ln(1-x) \cong -x$, we further approximate the maximum message length and practical frame length where

$$
\begin{aligned}
D_\varepsilon &\cong u_1\varepsilon + u_0 \\
u_1 &= \frac{C_q R}{\beta_1} \\
u_0 &= \frac{C_q R}{\beta_1} ln\left(\frac{C_q}{\eta^*}\frac{\beta_2}{\beta_1+\beta_2}\right) - C_q H_P - H
\end{aligned}
\tag{5.19}
$$

and

$$
\begin{aligned}
L_\varepsilon &\cong z_1\varepsilon + z_0 \\
z_1 &= \frac{R}{\beta_1} \\
z_0 &= \frac{R}{\beta_1} ln\left(\frac{C_q}{\eta^*}\frac{\beta_2}{\beta_1+\beta_2}\right).
\end{aligned}
\tag{5.20}
$$

We plot the practical frame length as a function of ε in Figure 5-3. In the region of interests, L_ε is linear in ε.

5.1.3 Performance Optimization – Delay

In this section, we find the minimum total expected delay. We find the total expected delay under two different assumptions for the setup and propagation delay: no setup and propagation delay and nonzero setup and propagation delay. Let τ_s be the expected setup delay defined in Chapter 3. Let T be the total delay and $E[T]$ be the expected total delay to send a file.

The expected total delay for no setup and propagation delay in given (5.21). The derivation of (5.21) can be found in Appendix A.1. We notice that for this case, the optimal frame length found in Section 5.1.2.1 also results in the minimum delay. We plot the total expected delay in Figure 5-4.

$\tau_s = 0$ **and** $\tau_p = 0$	
$E[T] = \frac{\frac{F}{D}\frac{L}{R}}{1-p} = \frac{F}{R}\frac{1}{\eta}$	(5.21)

Table 5-1: TPF total expected delay for no setup and propagation delay.

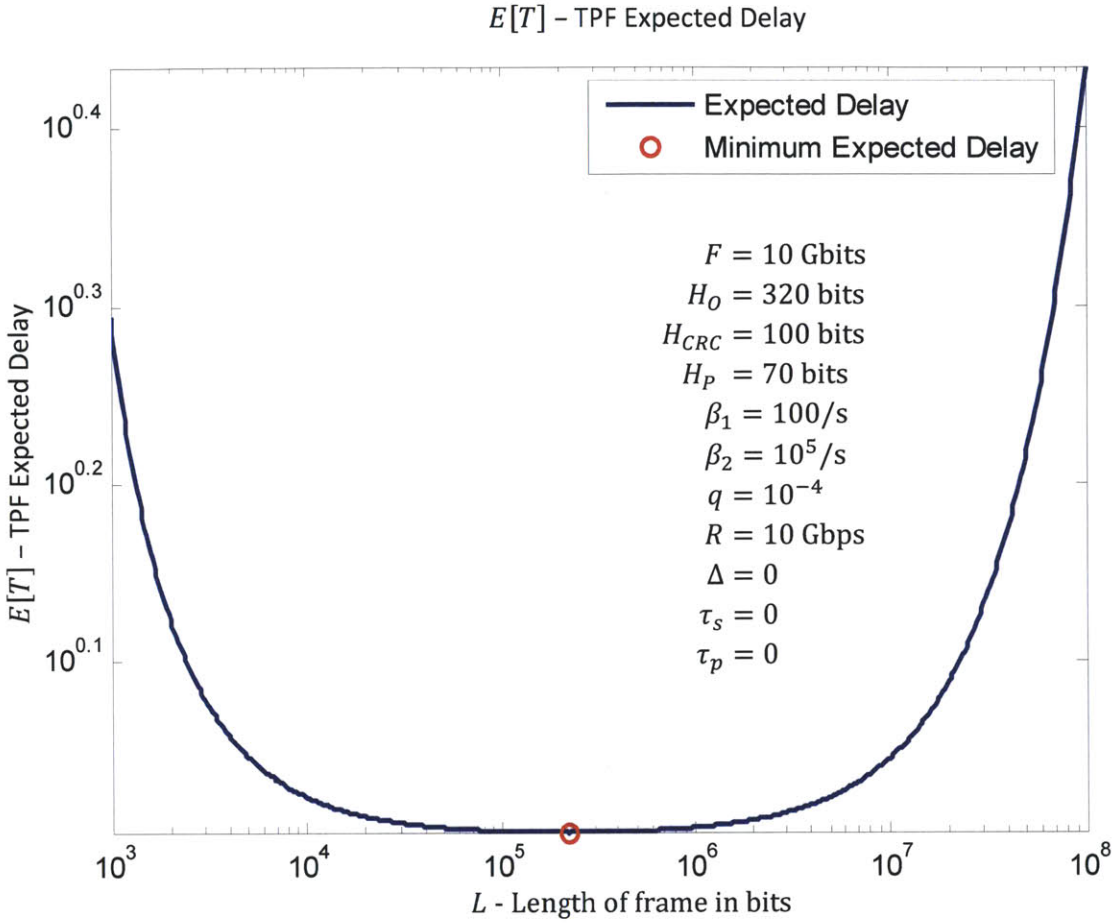


Figure 5-4: Expected TPF Delay (no setup and propagation delay)

The expected total delay for nonzero setup and propagation delay is given in (5.22). The derivation of (5.22) can be found in Appendix A.2. Let $p_{k\backslash x}$ be the probability that k outstanding frames remain after session termination out of x initial frames sent.

$\tau_s \neq 0$ **and** $\tau_p \neq 0$	
$p_{i\backslash n} = \binom{\lfloor n(1+\Delta) \rfloor}{n-i} (1-p)^{n-i} p^{\lfloor n(1+\Delta) \rfloor - n + i}$	(5.22)
$p_{n\backslash n} = \binom{\lfloor n(1+\Delta) \rfloor}{0} (1-p)^{0} p^{\lfloor n(1+\Delta) \rfloor} = p^{\lfloor n(1+\Delta) \rfloor}$	
$E[T] = \frac{\lfloor n(1+\Delta) \rfloor \frac{L}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{i\backslash n} E_i[T]}{1 - p_{n\backslash n}}$	

Table 5-2: TPF total expected delay for nonzero setup and propagation delay.

Let $E[T^*]$ be the optimal expected delay. The total delay is a function of both the additional session reservation duration and the frame length. To find the optimal expected delay, we must optimize both the session reservation duration and the frame length. We find the optimal additional session reservation in Section 5.1.3.1 and then find the optimal frame length in Section 5.1.3.2.

5.1.3.1 Optimal Additional Session Reservation

In TPF, users have the option to request additional time per session for retransmissions to avoid the need for additional session requests and additional setup delays. When the setup and propagation delay is zero, users do not experience an additional delay penalty when requesting a new session for frame retransmissions. Therefore, $\Delta = 0$ is optimal ($\Delta^* = 0$) as shown in Figure 5-5.

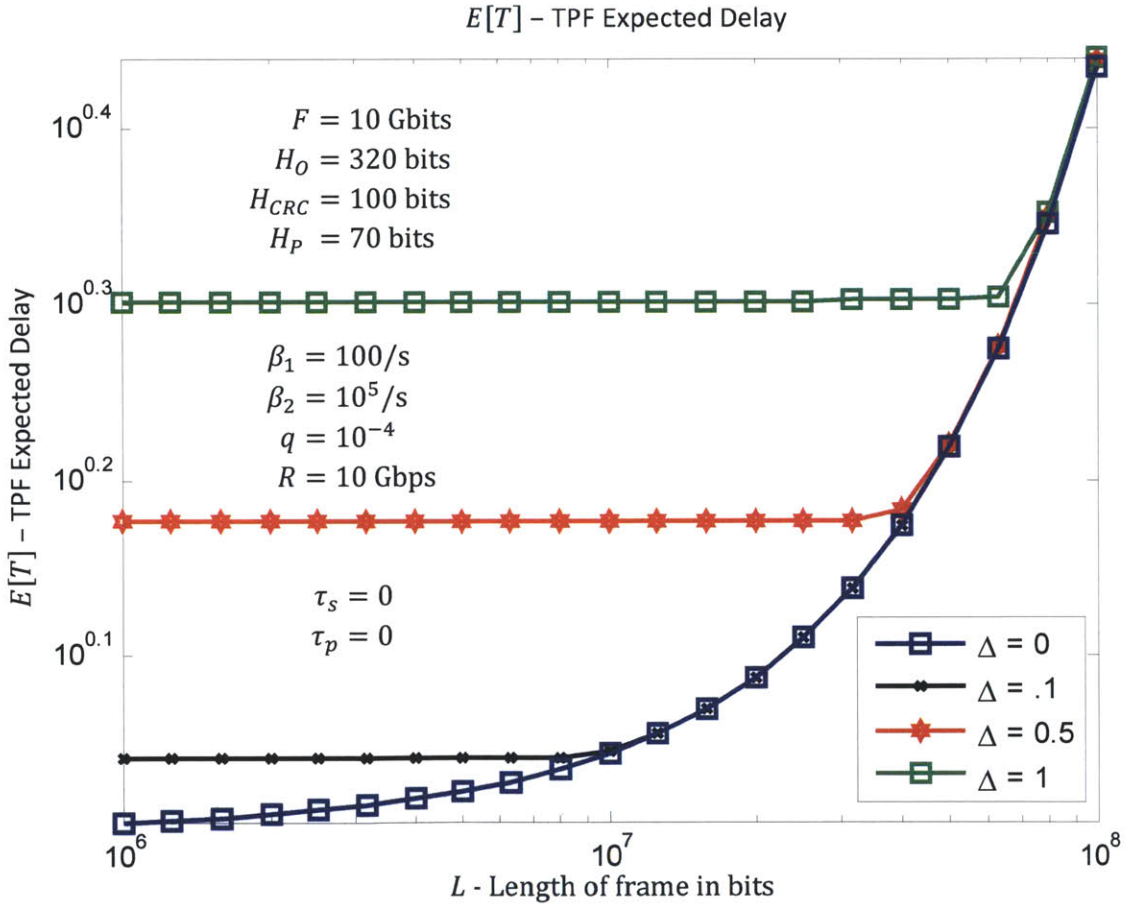


Figure 5-5: TPF delay vs. Δ when $\tau_s = 0$ and $\tau_p = 0$. Δ = 0 provides the optimal delay performance.

By plotting exact solutions in Figure 5-6, we show that when the setup delay is large, the optimal additional session reservation duration Δ, will be nonzero. On the other hand if the additional reserved duration is too large, transmission completes before the session expires and network resources are wasted. If the additional reserved duration it too short, transmission does not complete before the session expires and a new session request is necessary. We observe that if the network is moderately or heavily loaded, when additional sessions are required to complete transmission, the setup delay can dominate in the total delay.

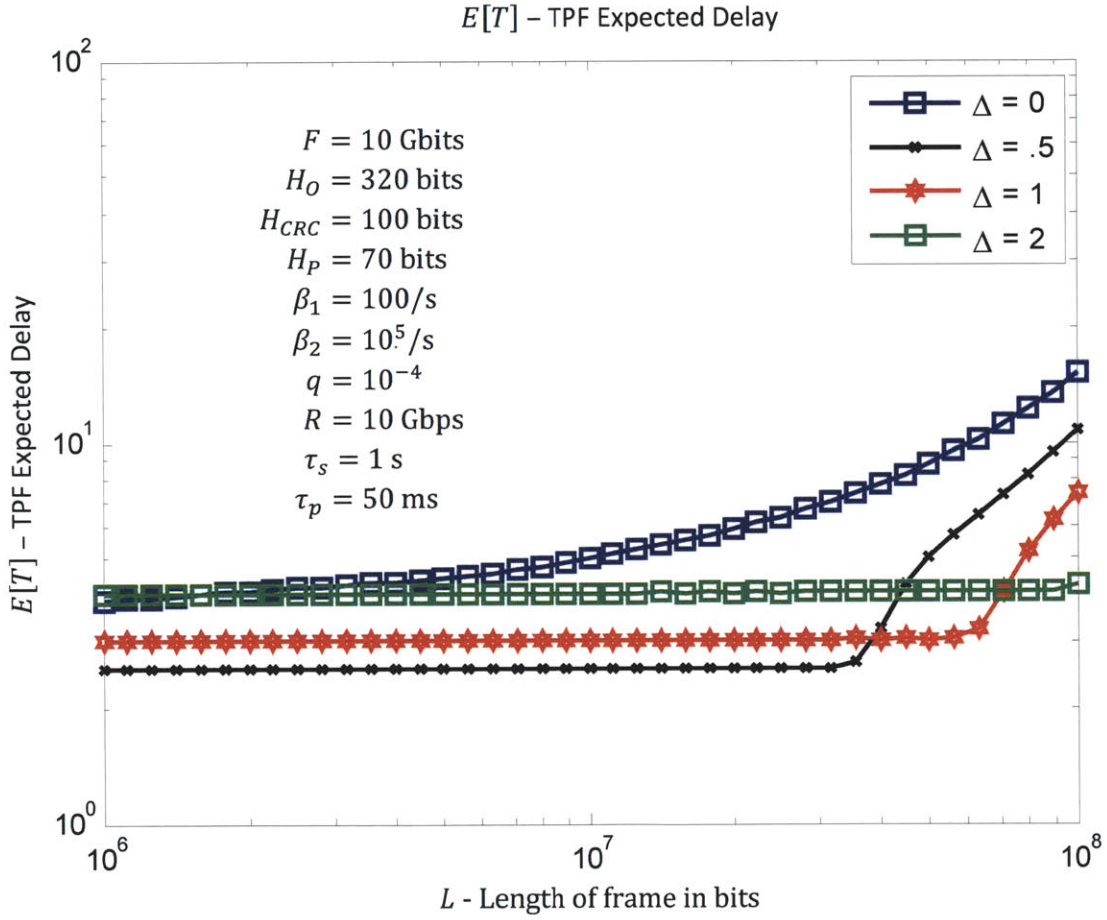


Figure 5-6: TPF delay over different values for Δ (non-zero setup delay).

In Figure 5-7, the expected total delay is plotted against Δ for a fixed frame length. Our goal is to find Δ^*, the optimal additional session reservation time that results in the minimum total delay. We set the derivative of the total expected delay with respect to Δ to zero to find the optimum Δ, where

$$\frac{\partial E_n[T]}{\partial \Delta} = \frac{\left(n\Delta\frac{L}{R} + \sum_{i=1}^{n-1}\left[p_{i/n}\frac{\partial E_i[T]}{\partial \Delta} + \frac{\partial p_{i/n}}{\partial \Delta}E_i[T]\right]\right)(1-p_{n/n})}{(1-p_{n/n})^2} \tag{5.23}$$

$$+\frac{\frac{\partial p_{n/n}}{\partial \Delta}\left(\lfloor n(1+\Delta)\rfloor\frac{L}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{i/n}E_i[T]\right)}{(1-p_{n/n})^2} = 0.$$

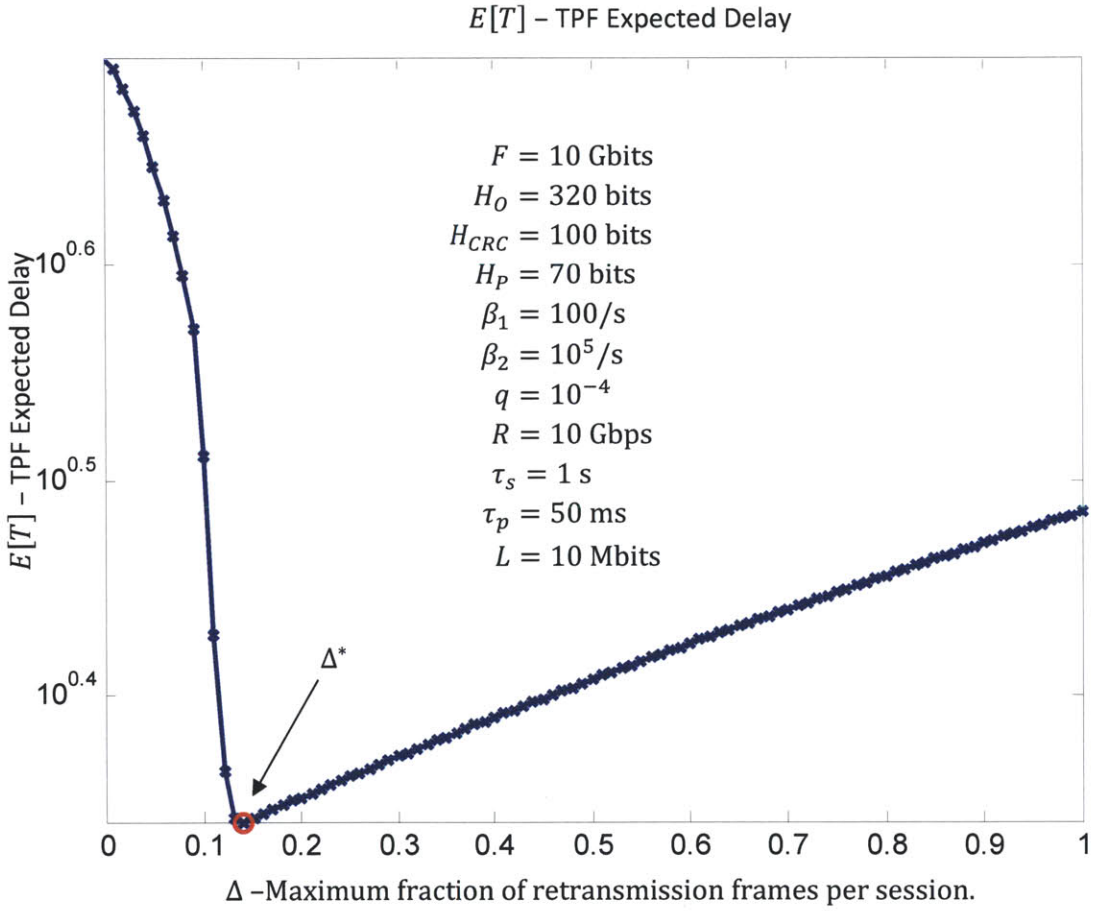


Figure 5-7: TPF total expected delay vs. Δ.

Equation (5.23), however, does not lead to a closed-form solution for Δ^*. Instead, we can approximate the total delay expression in the region of interest around Δ^* and $E[T^*]$. We assume that due to large setup delays shown in Chapter 3, users will try to avoid multiple requests for new flow sessions. Thus we assume that the probability of more than one retransmission session is small. Let θ

be the probability of a failed initial transmission defined in (5.5). Let $E[X_1]$ be the expected number of outstanding erroneous frames after the initial session. We approximate the total expected delay as

$$E[T] \cong n(1+\Delta)\frac{L}{R} + \tau_s + \tau_p + \theta\left(E[X_1]\frac{L}{R} + \tau_s + \tau_p\right). \tag{5.24}$$

We note that if Δ is such that the probability that a new session is needed for retransmissions is large, the approximations made in (5.24) do not account for multiple retransmission session delays and results in a delay much smaller than the actual expected delay. We find the expression for $E[X_1]$ in Appendix 0. Using the normal approximation for θ in (5.8) we approximate the total expected delay expression as

$$\begin{aligned} E[T] \cong\ & n(1+\Delta)\frac{L}{R} + \tau_s + \tau_p \\ & + \Phi^{c}\left(\frac{\mu-(n-1)}{\sigma}\right)\left(E[X_1]\frac{L}{R} + \tau_s + \tau_p\right). \end{aligned} \tag{5.25}$$

For large setup delays, we notice that

$$E[X_1]\frac{L}{R} \ll \tau_s \tag{5.26}$$

and can be ignored in the delay expression.

We further approximate θ with the bound given in (5.9). In addition, we approximate the variance given in (5.7) as

$$\begin{aligned} \sigma^2 &= np(1-p)(1+\Delta) \\ &\cong np(1-p). \end{aligned} \tag{5.27}$$

Using the approximations made in (5.24) - (5.27), the resulting total expected delay expression is

$$E[T] \cong n(1+\Delta)\frac{L}{R} + \tau_s + \tau_p + \frac{1}{2}e^{-\frac{(n(1+\Delta)(1-p)-(n-1))^2}{2np(1-p)}}(\tau_s + \tau_p). \tag{5.28}$$

Figure 5-8 shows that the delay approximations made in (5.28) hold in the region near Δ^*.

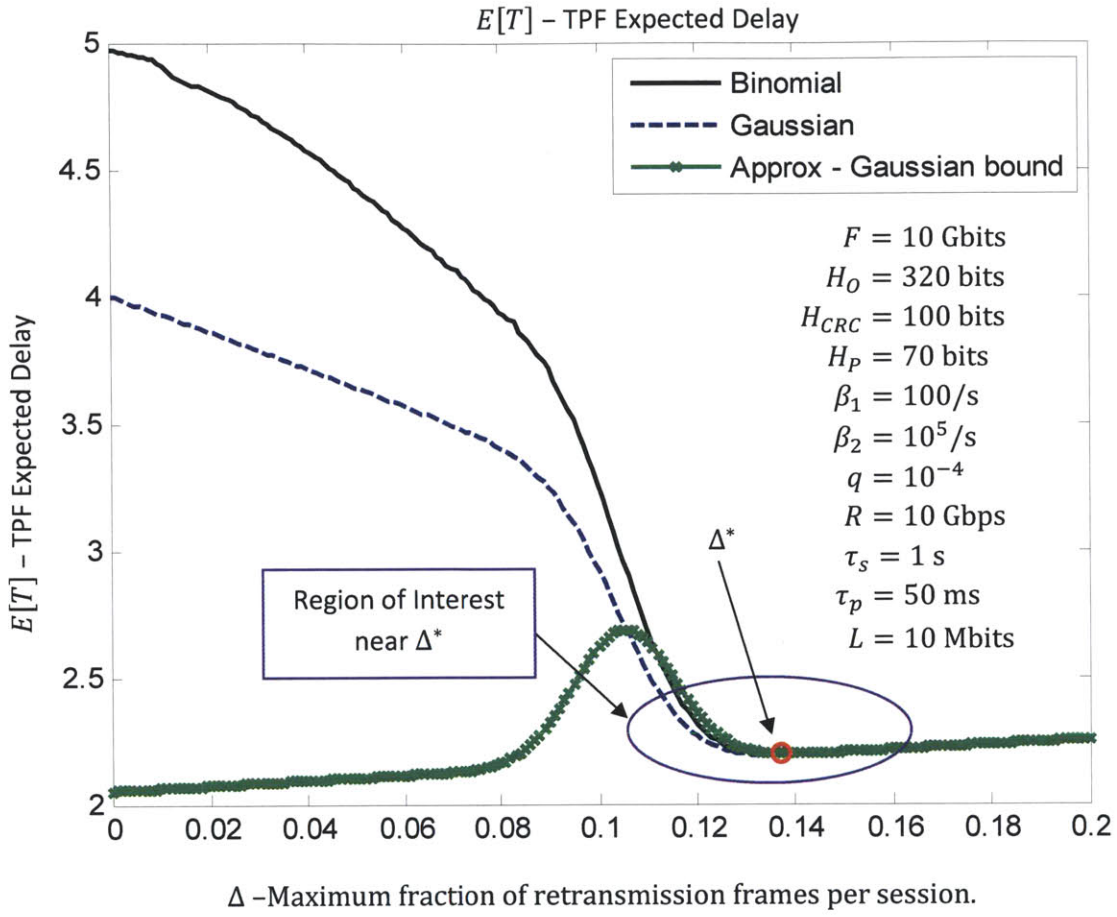


Figure 5-8: TPF total expected delay approximation. "Binomial" refers to the total expected delay expression in (5.22). "Gaussian" refers to the total expected delay approximation in (5.25). "Approx - Gaussian bound" refers to the total expected delay approximation in (5.28).

We take the derivative of the total expected delay with respect to Δ where

$$\frac{\partial E[T]}{\partial \Delta} = n\frac{L}{R} + \frac{1}{2}\frac{\partial \theta}{\partial \Delta}(\tau_s + \tau_p)$$

$$= \frac{nL}{R} - \frac{\tau_s + \tau_p}{2} e^{-\frac{(n(1+\Delta)(1-p)-(n-1))^2}{2np(1-p)}} \frac{n(1+\Delta)(1-p)-(n-1)}{p} = 0 \qquad (5.29)$$

$$\frac{2pnL}{(\tau_s + \tau_p)R} = e^{-\frac{(\Delta n(1-p)-np+1)^2}{2np(1-p)}} (\Delta n(1-p) - np + 1).$$

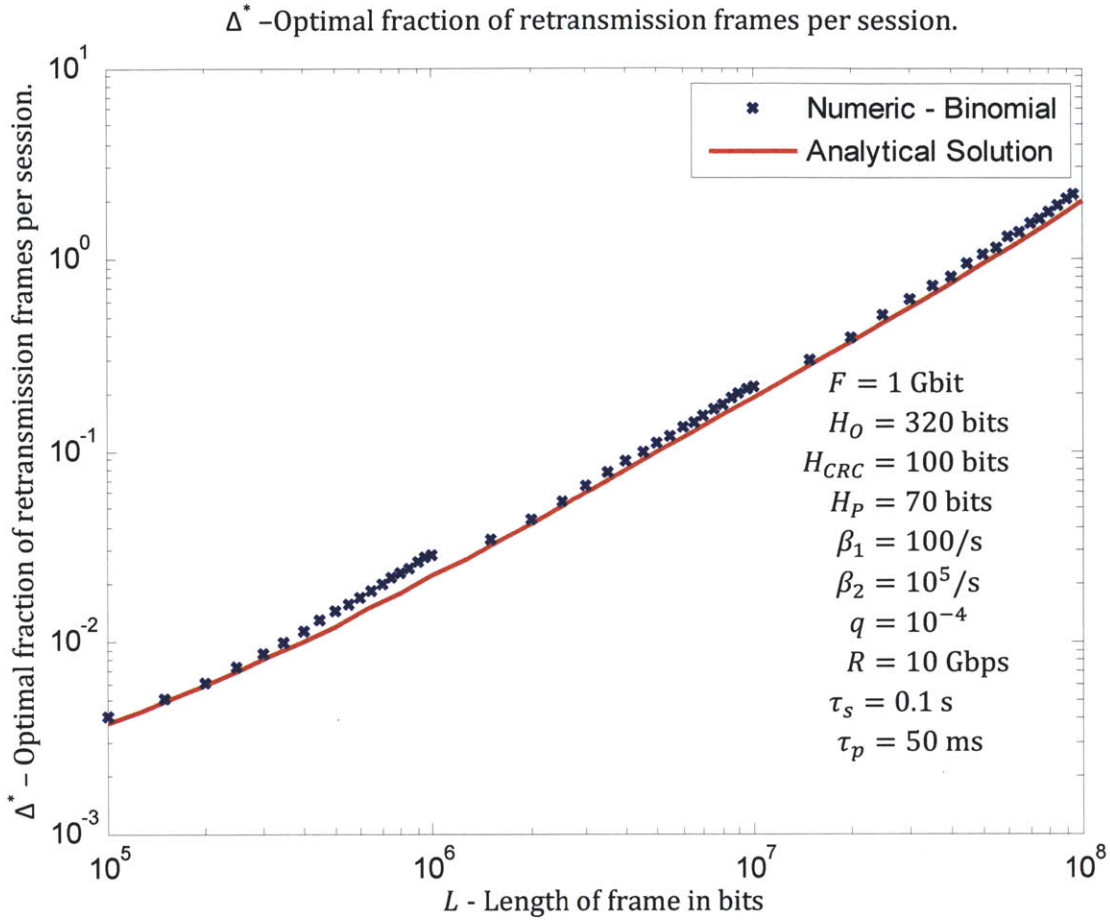


Figure 5-9: TPF optimal fraction of retransmission frames per session vs. frame length. "Numeric-Binomial" refers to numeric solutions to expression in (5.22). "Analytical Solution" refers to the expression for Δ^* in (5.30).

We solve (5.29) for Δ^* where

$$\Delta^* = \frac{p}{1-p} - \frac{1}{n(1-p)} + \sqrt{-\frac{p}{n(1-p)} W_{-1}\left(-\frac{np\left(\frac{2L}{(\tau_s+\tau_p)R}\right)^2}{(1-p)}\right)} \qquad (5.30)$$

and $W_{-1}(x)$ is the other real branch of the Lambert W-function[1]. Δ^* is plotted against frame length in Figure 5-9.

5.1.3.2 Optimal Frame Length

In the previous section, we found the optimal additional session reservation duration. In this section, we set $\Delta = \Delta^*$ in the expression for the total expected delay and solve for the optimal frame length. We assume that for frame lengths near the optimal frame length, the expected time between session arrivals in a fiber is much larger than the time to send a frame where $1/\beta_1 \gg L/R$. Thus we assume that

$$\frac{\beta_1 L}{R} \ll 1, \qquad for\ L \cong L^*. \tag{5.31}$$

We assume that near the optimal frame length, $D \gg H$ and $D \gg H_P$. Thus we approximate $L \cong D/C_q$. We can approximate p by the first two terms of the Maclaurin series expansion where

$$\begin{aligned} p &= 1 - \frac{\beta_2}{\beta_1 + \beta_2} e^{-\frac{\beta_1 L}{R}} \\ &\cong \frac{\beta_1}{\beta_1 + \beta_2}\left(1 + \frac{\beta_2 L}{R}\right) \\ &\cong \frac{\beta_1}{\beta_1 + \beta_2}\left(1 + \frac{\beta_2 D}{R C_q}\right). \end{aligned} \tag{5.32}$$

We also simplify $1 - p$ in the same way where

$$\begin{aligned} 1 - p &\cong \frac{\beta_2}{\beta_1 + \beta_2}\left(1 - \frac{\beta_1 L}{R}\right) \\ &\cong \frac{\beta_2}{\beta_1 + \beta_2}. \end{aligned} \tag{5.33}$$

Δ^* can be approximated using the asymptotic expression for $W_{-1}(x)$ where

---

[1] The Lambert W-function is the inverse function of $f(x) = xe^x$.

$$\Delta^* \cong \frac{p}{1-p} - \frac{D}{F(1-p)} + \sqrt{\frac{Dp}{F(1-p)} \ln\left(\frac{D(\tau_s+\tau_p)^2 R^2(1-p)}{4FpL^2}\right)}. \tag{5.34}$$

We substitute (5.32), (5.33), and $L \cong D/C_q$ into (5.34) where

$$\Delta^* \cong \frac{\beta_1}{\beta_2}\left(1+\frac{\beta_2 D}{RC_q}\right) - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{D\beta_1}{F\beta_2}\left(1+\frac{\beta_2 D}{RC_q}\right)\ln\left(\frac{(\tau_s+\tau_p)^2 \beta_2 R^2 C_q^2}{4F\beta_1 D\left(1+\frac{\beta_2 D}{RC_q}\right)}\right)}. \tag{5.35}$$

We assume $\theta \cong 0$ in the region of interest near $E[T^*]$. The resulting total expected delay expression becomes

$$E[T] \cong \frac{F}{D}\frac{D+H+C_qH_P}{C_qR}(1+\Delta) + \tau_s + \tau_p. \tag{5.36}$$

Substituting (5.35) into (5.36) results in

$$E[T] \cong \frac{F}{D}\frac{D+H+C_qH_P}{C_qR}\left(1+\frac{\beta_1}{\beta_2}\left(1+\frac{\beta_2 D}{RC_q}\right) - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{D\beta_1}{F\beta_2}\left(1+\frac{\beta_2 D}{RC_q}\right)\ln\left(\frac{(\tau_s+\tau_p)^2 \beta_2 R^2 C_q^2}{4F\beta_1 D\left(1+\frac{\beta_2 D}{RC_q}\right)}\right)}\right) + \tau_s + \tau_p. \tag{5.37}$$

We differentiate the total expected delay with respect to the message length where

$$\frac{\partial E[T]}{\partial D} = -\frac{F}{D^2}\frac{H+C_qH_P}{C_qR}(1+\Delta) + \frac{F}{D}\frac{D+H+C_qH_P}{C_qR}\frac{\partial \Delta}{\partial D} \tag{5.38}$$

$$= -\frac{F}{D^2}\frac{H + C_q H_P}{C_q R}\left(1 + \frac{\beta_1}{\beta_2}\left(1 + \frac{\beta_2 D}{RC_q}\right) - \frac{\beta_1 + \beta_2}{F\beta_2}\right.$$

$$\left. + \sqrt{\frac{D\beta_1}{F\beta_2}\left(1 + \frac{\beta_2 D}{RC_q}\right)\ln\left(\frac{\left(\tau_s + \tau_p\right)^2 \beta_2 R^2 C_q^2}{4F\beta_1 D\left(1 + \frac{\beta_2 D}{RC_q}\right)}\right)}\right)$$

$$+ \frac{F}{D}\frac{D + H + C_q H_P}{C_q R}\left(\frac{\beta_1}{RC_q} + \frac{\frac{\beta_1}{F\beta_2}\left(1 + 2\frac{\beta_2 D}{RC_q}\right)\left(\log\left(\frac{\left(\tau_s + \tau_p\right)^2 \beta_2 R^2 C_q^2}{4F\beta_1 D\left(1 + \frac{\beta_2 D}{RC_q}\right)}\right) - 1\right)}{2\sqrt{\frac{\beta_1 D}{F\beta_2}\left(1 + \frac{\beta_2 D}{RC_q}\right)\log\left(\frac{\left(\tau_s + \tau_p\right)^2 \beta_2 R^2 C_q^2}{4F\beta_1 D\left(1 + \frac{\beta_2 D}{RC_q}\right)}\right)}}\right)$$

$$= 0.$$

Equation (5.38) is a transcendental equation and does not lead to a closed-form solution for L^*. Table 5-3 shows the numerical solutions for the optimal frame size.

Optimal Frame Length L^*	
Actual - Binomial	130 Kbits
Approximation –Solve	150 Kbits
Parameters	**Value**
β_1	100
β_2	10^5
R	10 Gbps
H_O	320 bits
H_{CRC}	100 bits
H_P	70 bits
F	1 Gbit
τ_s	100 ms
τ_p	50 ms
q	10^{-4}

Table 5-3 TPF optimal frame length. "Actual –Binomial" is the numerical solution for L^* in (5.22). "Approximation – Solve" is the numerical solution for L^* in (5.38).

Let D^* be the solution to (5.38). Let L^* be the optimal frame length where

$$L^* \cong \frac{D^* + H}{C_q} + H_P. \tag{5.39}$$

The expected minimum delay is

$$E[T^*] \cong \frac{F}{D^*} \frac{D^* + H + C_q H_P}{C_q R} (1 + \Delta^*) + \tau_s + \tau_p. \tag{5.40}$$

which is the delay for one transmission. What this result is saying is that the additional setup and queueing delay of retransmission is so large that the optimum strategy is to finish transmission in one session with probability close to one.

5.1.3.3 Practical Frame Length

A large frame length may be desirable to reduce the total number of frames sent and minimize the overhead associated with segmentation and reassembly of a file. We notice that the delay expression as a function of frame length is shallow near the optimal frame length. Thus, users can choose to segment a file into larger frames and experience a small delay performance penalty in exchange for less segmentation and reassembly overhead. In this section, we find the practical frame length, the frame length that results in an expected total delay ε away from the optimal expected total delay.

We observe that the practical frame length will be large (> 1 Mbit). For frame lengths near the practical frame length, the expected duration of an outage period is assumed to be much smaller than the time to send a frame. Thus we assume that

$$\frac{\beta_2 L}{R} \gg 1, \qquad for\ L \cong L_\varepsilon. \tag{5.41}$$

We note that the approximation made in (5.31) does not hold for large frame sizes. We further approximate the expression for p in (5.32) where

$$p \cong \frac{\beta_1}{\beta_1 + \beta_2} \frac{\beta_2 D}{R C_q}. \tag{5.42}$$

From (5.42) we can further approximate the expression for Δ^* in (5.34) where

$$\Delta^* \cong D\left(\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}\sqrt{\ln\left(\frac{(\tau_s+\tau_p)^2(C_qR)^3}{4F\beta_1 D^2}\right)}\right). \tag{5.43}$$

Let c be the constant term within the logarithm in (5.43) where

$$c = \frac{(\tau_s+\tau_p)^2(C_qR)^3}{4F\beta_1}. \tag{5.44}$$

We assume $\theta \cong 0$ in the region of interest. The resulting total expected delay expression near the practical frame length can be approximated as in (5.36). In addition, we notice that $D \gg H$ and $D \gg H_P$ near the practical frame length. Thus we can approximate $L \cong D/C_q$. The total expected delay expression near the practical frame length can be simplified to

$$\begin{aligned} E[T] &\cong \frac{F}{RC_q}(1+\Delta) + \tau_s + \tau_p \\ &= \frac{F}{RC_q}\left(1 + D\left(\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}\sqrt{\ln\frac{c}{D^2}}\right)\right) + \tau_s + \tau_p. \end{aligned} \tag{5.45}$$

The practical frame length has delay ε away from the optimal delay where

$$E[T_\varepsilon] = (1+\varepsilon)E[T^*]. \tag{5.46}$$

We solve (5.46) for the practical frame length where

$$\begin{aligned} E[T_\varepsilon] &= \frac{F}{RC_q}\left(1 + D\left(\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}\sqrt{\ln\frac{c}{D^2}}\right)\right) + \tau_s + \tau_p \\ D\left(\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}\sqrt{\ln\frac{c}{D^2}}\right) &= \frac{RC_q}{F}\left(E[T_\varepsilon] - \tau_s - \tau_p\right) - 1. \end{aligned} \tag{5.47}$$

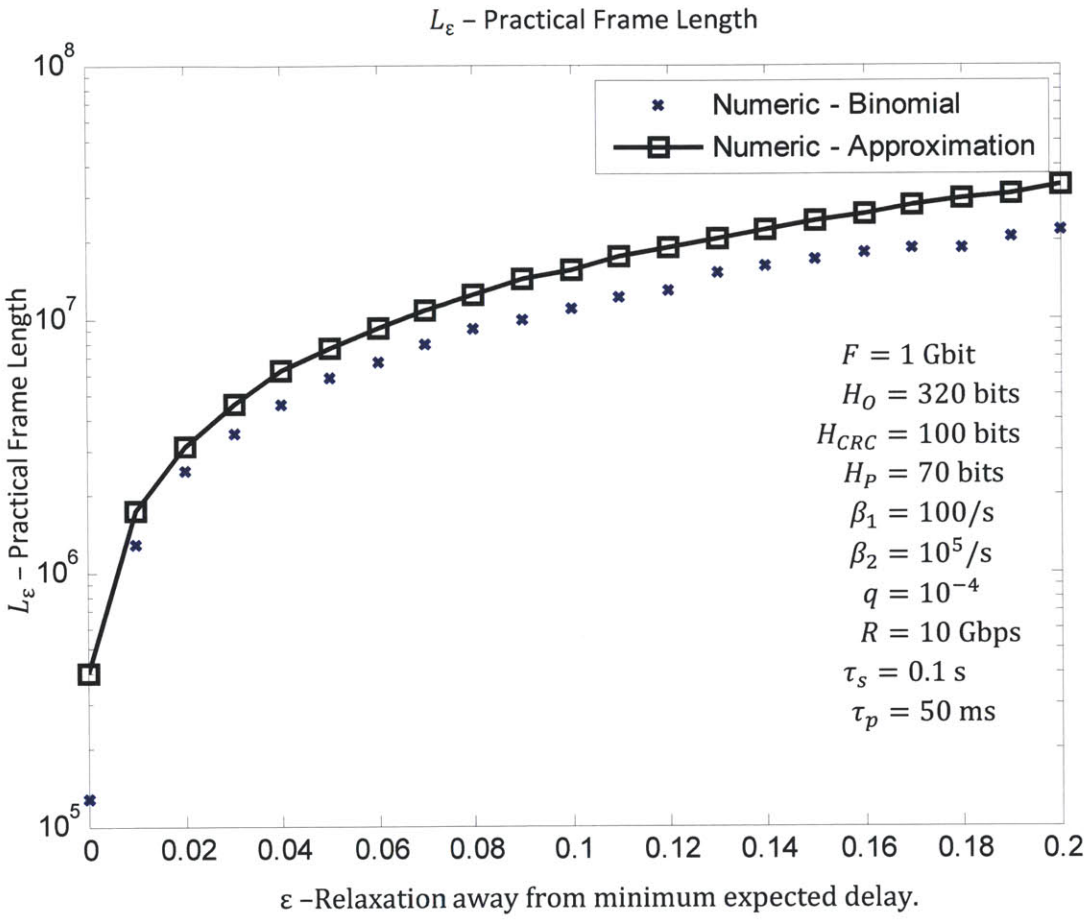


Figure 5-10: TPF practical frame length ε away from the optimal expected total delay. "Numeric – Binomial" corresponds to the numerical solution corresponding to (5.22). "Numeric – Approximation" corresponds to the numerical solution corresponding to (5.47).

Numeric solutions for the practical frame length are shown in Figure 5-10. We provide upper and lower bounds for the maximum frame size in Table 5-4.

Lower Bound for L_ε:

The right hand side of (5.47) can be bounded by

$$D\left(\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}+\sqrt{\frac{\beta_1}{FC_qR}}\ln\frac{c}{D^2}\right)>\frac{RC_q}{F}\left(E[T_\varepsilon]-\tau_s-\tau_p\right)-1. \tag{5.48}$$

This leads to a lower bound for the practical frame length where

$$L_\varepsilon \geq -\frac{\frac{RC_q}{F}\left(E[T_\varepsilon]-\tau_s-\tau_p\right)-1}{2\sqrt{\frac{\beta_1}{FC_qR}}\,W_{-1}\left(-\frac{E[T_\varepsilon]-\tau_s-\tau_p-\frac{F}{C_qR}}{\tau_s+\tau_p}\sqrt{e^{-\sqrt{\frac{FC_qR}{\beta_1}}\left(\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}\right)}}\right)}. \tag{5.49}$$

Upper Bound for L_ε:

The right hand side of (5.47) can be bounded by

$$D\left(\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}+\sqrt{\frac{\beta_1}{FC_qR}}\right)<\frac{RC_q}{F}\left(E[T_\varepsilon]-\tau_s-\tau_p\right)-1. \tag{5.50}$$

This leads to an upper bound for the practical frame length where

$$L_\varepsilon \leq \frac{\frac{RC_q}{F}\left((1+\varepsilon)E[T]^*-\tau_s-\tau_p\right)-1}{\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}+\sqrt{\frac{\beta_1}{FC_qR}}}. \tag{5.51}$$

Table 5-4: TPF upper and lower bounds for the practical frame length given in (5.49) and (5.51).

We plot the upper and lower bounds for the practical frame length in Figure 5-11.

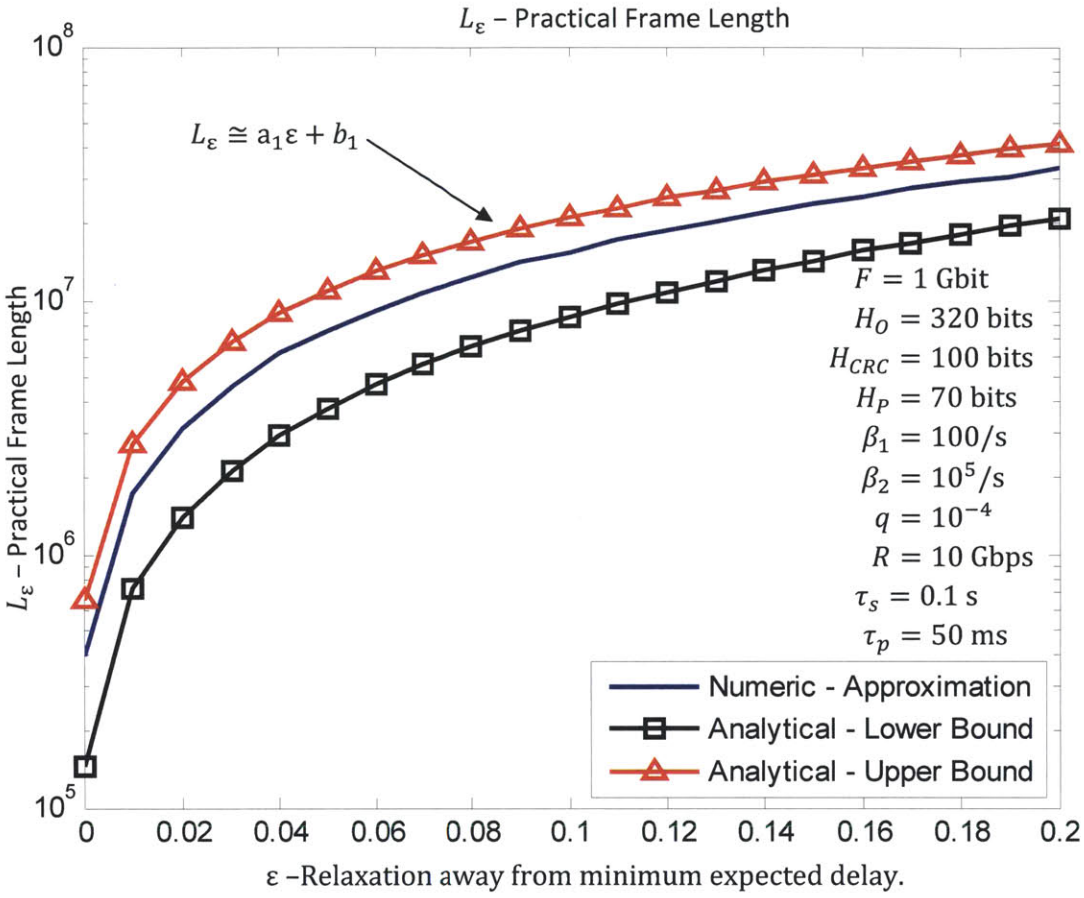


Figure 5-11: TPF upper and lower bounds for the practical frame length. The lower bound corresponds to (5.49). The upper bound corresponds to (5.51). The upper bound is linear in ε[2].

5.2 Transport Protocol with Framing and Interleaving (TPFI): FEC corrects up to one burst error

In this section, we find the expected throughput and delay performance of the Transport Protocol with Framing and Interleaving (TPFI). In TPFI, a large file is segmented into frames at the source and reassembled at the destination. We assume interleaving is performed at the channel input and FEC can

[2] The upper bound is linear in ε. $L_\varepsilon \cong a_1\varepsilon + b_1$ where $a_1 = \frac{\frac{RC_qE[T]^*}{F}}{\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}}$ and $b_1 = \frac{\frac{RC_q}{F}(E[T]^* - \tau_s - \tau_p) - 1}{\frac{\beta_1}{RC_q} - \frac{\beta_1+\beta_2}{F\beta_2} + \sqrt{\frac{\beta_1}{FC_qR}}}$.

be used to correct for both random errors and burst errors. ARQ allows frames that are received with error to be retransmitted. Only erroneous frames are retransmitted.

In this section, we assume that frames are of finite length, thus perfect interleaving cannot be achieved. We make the simplifying assumption that the TPFI interleaver depth is such that FEC can only correct up to one burst error. If a frame experiences two or more burst errors during transmission, the frame is uncorrectable and must be discarded. In Section 5.3, we consider the case where interleaving and FEC is used to correct for a general number of bursts.

5.2.1 Error Probabilities

In this section, we define two error probabilities: p_I, the probability of an erroneous frame and θ_I, the probability of a failed initial transmission. Let C_ξ be capacity of the interleaved channel with crossover probability ξ defined in Chapter 3. Let L_I be the total number of bits per frame (message plus overhead) where

$$L_I = \frac{D+H}{C_\xi} + H_p. \tag{5.52}$$

The probability of an erroneous frame is the probability of more than one outage in a frame time where

$$\begin{aligned} p_I &= 1 - P(0 \text{ bursts}) - P(1 \text{ burst} | S_0 = \text{non–outage}) P(S_0 = \text{non–outage}) \\ &= 1 - e^{-\frac{\beta_1 L_I}{R}} - \frac{\beta_2}{\beta_1 + \beta_2} \frac{\beta_1 L_I}{R} e^{-\frac{\beta_1 L_I}{R}}. \end{aligned} \tag{5.53}$$

We plot both p_I and p against D, the number of message bits per frame in Figure 5-12. We notice that the probability of an erroneous frame with interleaving is smaller than the probability of an erroneous frame without interleaving. As in TPF, in TPFI, a file is segmented into n sequential frames, where $n = \lceil F/D \rceil$. The probability that a file is received in error is equal to the probability that one or more frames is received in error.

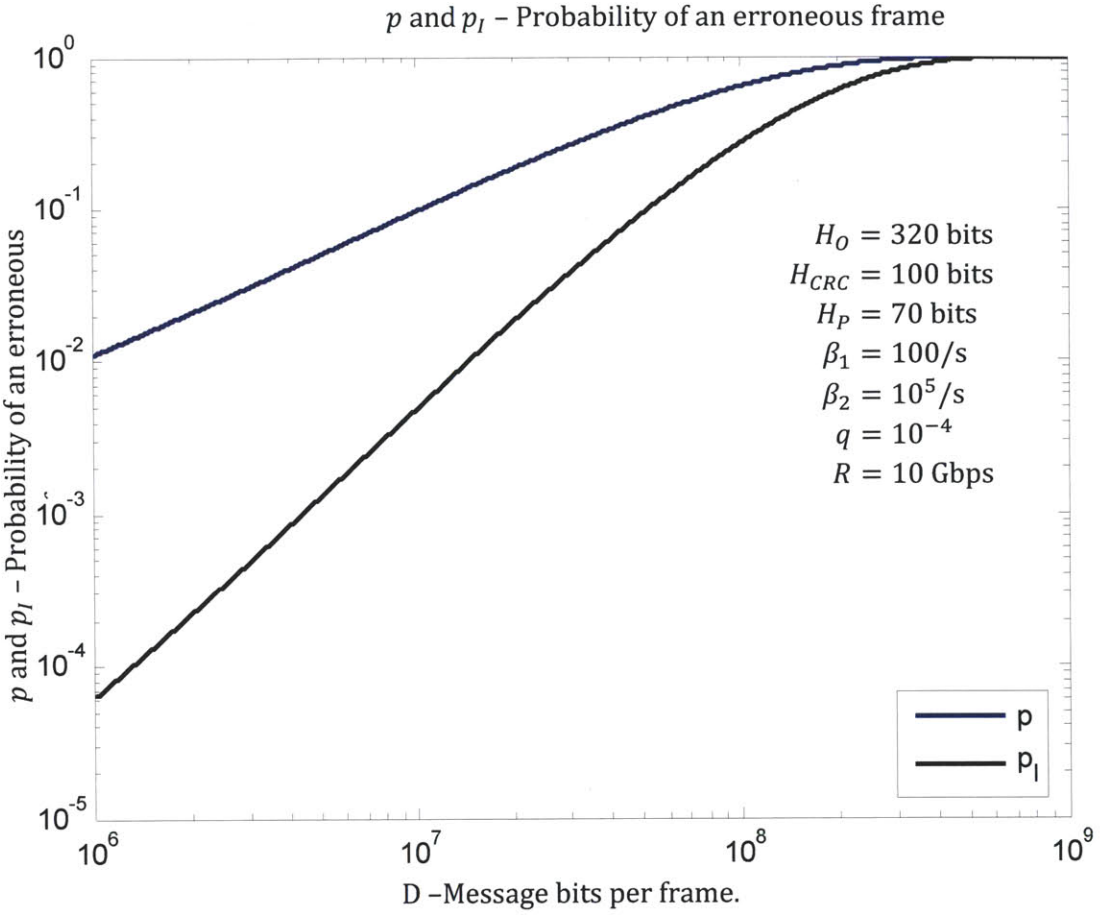


Figure 5-12: TPFI probability of an erroneous frame (p_I) and TPF probability of an erroneous frame (p) vs. D.

If additional time per session is not reserved for frame retransmissions, the probability of a failed initial transmission is

$$\theta_I = \sum_{k=0}^{n-1} \binom{n}{k} (1-p_I)^k {p_I}^{n-k} \qquad (5.54)$$
$$= 1-(1-p_I)^n.$$

If additional time per session is reserved for frame retransmissions, the probability of a failed initial transmission is

$$\theta_I = \sum_{k=0}^{n-1} \binom{\lfloor n(1+\Delta) \rfloor}{k} (1-p_I)^k {p_I}^{\lfloor n(1+\Delta) \rfloor - k}. \qquad (5.55)$$

The distribution in (5.55) has mean

$$\mu_I = \lfloor n(1+\Delta) \rfloor (1-p_I) \tag{5.56}$$

and variance

$$\sigma_I{}^2 = \lfloor n(1+\Delta) \rfloor (1-p_I)p_I. \tag{5.57}$$

θ_I can be bounded as in (5.9) by

$$\begin{aligned} \theta_I &> 1 - \frac{1}{2} e^{-\frac{(n-1-\mu_I)^2}{2\sigma_I{}^2}}, \quad for\ n-1 > \mu_I, \\ \theta_I &< \frac{1}{2} e^{-\frac{(\mu_I - n+1)^2}{2\sigma_I{}^2}}, \quad for\ n-1 \leq \mu_I. \end{aligned} \tag{5.58}$$

5.2.2 Performance Optimization - Throughput

Let η_I be the expected TPFI throughput. The derivation of the expected throughput in the same as in Section 5.1.2 where

$$\begin{aligned} \eta_I &= \frac{D}{L_I}(1-p_I) \\ &= \frac{D}{L_I} e^{-\frac{\beta_1 L_I}{R}} \left(1 + \frac{\beta_2}{\beta_1+\beta_2} \frac{\beta_1 L_I}{R}\right). \end{aligned} \tag{5.59}$$

In the next section, we find the optimal frame length that maximizes the expected throughput.

5.2.2.1 Optimal Frame Length

We substitute (5.52) into (5.59) to find the expected throughput as a function of the message size where

$$\begin{aligned} \eta_I &= \frac{D}{L_I}(1-p_I) \\ &= \frac{DC_\xi}{D+H+C_\xi H_P} e^{-\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}} \left(1 + \frac{\beta_2}{\beta_1+\beta_2} \frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}\right). \end{aligned} \tag{5.60}$$

We take the derivative of η_I with respect to D to find the optimal message length where

$$\frac{\partial \eta_I}{\partial D} = e^{-\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}}\left(\frac{\beta_2\beta_1}{\beta_1+\beta_2}\frac{C_\xi R-\beta_1 D}{C_\xi R^2} + \frac{C_\xi(H+C_\xi H_P)R-\beta_1 D(D+H+C_\xi H_P)}{R(D+H+C_\xi H_P)^2}\right)=0. \tag{5.61}$$

Equation (5.61) simplifies to

$$-\frac{\beta_2{\beta_1}^2}{\beta_1+\beta_2}D^3+\beta_1\left(\frac{\beta_2}{\beta_1+\beta_2}\left(C_\xi R-2\beta_1(H+C_\xi H_P)\right)-C_\xi R\right)D^2 + (H+C_\xi H_P)\beta_1\left(\frac{\beta_2}{\beta_1+\beta_2}\left(2C_\xi R-\beta_1(H+C_\xi H_P)\right)-C_\xi R\right)D + (H+C_\xi H_P)C_\xi R\left(C_\xi R+\frac{\beta_2\beta_1(H+C_\xi H_P)}{\beta_1+\beta_2}\right)=0. \tag{5.62}$$

We solve (5.62) for the optimal message length. Let $D_I{}^*$ the length of the message at the optimal frame length where

$$D_I{}^* = -\frac{b}{3a}-\frac{1}{3a}\sqrt[3]{\frac{1}{2}\left(2b^3-9abc+27a^2d+\sqrt{(2b^3-9abc+27a^2d)^2-4(b^2-3ac)^3}\right)} -\frac{1}{3a}\sqrt[3]{\frac{1}{2}\left(2b^3-9abc+27a^2d-\sqrt{(2b^3-9abc+27a^2d)^2-4(b^2-3ac)^3}\right)} \tag{5.63}$$

and

$$\begin{aligned} a &= -\frac{\beta_2{\beta_1}^2}{\beta_1+\beta_2}\\ b &= \beta_1\left(\frac{\beta_2}{\beta_1+\beta_2}\left(C_\xi R-2\beta_1(H+C_\xi H_P)\right)-C_\xi R\right)\\ c &= (H+C_\xi H_P)\beta_1\left(\frac{\beta_2}{\beta_1+\beta_2}\left(2C_\xi R-\beta_1(H+C_\xi H_P)\right)-C_\xi R\right)\\ d &= (H+C_\xi H_P)C_\xi R\left(C_\xi R+\frac{\beta_2\beta_1(H+C_\xi H_P)}{\beta_1+\beta_2}\right). \end{aligned} \tag{5.64}$$

Let $L_I{}^*$ be the optimal frame length where

$$L_I{}^* = \frac{D^* + H}{C_\xi} + H_P. \tag{5.65}$$

Let $\eta_I{}^*$ be the maximum expected throughput where

$$\eta_I{}^* = \frac{D}{L_I{}^*} e^{-\frac{\beta_1 L_I{}^*}{R}} \left(1 + \frac{\beta_2}{\beta_1 + \beta_2} \frac{\beta_1 L_I{}^*}{R}\right). \tag{5.66}$$

We plot the TPFI throughput as a function of frame length in Figure 5-13.

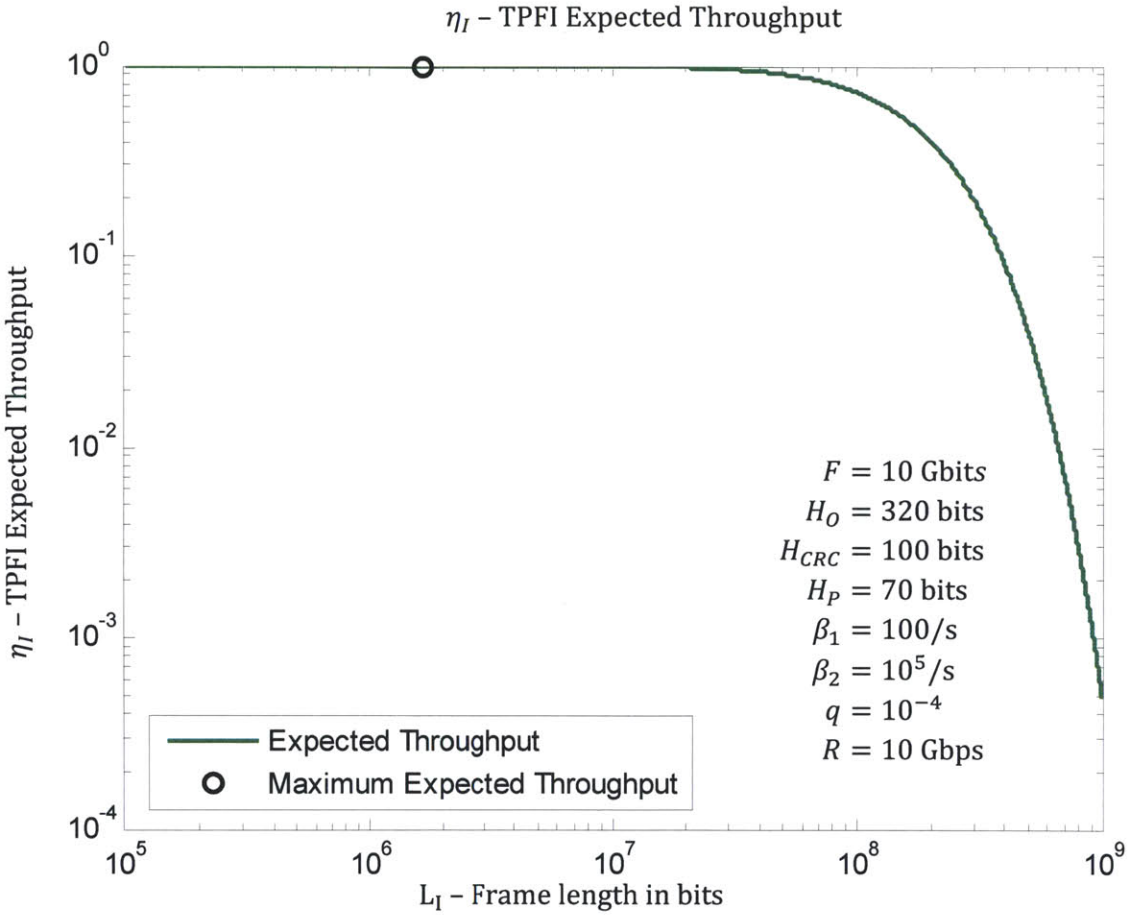


Figure 5-13: TPFI throughput as a function of frame length. The black circle indicates the location of the maximum expected throughput.

5.2.2.2 Practical Frame Length

As in TPF, users who wish to avoid large segmentation and reassembly overhead can choose to partition a file into frames of length larger than the optimal frame length. We can approximate the expected throughput at the practical frame length as

$$\eta_I \cong C_\xi e^{-\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}}\left(1+\frac{\beta_2}{\beta_1+\beta_2}\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}\right). \tag{5.67}$$

Let η_ε^I be the expected throughput ε away from the optimal expected throughput. At the practical frame length,

$$\eta_\varepsilon^I = (1-\varepsilon)\eta_I{}^*. \tag{5.68}$$

We substitute (5.67) into (5.68) and solve for the practical frame length where

$$e^{-\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}}\left(1+\frac{\beta_2}{\beta_1+\beta_2}\frac{\beta_1(D+H+C_\xi H_P)}{C_\xi R}\right) \cong \frac{(1-\varepsilon)\eta_I{}^*}{C_\xi}. \tag{5.69}$$

Let D_ε^I be the length of the message at the practical frame length and let L_ε^I be the practical frame length. The maximum message and frame lengths are

$$\begin{aligned} D_\varepsilon^I &\cong -\frac{RC_\xi}{\beta_1}\mathrm{W}_{-1}\left[-\frac{e^{-\frac{\beta_1+\beta_2}{\beta_2}}(1-\varepsilon)\eta_I{}^*}{C_\xi\frac{\beta_2}{\beta_1+\beta_2}}\right]-\frac{RC_\xi}{\beta_1}\frac{\beta_1+\beta_2}{\beta_2}-H-C_\xi H_P \\ L_\varepsilon^I &\cong -\frac{R}{\beta_1}\mathrm{W}_{-1}\left[-\frac{e^{-\frac{\beta_1+\beta_2}{\beta_2}}(1-\varepsilon)\eta_I{}^*}{C_\xi\frac{\beta_2}{\beta_1+\beta_2}}\right]-\frac{R}{\beta_1}\frac{\beta_1+\beta_2}{\beta_2}. \end{aligned} \tag{5.70}$$

We plot the practical frame length as a function of ε in Figure 5-14. Observe that the frame size can be quite large ($> 10^7$) with little loss of efficiency.

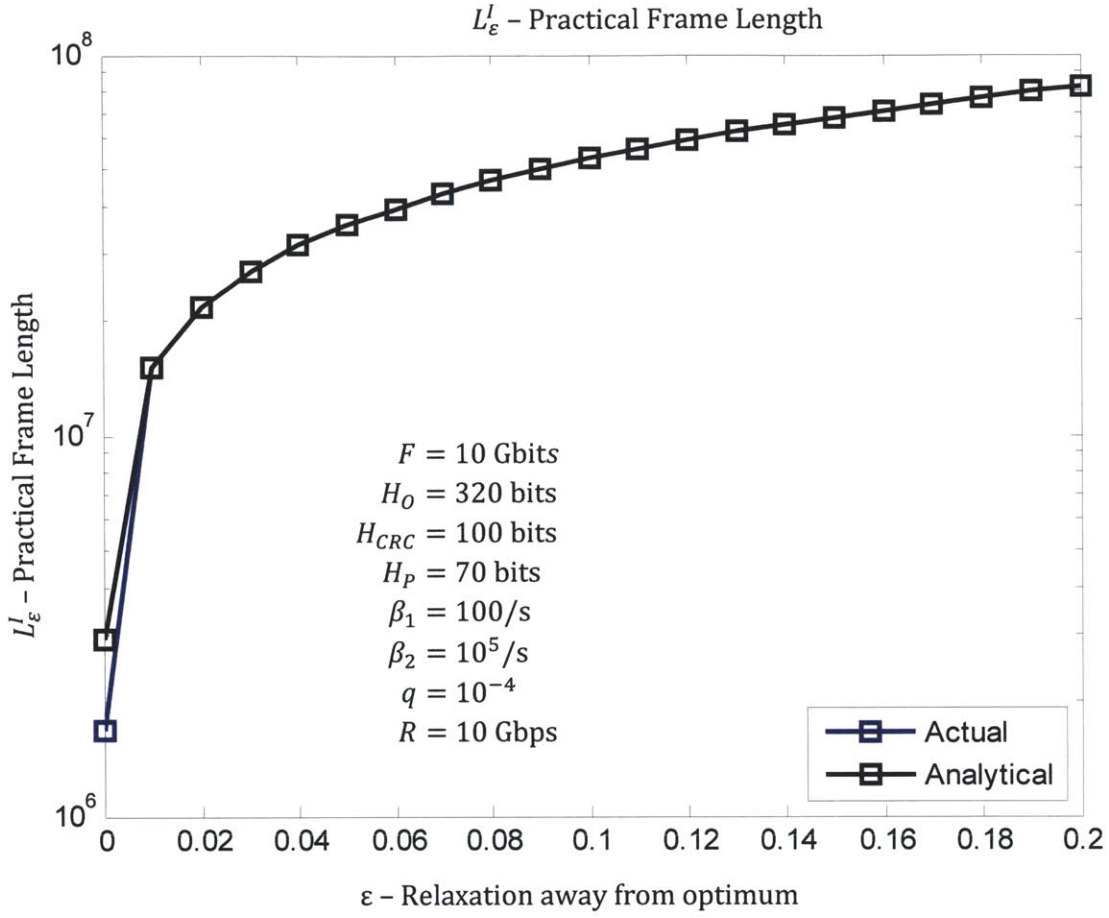


Figure 5-14: TPFI practical frame length vs. ε. "Actual" corresponds to numerical solutions for the practical frame length in (5.69). "Analytical" corresponds to the expression for the practical frame length in (5.70)".

5.2.3 Performance Optimization – Delay

In this section, we find the minimum total expected delay. Let T_I be the total delay and $E[T_I]$ be the expected total delay to send a file. Let $E[T_I{}^*]$ be the optimal expected delay. The derivation for $E[T_I]$ follows from Section 5.1.3. The expected total delay for no setup and propagation delay in given (5.71). The expected total delay for nonzero setup and propagation delay in given in (5.72).

$\tau_s = 0$ and $\tau_p = 0$	
$E[T_I] = \frac{\frac{F}{D}\frac{L_I}{R}}{1-p_I} = \frac{F}{R}\frac{1}{\eta_I}$	(5.71)

Table 5-5: TPFI total expected delay for no setup and propagation delay.

$\tau_s \neq 0$ and $\tau_p \neq 0$	
$p_{I_{i\backslash n}} = \binom{\lfloor n(1+\Delta)\rfloor}{n-i}(1-p_I)^{n-i}p_I^{\lfloor n(1+\Delta)\rfloor - n+i}$ $p_{I_{n\backslash n}} = \binom{\lfloor n(1+\Delta)\rfloor}{0}(1-p_I)^0 p_I^{\lfloor n(1+\Delta)\rfloor} = p_I^{\lfloor n(1+\Delta)\rfloor}$ $E[T_I] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L_I}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{I_{i\backslash n}} E_i[T_I]}{1 - p_{I_{n\backslash n}}}$	(5.72)

Table 5-6: TPFI total expected delay for nonzero setup and propagation delay.

5.2.3.1 Optimal Additional Session Reservation

As in TPF, in TPFI, users have the option to request additional time per session for retransmissions to avoid the need for additional session requests and additional setup delays. Let Δ_I^* be the optimal additional session reservation. The derivation for Δ_I^* follows from the derivation for Δ^* where

$$\Delta_I^* = \frac{p_I}{1-p_I} - \frac{1}{n(1-p_I)} + \sqrt{-\frac{p_I}{n(1-p_I)} W_{-1}\left(-\frac{np_I\left(\frac{2L_I}{(\tau_s+\tau_p)R}\right)^2}{(1-p_I)}\right)} \tag{5.73}$$

$$\cong \frac{p}{1-p} - \frac{D}{F(1-p)} + \sqrt{\frac{Dp}{F(1-p)}\ln\left(\frac{D(\tau_s+\tau_p)^2 R^2(1-p)}{4FpL^2}\right)}.$$

We plot Δ_I^* against frame length in Figure 5-15.

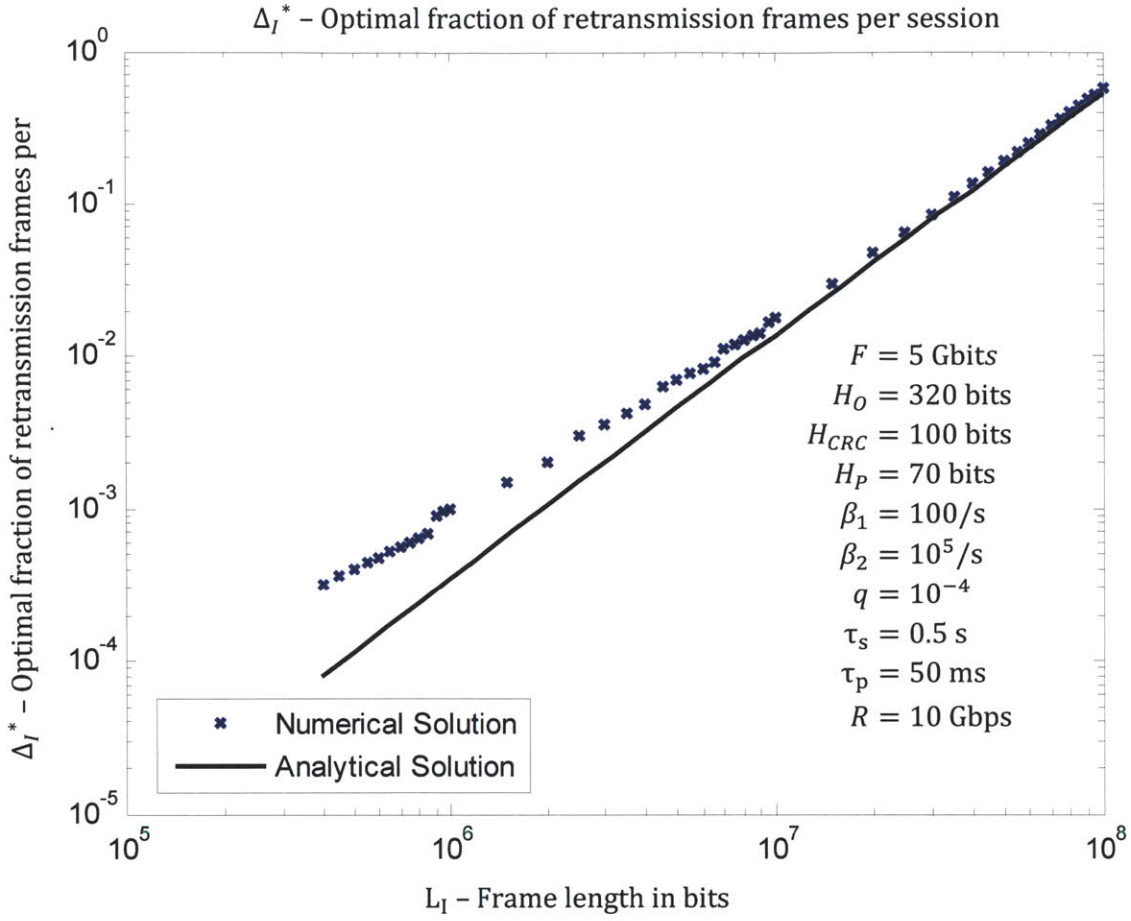


Figure 5-15: TPFI optimal fraction of retransmission frames per session - Δ_I^* vs. frame length. "Numerical Solution" corresponds to solving (5.72) numerically. "Analytical Solution" corresponds to the expression in (5.73).

5.2.3.2 Optimal Frame Length

As in Section 5.1.3.2, we assume that $\beta_1 L_I/R \ll 1$ near the optimal frame length. $L_I \cong D/C_q$ for $L_I \cong L_I^*$. We can approximate p_I by the first two terms of the Maclaurin series expansion where

$$\begin{aligned} p_I &= 1 - e^{-\frac{\beta_1 L_I}{R}} - \frac{\beta_2}{\beta_1+\beta_2}\frac{\beta_1 L_I}{R} e^{-\frac{\beta_1 L_I}{R}} \\ &\cong \frac{\beta_1 L_I}{R} - \frac{\beta_2}{\beta_1+\beta_2}\left(\frac{\beta_1 L_I}{R} - \left(\frac{\beta_1 L_I}{R}\right)^2\right) \\ &= \frac{{\beta_1}^2 L_I}{R(\beta_1+\beta_2)}\left(1+\frac{\beta_2 L_I}{R}\right) \\ &\cong \left(\frac{\beta_1}{RC_\xi}\right)^2 \frac{\beta_2 D^2}{\beta_1+\beta_2}. \end{aligned} \tag{5.74}$$

We notice that the probability of an erroneous frame is small near the optimal frame length. Thus, we can approximate

$$1 - p \cong 1. \tag{5.75}$$

We substitute (5.74), (5.75), and $L_I \cong D/C_q$ into (5.73) where

$$\Delta_I^* \cong \left(\frac{\beta_1}{RC_\xi}\right)^2 \frac{\beta_2 D^2}{\beta_1+\beta_2} - \frac{D}{F} + D\sqrt{\frac{D}{F}\left(\frac{\beta_1}{RC_\xi}\right)^2 \frac{\beta_2}{\beta_1+\beta_2} \ln\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2 (C_\xi R)^4 (\beta_1+\beta_2)}{4F\beta_2 D^3}\right)}. \tag{5.76}$$

The total delay expression can be approximated as in (5.36). However, in TPFI, interleaving results in an additional transmission delay. In our analysis, we assume that interleaving is performed across a single frame rather than across multiple frames. Thus, we can approximate the delay introduced by interleaving as an additional frame delay. The total expected delay is

$$E[T_I] \cong \frac{F}{D}\frac{L_I}{R}(1+\Delta_I) + \tau_s + \tau_p + \frac{L_I}{R}. \tag{5.77}$$

We substitute (5.76) into (5.77). The resulting delay expression is approximated as

$$E[T_I] \cong \frac{F+D}{D}\frac{D+H+C_\xi H_P}{C_\xi R}\left(1+\left(\frac{\beta_1}{RC_\xi}\right)^2\frac{\beta_2 D^2}{\beta_1+\beta_2}-\frac{D}{F}\right.$$
$$\left.+D\sqrt{\frac{D}{F}\left(\frac{\beta_1}{RC_\xi}\right)^2\frac{\beta_2}{\beta_1+\beta_2}\ln\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2(C_\xi R)^4(\beta_1+\beta_2)}{4F\beta_2 D^3}\right)}\right)+\tau_s+\tau_p. \quad (5.78)$$

We differentiate the total expected delay with respect to the message length where

$$\frac{\partial E[T_I]}{\partial D}=\frac{\partial}{\partial D}\left(\frac{F+D}{D}\frac{D+H+C_\xi H_P}{C_\xi R}\right)(1+\Delta_I)+\frac{F+D}{D}\frac{D+H+C_\xi H_P}{C_\xi R}\frac{\partial \Delta_I}{\partial D}=0. \quad (5.79)$$

Equation (5.79) is a transcendental equation and does not lead to a closed-form solution for L_I^*. Table 5-7 shows the numerical solutions for the optimal frame size.

Optimal Frame Length L^*	
Actual - Binomial	750 Kbits
Approximation –Solve	800 Kbits
Parameters	**Value**
β_1	100
β_2	10^5
R	10 Gbps
H_O	320 bits
H_{CRC}	100 bits
H_P	70 bits
F	5 Gbit
τ_s	500 ms
τ_p	50 ms
q	10^{-4}

Table 5-7: TPFI numerical solutions for the optimal frame length (L_I^*).

5.2.3.3 Practical Frame Length

As in TPF, in TPFI, a large frame length may be desirable to minimize the overhead associated with segmentation and reassembly of a file. We assume that near the practical frame length, $L_I \cong D/C_\xi$. We further approximate the delay expression in (5.77) as

$$E[T_I] \cong \frac{F}{C_\xi R}(1+\Delta_I) + \tau_s + \tau_p + \frac{D}{C_\xi R}. \tag{5.80}$$

The practical frame length has an expected delay ε away from the optimal delay where

$$E[T_\varepsilon^I] = (1+\varepsilon)E[T_I{}^*]. \tag{5.81}$$

We solve (5.81) for the practical frame length where

$$\begin{aligned} E[T_\varepsilon^I] \cong \frac{F}{C_\xi R}\Bigg(& 1 + \left(\frac{\beta_1}{RC_\xi}\right)^2 \frac{\beta_2 D^2}{\beta_1+\beta_2} - \frac{D}{F} \\ & + D\sqrt{\frac{D}{F}\left(\frac{\beta_1}{RC_\xi}\right)^2 \frac{\beta_2}{\beta_1+\beta_2} \ln\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2 (C_\xi R)^4(\beta_1+\beta_2)}{4F\beta_2 D^3}\right)}\Bigg) + \tau_s + \tau_p + \frac{D}{C_\xi R} \end{aligned} \tag{5.82}$$

$$\left(\left(E[T_\varepsilon^I] - \tau_s - \tau_p\right)\frac{C_\xi R}{F} - 1\right)\frac{RC_\xi}{\beta_1}$$

$$= \frac{\beta_1}{RC_\xi}\frac{\beta_2 D^2}{\beta_1+\beta_2} + D\sqrt{\frac{D}{F}\frac{\beta_2}{\beta_1+\beta_2}\ln\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2 (C_\xi R)^4(\beta_1+\beta_2)}{4F\beta_2 D^3}\right)}.$$

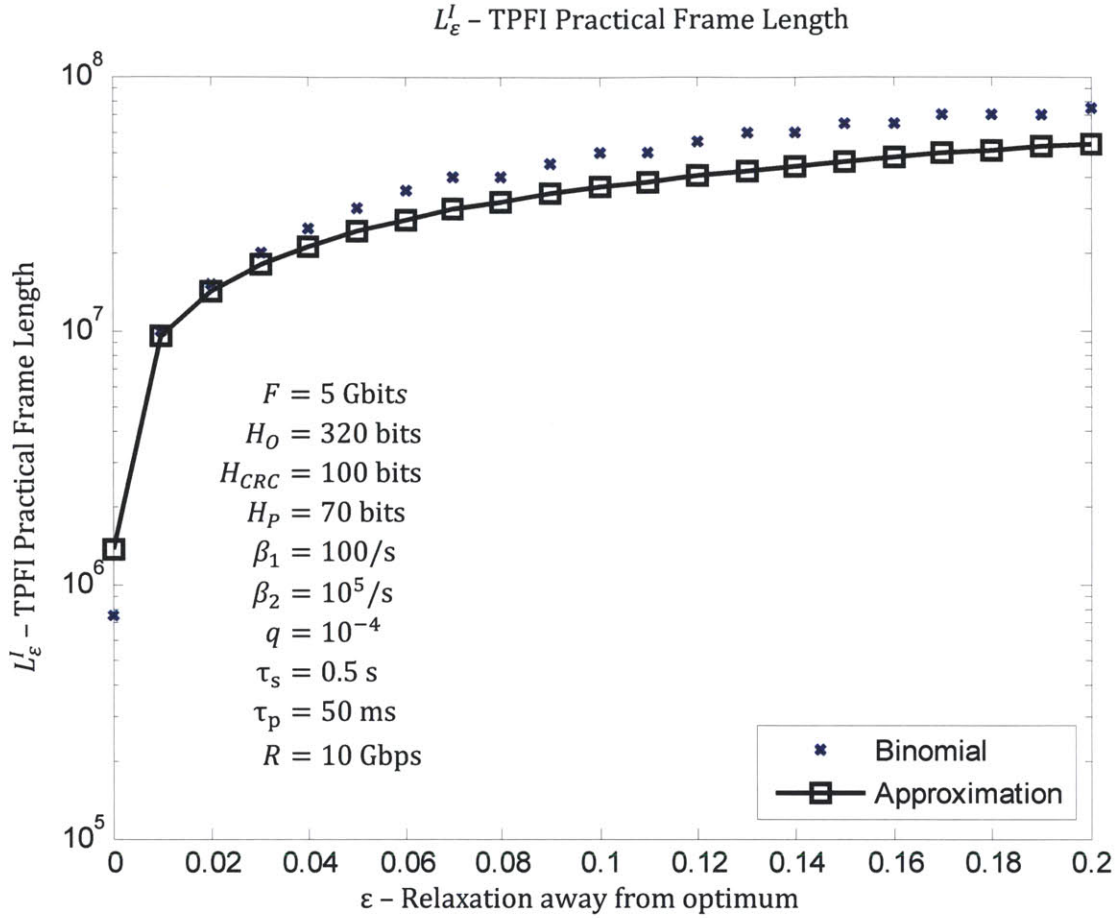


Figure 5-16: TPFI practical frame length vs. ε. "Binomial" corresponds to solving (5.72) numerically. "Analytical Solution" corresponds to numerically solving the expression in (5.82).

Numeric solutions for the practical frame length are shown in Figure 5-16. We provide an upper and lower bounds for the maximum frame size in Table 5-8.

Lower Bound for L_ε:

The right hand side of (5.82) can be bounded by

$$\frac{\beta_1}{RC_\xi}\frac{\beta_2 D^2}{\beta_1+\beta_2}+D>\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}. \tag{5.83}$$

This leads to a lower bound for the practical frame length where

$$L_\varepsilon^I>\frac{-1+\sqrt{1+\frac{4\beta_2}{\beta_1+\beta_2}\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)}}{2\frac{\beta_1}{RC_\xi}\frac{\beta_2}{\beta_1+\beta_2}}. \tag{5.84}$$

Upper Bound for L_ε:

The right hand side of (5.82) can be bounded by

$$\frac{\beta_1}{RC_\xi}\frac{\beta_2 D^2}{\beta_1+\beta_2}+\frac{D^2}{F}\frac{\beta_2}{\beta_1+\beta_2}\ln\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2\left(C_\xi R\right)^4(\beta_1+\beta_2)}{4F\beta_2 D^3}\right) \tag{5.85}$$

$$<\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}.$$

This leads to an upper bound for the practical frame length where

$$L_\varepsilon^I\le\sqrt{-\frac{\frac{2F(\beta_1+\beta_2)}{3\beta_2}\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}}{\mathrm{W}_{-1}\left[-\frac{2\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}e^{-\frac{2F\beta_1}{3RC_\xi}}}{\frac{3}{F}\frac{\beta_2}{\beta_1+\beta_2}\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2\left(C_\xi R\right)^4(\beta_1+\beta_2)}{4F\beta_2}\right)^{2/3}}\right]}}. \tag{5.86}$$

Table 5-8: TPFI upper and lower bounds for the maximum frame size.

The upper and lower bounds for the maximum frame size are plotted in Figure 5-17.

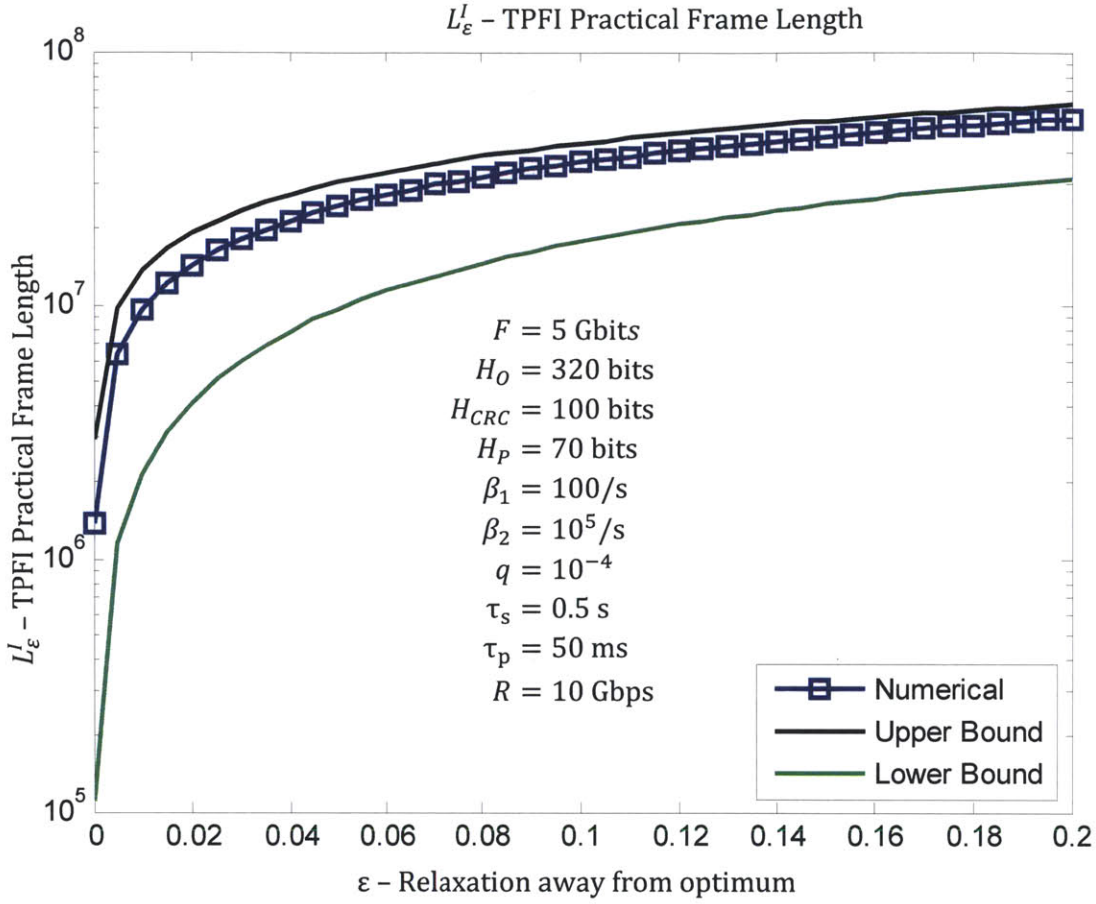


Figure 5-17: TPFI practical frame length: upper and lower bounds.

5.3 Transport Protocol with Framing and Interleaving (TPFI): FEC corrects Γ burst errors

In Section 5.2, we assumed that frames are of finite length, thus perfect interleaving cannot be achieved. In addition, we assumed that up to one burst error can be corrected with FEC. In this section, we assume that FEC can correct up to Γ burst errors per frame. In general, we pick Γ greater than the expected number of bursts in one frame. If a frame experiences more than Γ burst errors during transmission, the frame is uncorrectable and must be discarded. We assume that interleaving is performed across a single frame rather than across multiple frames, since the reason for having frames is for retransmission purposes, thus the maximum number of burst errors that can be corrected per frame is correlated with the frame length.

5.3.1 Error Probabilities

In this section, we define the two error probabilities: p_Γ, the probability of an erroneous frame and θ_Γ, the probability of a failed initial transmission. Let D be the number of message bits per frame and L_I be the total number of bits per frame as defined in Section 5.2.1. The probability of an erroneous frame is the probability of more than Γ outage periods in a frame time where

$$\begin{aligned} p_\Gamma &= 1 - \sum_{i=0}^{\Gamma-1} P(i \text{ bursts}) - P(\Gamma \text{ bursts} | S_0 = \text{non-outage}) P(S_0 = \text{ non-outage}) \\ &= 1 - e^{-\frac{\beta_1 L_I}{R}} \sum_{i=0}^{\Gamma-1} \frac{\left(\frac{\beta_1 L_I}{R}\right)^i}{i!} - \frac{\beta_2}{\beta_1 + \beta_2} e^{-\frac{\beta_1 L_I}{R}} \frac{\left(\frac{\beta_1 L_I}{R}\right)^\Gamma}{\Gamma!}. \end{aligned} \tag{5.89}$$

The expression for θ_Γ follows from (5.55) where

$$\theta_\Gamma = \sum_{k=0}^{n-1} \binom{\lfloor n(1+\Delta) \rfloor}{k} (1 - p_\Gamma)^k p_\Gamma^{\lfloor n(1+\Delta) \rfloor - k}. \tag{5.90}$$

5.3.2 Performance Optimization - Throughput

Let η_Γ be the expected TPFI throughput. The derivation of the expected throughput follows from Section 5.1.2 where

$$\begin{aligned} \eta_\Gamma &= (1 - p_\Gamma) \frac{D}{L_I} \\ &= e^{-\frac{\beta_1 L_I}{R}} \left(\sum_{i=0}^{\Gamma-1} \frac{\left(\frac{\beta_1 L_I}{R}\right)^i}{i!} - \frac{\beta_2}{\beta_1 + \beta_2} \frac{\left(\frac{\beta_1 L_I}{R}\right)^\Gamma}{\Gamma!} \right) \frac{D}{L_I}. \end{aligned} \tag{5.91}$$

Substituting the expression of L_I into (5.91) results in an expression for the expected throughput as a function of the message size, D. The optimal frames length can be found by solving for when the derivative of η_Γ with respect to D is zero.

Let η_Γ^* be the optimal throughput. In Figure 5-18, we show numerical solutions for the maximum expected throughput for different values of Γ. We notice that as the number of correctable burst errors increases, so does the optimal frame length and maximum expected throughput. This is to be expected if we recognize that as the frame size increases, the expected number of bursts also increases linearly with the frame size. With optimum coding, the same code rate should be able to correct for the same fraction of burst error assuming the interleaver size is large enough. In fact, the interleaver size does not have to be much bigger than the interarrival time of the bursts; perhaps of the order of 1-3 times the interarrival times. Figure 5-18 shows that almost 100% of the achievable throughput is attained when the frame size is ~2 x the interarrival time of burst errors.

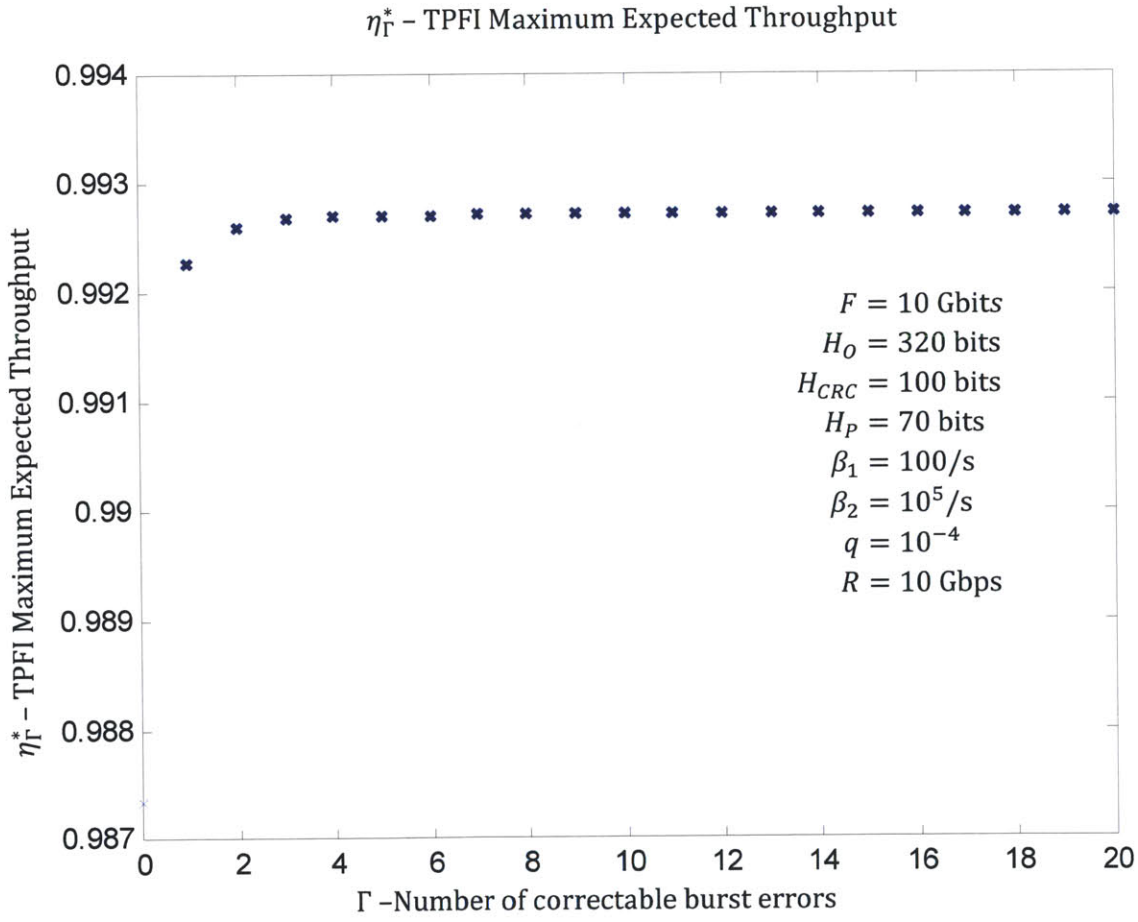


Figure 5-18: Maximum expected throughput vs. Γ (number of correctable burst errors).

Let $D_\Gamma{}^*$ be the length of the message at the optimal frame length. Let $L_\Gamma{}^*$ be the optimal frame length. We conjecture that the expected number of burst errors and the number of correctable

burst errors at the optimal frame length asymptotically approaches a constant ϖ with increasing Γ where

$$\lim_{\Gamma \to \infty} \frac{L^*(\Gamma)\beta_1}{R\Gamma} = \varpi. \tag{5.92}$$

Thus, we can approximate $L_\Gamma{}^*$ as a linear function of Γ where

$$L_\Gamma{}^* \cong \frac{\varpi R}{\beta_1}\Gamma. \tag{5.93}$$

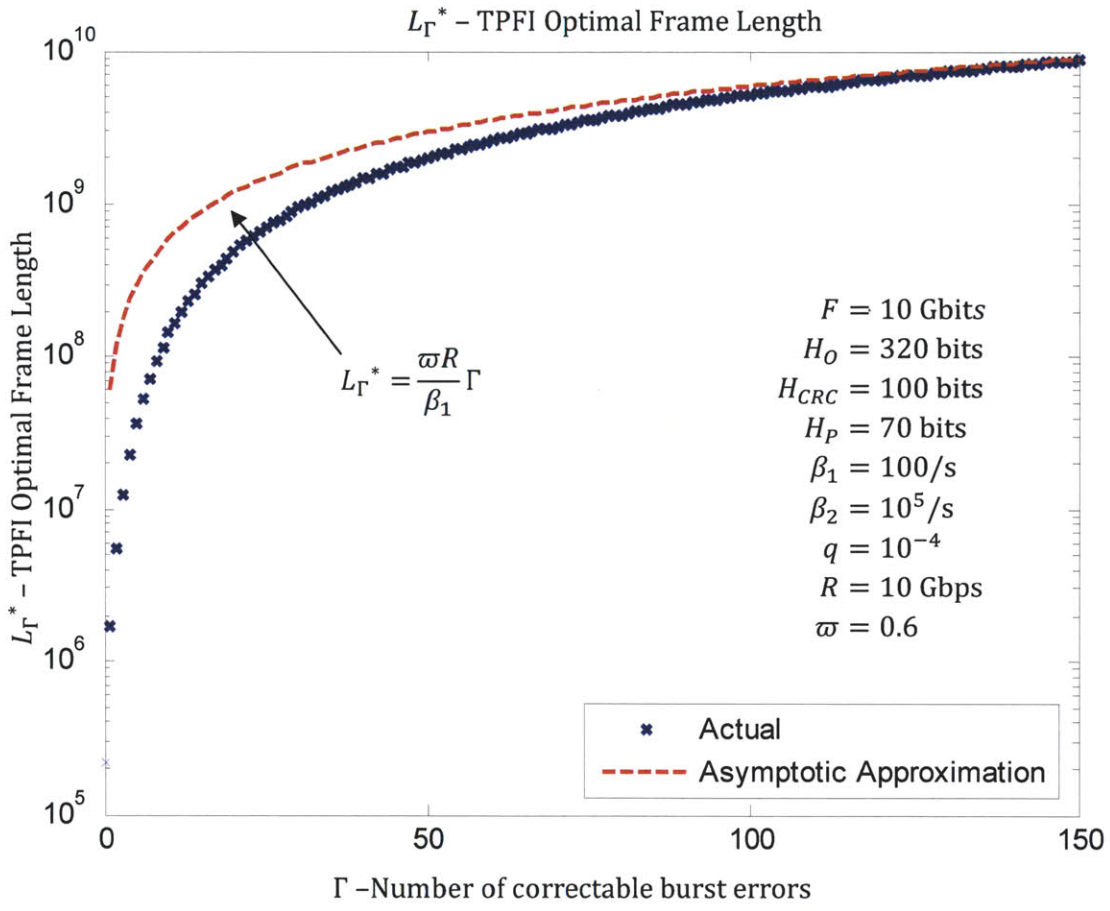


Figure 5-19: TPFI optimal frame length vs. Γ (number of correctable burst errors).

In Figure 5-19, we show numerical solutions for the optimal frame length and the asymptotic expression seen in (5.93). For the parameters chosen, we observe that $\varpi \cong 0.6$. The exact expression for ϖ, however, requires further research.

Let η_ε^Γ be the expected throughput ε away from the optimal expected throughput. At the practical frame length,

$$\begin{aligned}\eta_\varepsilon^\Gamma &= (1-\varepsilon)\eta_\Gamma{}^* \\ &= e^{-\frac{\beta_1 L_I}{R}}\left(\sum_{i=0}^{\Gamma-1}\frac{\left(\frac{\beta_1 L_I}{R}\right)^i}{i!} - \frac{\beta_2}{\beta_1+\beta_2}\frac{\left(\frac{\beta_1 L_I}{R}\right)^\Gamma}{\Gamma!}\right)\frac{D}{L_I}.\end{aligned} \tag{5.94}$$

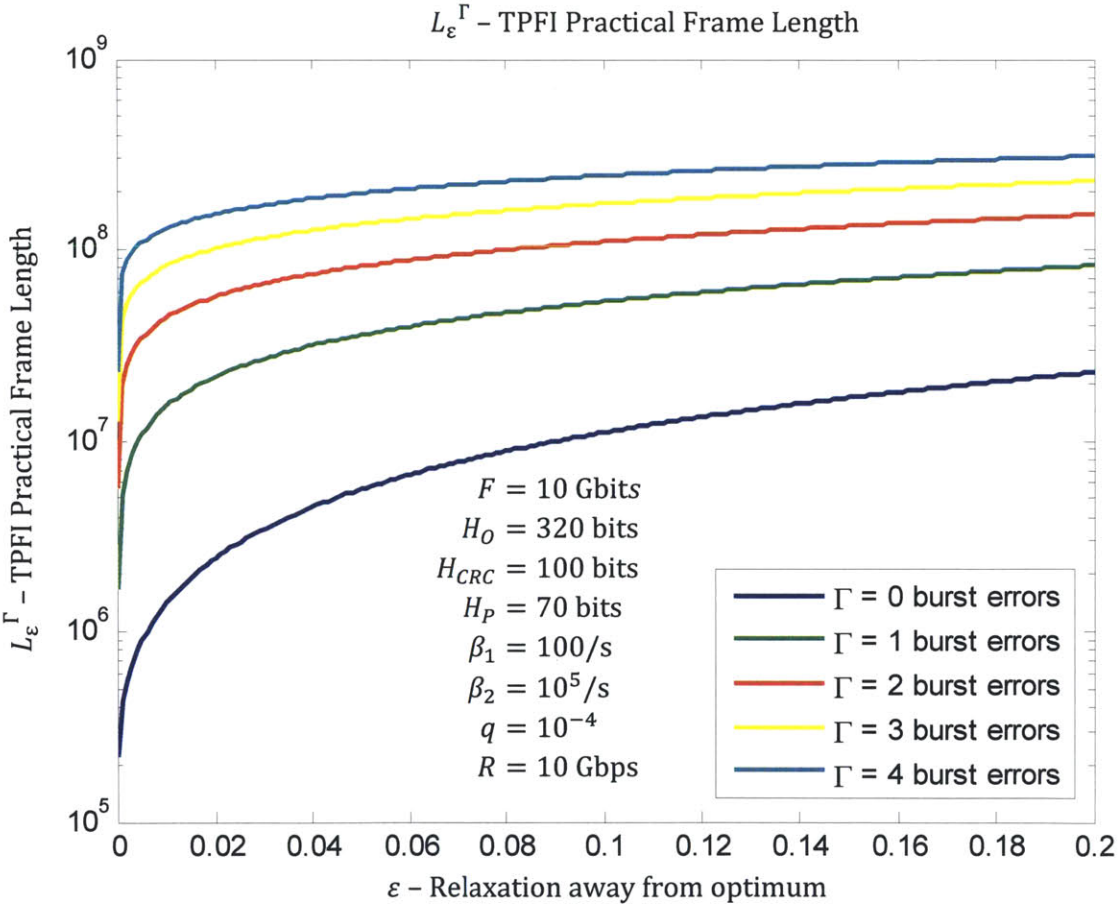


Figure 5-20: TPFI practical frame length vs. ε.

Let $L_\varepsilon{}^\Gamma$ be the practical frame length. We numerically solve (5.94) for $L_\varepsilon{}^\Gamma$. The practical frame length plotted against epsilon is shown in Figure 5-20. We conjecture that the expected number of burst errors and the number of correctable burst errors at the practical frame length also asymptotically approaches a constant $\not{f}$ with increasing Γ. Thus, we can approximate $L_\varepsilon{}^\Gamma$ as a linear function of Γ where

$$L_\varepsilon{}^\Gamma \cong \frac{\not{f}R}{\beta_1}\Gamma. \tag{5.95}$$

In Figure 5-21, we plot the practical frame length against the number of correctable burst errors for a fixed ε. We also plot the asymptotic approximation of $L_\varepsilon{}^\Gamma$ for $\not{f} = 0.9$ in Figure 5-21. The exact expression for $\not{f}$, requires further research.

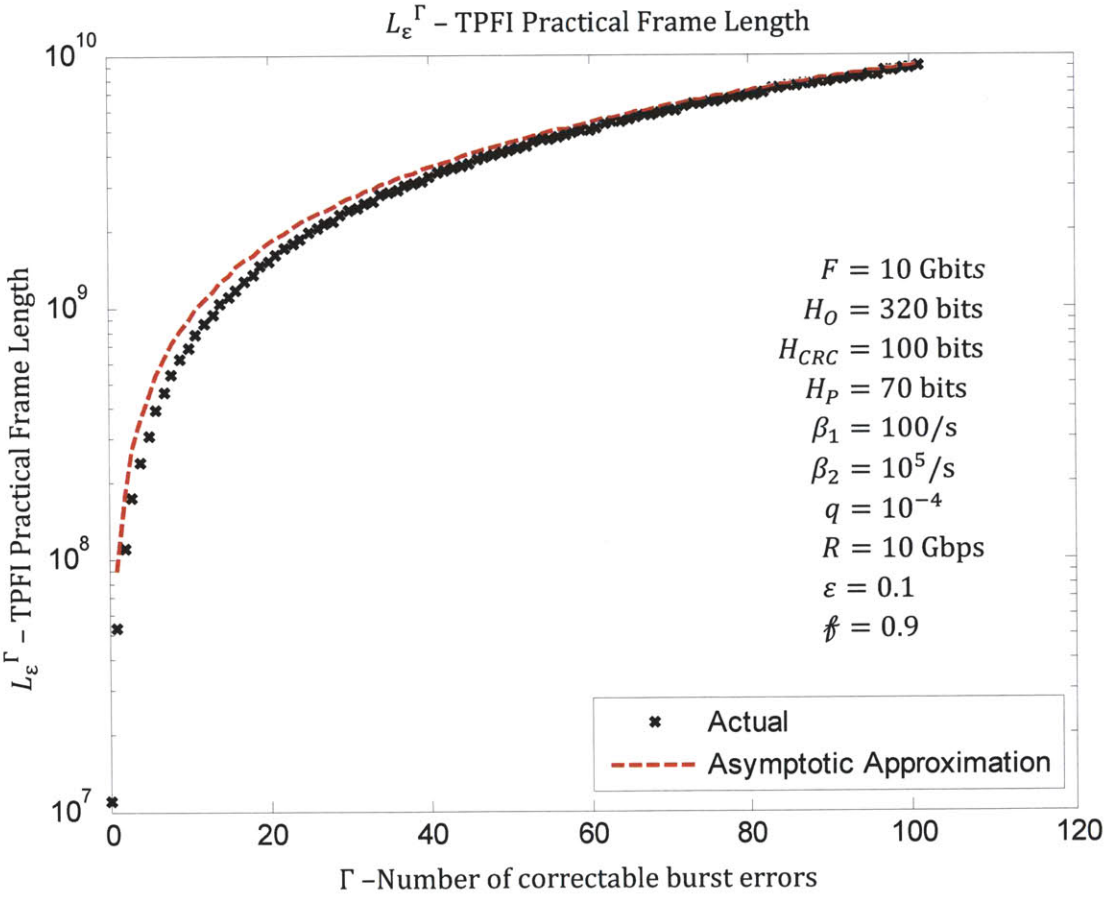


Figure 5-21: TPFI practical frame length vs. Γ (number of correctable burst errors).

5.3.3 Performance Optimization – Delay

Let $E[T_\Gamma]$ be the total expected delay. The derivation for $E[T_\Gamma]$ follows from Section 5.1.3. The expected total delay for no setup and propagation delay in given (5.96). The expected total delay for nonzero setup and propagation delay in given in (5.97).

$\tau_s = 0$ and $\tau_p = 0$	
$E[T_\Gamma] = \frac{\frac{F}{D}\frac{L_I}{R}}{1 - p_\Gamma}$	(5.96)

Table 5-9: TPFI (FEC corrects for Γ burst errors) total expected delay for no setup and propagation delay.

$\tau_s \neq 0$ and $\tau_p \neq 0$	
$p_{\Gamma_{i\backslash n}} = \binom{\lfloor n(1+\Delta)\rfloor}{n-i}(1-p_\Gamma)^{n-i}p_\Gamma{}^{\lfloor n(1+\Delta)\rfloor - n + i}$ $p_{\Gamma_{n\backslash n}} = \binom{\lfloor n(1+\Delta)\rfloor}{0}(1-p_\Gamma)^{0}p_\Gamma{}^{\lfloor n(1+\Delta)\rfloor} = p_\Gamma{}^{\lfloor n(1+\Delta)\rfloor}$ $E[T_\Gamma] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L_I}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{\Gamma_{i\backslash n}} E_i[T_\Gamma]}{1 - p_{\Gamma_{n\backslash n}}}$	(5.97)

Table 5-10: TPFI (FEC corrects for Γ burst errors) total expected delay for nonzero setup and propagation delay.

Let $E[X_1]$ be the expected number of outstanding erroneous frames after the initial session where

$$E[X_1] = \sum_{k=0}^{n-1}(n-k)\binom{\lfloor n(1+\Delta)\rfloor}{k}(1-p_\Gamma)^k p_\Gamma{}^{\lfloor n(1+\Delta)\rfloor - k}. \tag{5.98}$$

The expected total delay can be approximated as in (5.26). In RFPI, however, interleaving introduces an additional frame delay where

$$E[T_\Gamma] \cong n(1+\Delta)\frac{L_I}{R} + \tau_s + \tau_p + \frac{L_I}{R} + \theta_\Gamma\left(E[X_1]\frac{L_I}{R} + \tau_s + \tau_p + \frac{L_I}{R}\right). \tag{5.99}$$

We note that the approximations made in (5.99) hold only if the probability that a new session is needed for frame retransmissions is small. Substituting the expression for L_I in (5.52), the expression for n as a function of D, the expression for θ_Γ in (5.90), and the expression for $E[X_1]$ in (5.98) into

(5.99) results in an expression for the expected total delay as a function of the message length, D. Let $E[T_\varepsilon^\Gamma]$ be the delay ε away from the optimal expected delay where

$$\begin{aligned} E[T_\varepsilon^\Gamma] &= (1+\varepsilon)E[T_\Gamma{}^*] \\ &\cong n(1+\Delta)\frac{L_I}{R} + \tau_s + \tau_p + \frac{L_I}{R} + \theta_\Gamma\left(E[X_1]\frac{L_I}{R} + \tau_s + \tau_p + \frac{L_I}{R}\right). \end{aligned} \tag{5.100}$$

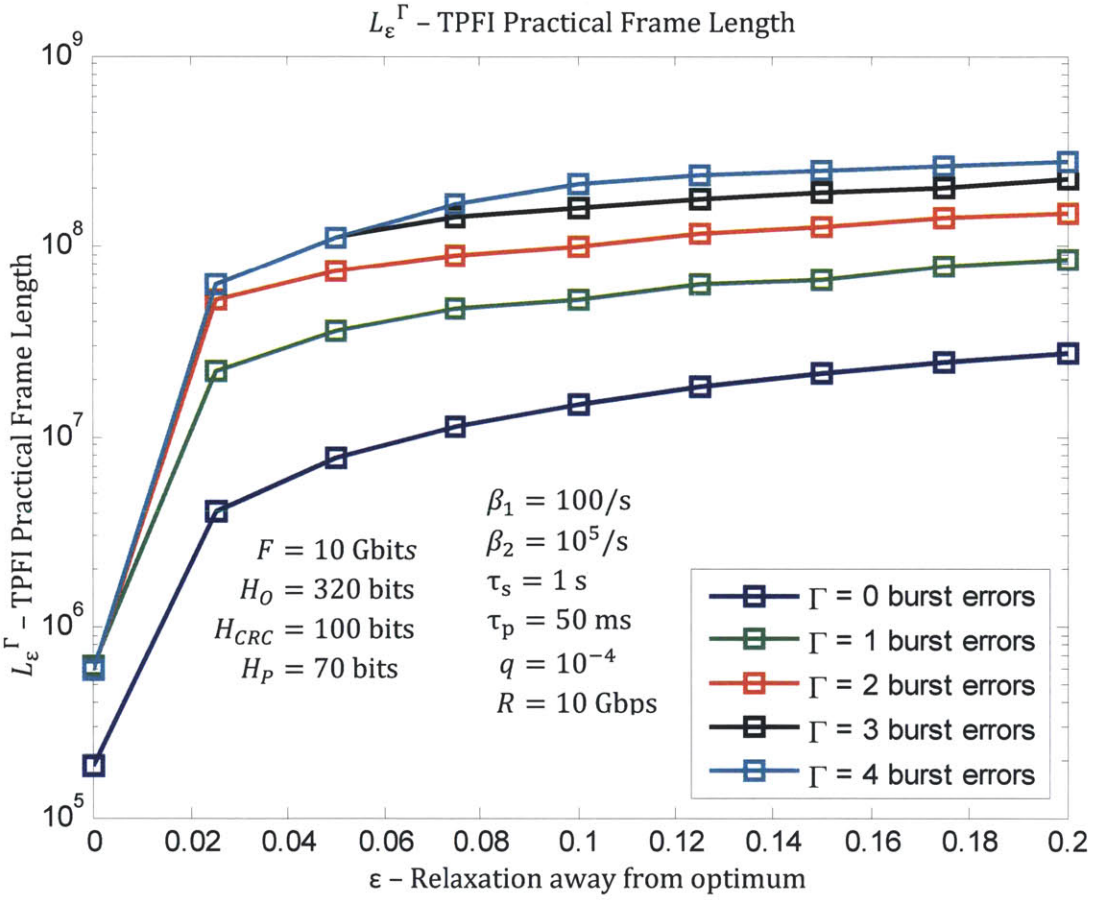


Figure 5-22: TPFI practical frame length with different values of Γ.

We numerically solve (5.100) for $L_\varepsilon{}^\Gamma$. The practical frame length plotted against epsilon is shown in Figure 5-22. The determination of the best frame size is important for OFS. On the one hand we want the frame size to be big to minimize the overhead of segmentation and reassembly. On the other hand we want the frame size not to be too large so that the interleaver size is too big and requires too much memory and when frame retransmission is needed, the frame size is not too large

and require a large transmission time. From the result of the last two chapters, the best choice of frame size is around 10^8 bits for the system parameters indicated.

Chapter 6

6 Conclusion

6.1 Summary of Results

In this section, we summarize the throughput and delay results for the four example OFS protocols presented in this work. Table 6-1 summarizes the expected throughput and delay expressions for STP, STPI, TPF, and TPFI.

Throughput	
STP (Section 4.1)	$\mathcal{G} = \frac{F}{W}\frac{\beta_2}{\beta_1+\beta_2}e^{-\beta_1\frac{W}{R}}$
STPI (Section 4.2)	$\mathcal{G}_I \geq \max_{0\leq U< C_\xi} \frac{F}{N}\left(1-2^{-NE(U)}\right)$
TPF (Section 5.1)	$\eta = \frac{D}{L}\frac{\beta_2}{\beta_1+\beta_2}e^{-\frac{\beta_1 L}{R}}$
TPFI (Section 5.2) FEC corrects for 1 burst error	$\eta_I = \frac{D}{L_I}e^{-\frac{\beta_1 L_I}{R}}\left(1+\frac{\beta_2}{\beta_1+\beta_2}\frac{\beta_1 L_I}{R}\right)$
TPFI (Section 5.3) FEC corrects for Γ burst errors	$\eta_\Gamma = e^{-\frac{\beta_1 L_I}{R}}\left(\sum_{i=0}^{\Gamma-1}\frac{\left(\frac{\beta_1 L_I}{R}\right)^i}{i!}-\frac{\beta_2}{\beta_1+\beta_2}\frac{\left(\frac{\beta_1 L_I}{R}\right)^\Gamma}{\Gamma!}\right)\frac{D}{L_I}$
Delay	
STP (Section 4.1)	$E[\mathcal{T}] = \left(\tau_s+\frac{W}{R}\right)\frac{\beta_1+\beta_2}{\beta_2}e^{\beta_1\frac{W}{R}}$
STPI (Section 4.2)	$E[\mathcal{T}_I] \leq \min_{0\leq U< C_\xi}\frac{\tau_s+\frac{N}{R}}{1-2^{-NE(U)}}$
TPF (Section 5.1)	$E[T] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L}{R}+\tau_s+\tau_p+\sum_{i=1}^{n-1}p_{i\backslash n}E_i[T]}{1-p_{n\backslash n}}$
TPFI (Section 5.2) FEC corrects for 1 burst error	$E[T_I] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L_I}{R}+\tau_s+\tau_p+\sum_{i=1}^{n-1}p_{I\,i\backslash n}E_i[T_I]}{1-p_{I\,n\backslash n}}$
TPFI (Section 5.3) FEC corrects for Γ burst errors	$E[T_\Gamma] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L_I}{R}+\tau_s+\tau_p+\sum_{i=1}^{n-1}p_{\Gamma\,i\backslash n}E_i[T_\Gamma]}{1-p_{\Gamma\,n\backslash n}}$

Table 6-1: Summary of throughput and delay expressions for STP, STPI, TPF, and TPFI.

In Figure 6-1, we plot the expected throughput for STP, STPI, TPF, and TPFI.

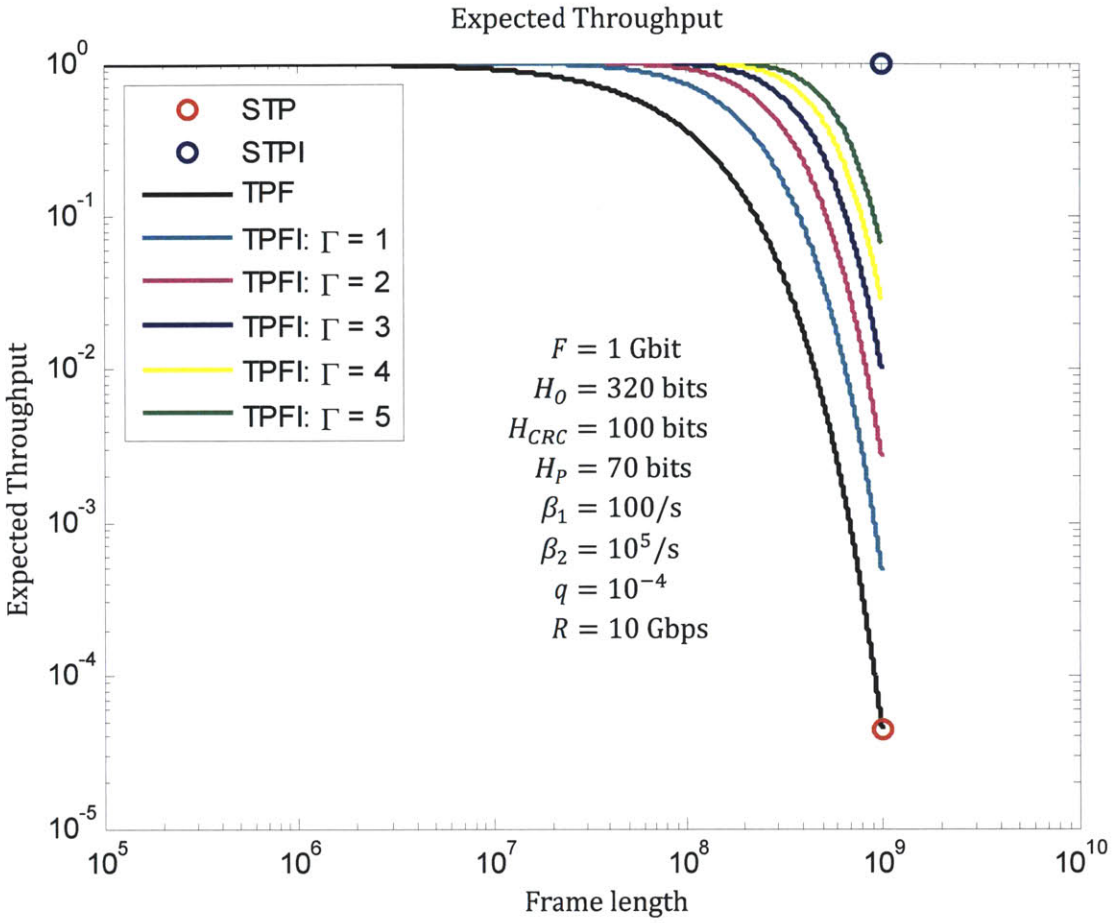


Figure 6-1: STP, STPI, TPF, and TPFI expected throughput vs. frame length. For STP, the frame length is the entire file plus overhead - W (Section 4.1). For STPI, the frame length is the entire file plus overhead - N (Section 4.2). For TPI and TPFI the frame lengths are L (Section 5.1) and L_I (Section 5.2) respectively.

In Figure 6-2, we plot the expected throughput for STP, STPI, TPF, and TPFI as in Figure 6-1, but with a rescaled axes.

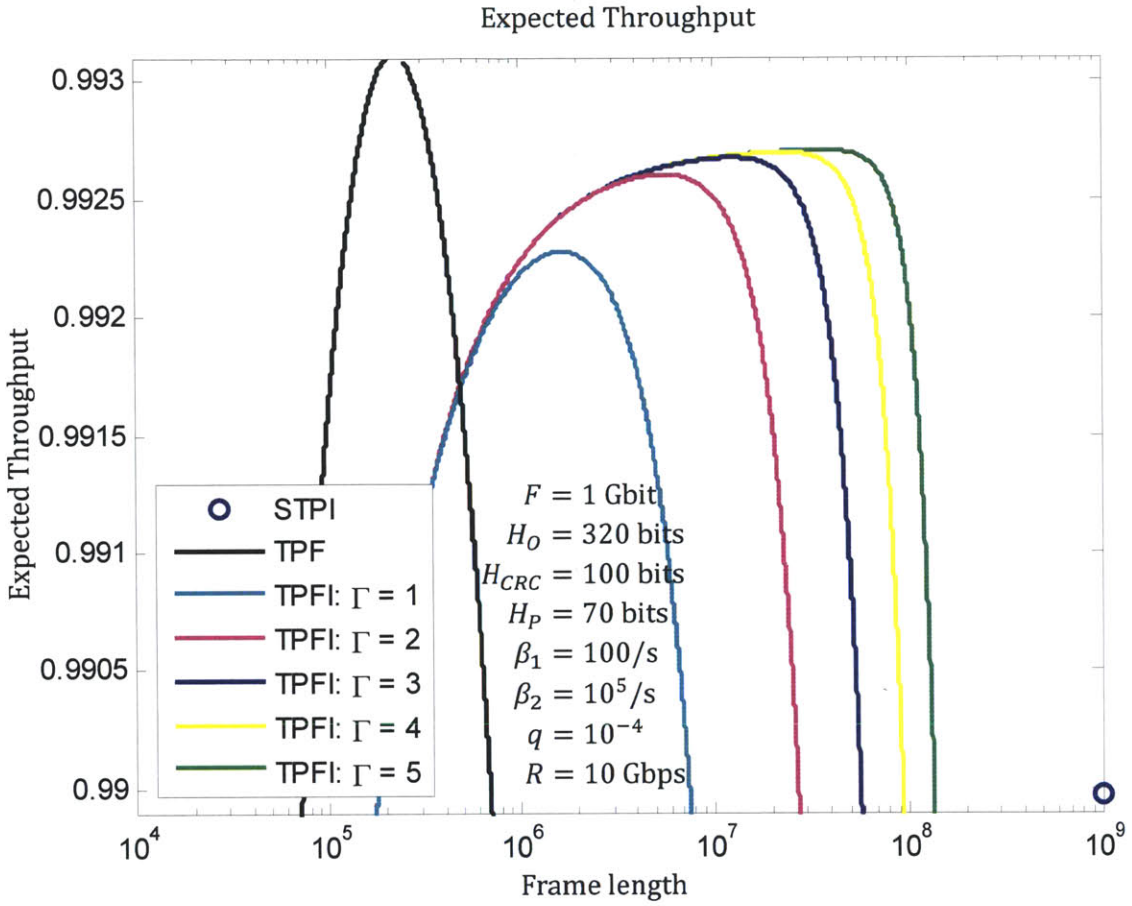


Figure 6-2: STP, STPI, TPF, and TPFI expected throughput vs. frame length with rescaled axes.

In , Figure 6-3, we plot the normalized expected total delay for STP, STPI, TPF, and TPFI: Λ where

$$\Lambda = \frac{R}{F} E[\text{delay}]. \tag{5.101}$$

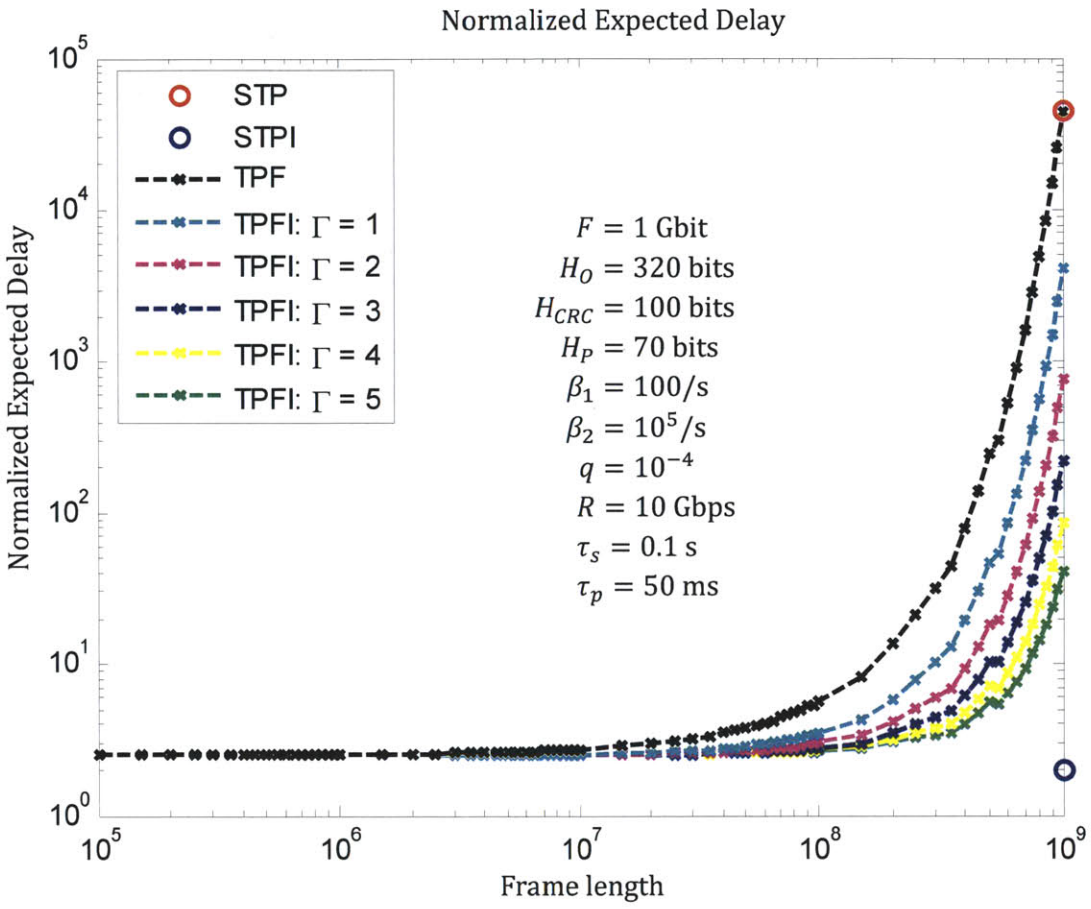


Figure 6-3: STP, STPI, TPF, and TPFI normalized expected delay vs. frame length. For STP, the frame length is the entire file plus overhead - W (Section 4.1). For STPI, the frame length is the entire file plus overhead - N (Section 4.2). For TPI and TPFI the frame lengths are L (Section 5.1) and L_I (Section 5.2) respectively.

In Figure 6-4, we plot the normalized expected delay for STP, STPI, TPF, and TPFI as in Figure 6-3, but with a rescaled axes.

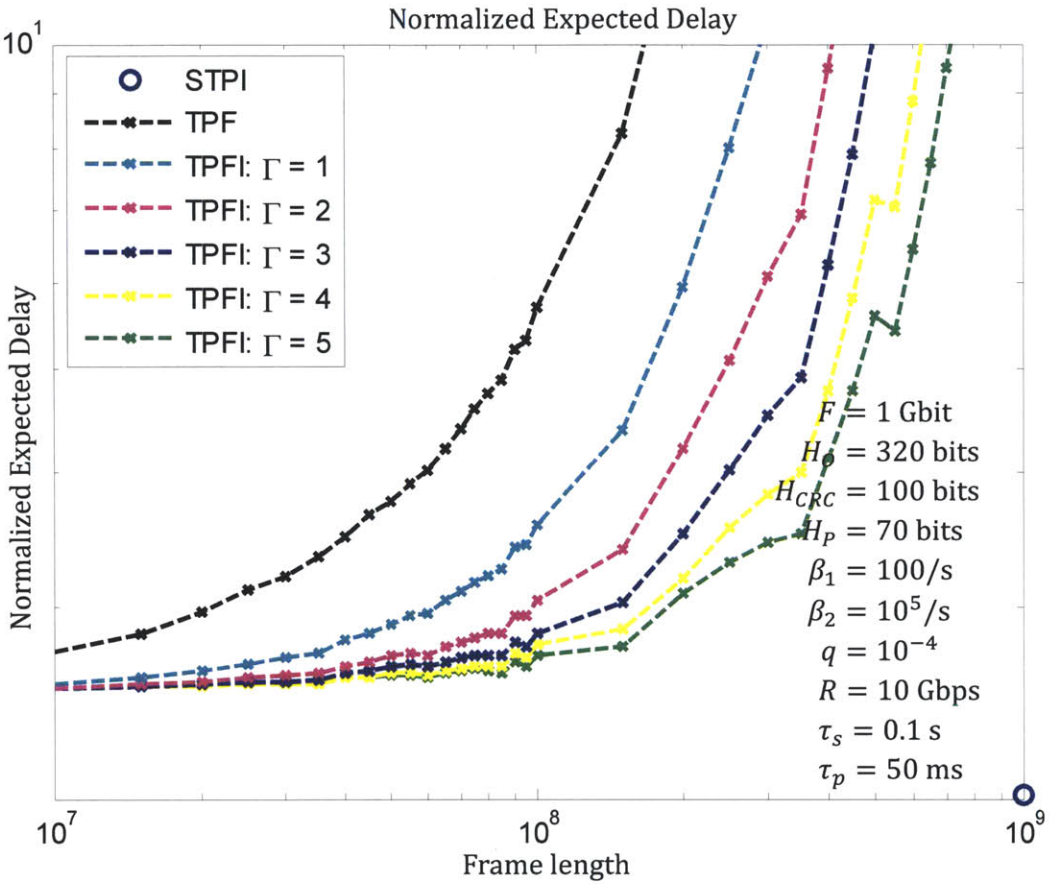


Figure 6-4: STP, STPI, TPF, and TPFI normalized expected delay vs. frame length with rescaled axes.

Table 6-2 summarizes the expressions for the optimum frame length for TPF and TPFI. Table 6-3 summarizes the expressions for the practical frame length for TPF and TPFI with respect to throughput.

Optimum Frame Length with respect to Throughput	
TPF (Section 5.1)	$L^* = \frac{1}{C_q}\sqrt{(H + C_q H_P)\left(\frac{H + C_q H_P}{4} + \frac{C_q R}{\beta_1}\right)} + \frac{1}{2}\left(H_P + \frac{H}{C_q}\right)$
TPFI (Section 5.2) FEC corrects for 1 burst error	$L_I^* = -\frac{b}{3aC_\xi} - \frac{1}{3aC_\xi}\sqrt[3]{\frac{1}{2}\left(2b^3 - 9abc + 27a^2d + \sqrt{(2b^3 - 9abc + 27a^2d)^2 - 4(b^2 - 3ac)^3}\right)} - \frac{1}{3aC_\xi}\sqrt[3]{\frac{1}{2}\left(2b^3 - 9abc + 27a^2d - \sqrt{(2b^3 - 9abc + 27a^2d)^2 - 4(b^2 - 3ac)^3}\right)} + \frac{H}{C_\xi} + H_P$ where $a = -\frac{\beta_2 {\beta_1}^2}{\beta_1 + \beta_2}$ $b = \beta_1\left(\frac{\beta_2}{\beta_1 + \beta_2}\left(C_\xi R - 2\beta_1(H + C_\xi H_P)\right) - C_\xi R\right)$ $c = (H + C_\xi H_P)\beta_1\left(\frac{\beta_2}{\beta_1 + \beta_2}\left(2C_\xi R - \beta_1(H + C_\xi H_P)\right) - C_\xi R\right)$ $d = (H + C_\xi H_P)C_\xi R\left(C_\xi R + \frac{\beta_2\beta_1(H + C_\xi H_P)}{\beta_1 + \beta_2}\right)$
TPFI (Section 5.3) FEC corrects for Γ burst errors	$L_\Gamma^* \cong \frac{\varpi R}{\beta_1}\Gamma$

Table 6-2: Summary of the optimal frame length expressions with respect to throughput for TPF and TPFI.

Practical Frame Length with respect to Throughput	
TPF (Section 5.1)	$L_\varepsilon \cong \frac{R}{\beta_1}\varepsilon + \frac{R}{\beta_1} ln\left(\frac{C_q}{\eta^*}\frac{\beta_2}{\beta_1 + \beta_2}\right)$
TPFI (Section 5.2) FEC corrects for 1 burst error	$L_\varepsilon^I \cong -\frac{R}{\beta_1}\mathrm{W}_{-1}\left[-\frac{e^{-\frac{\beta_1+\beta_2}{\beta_2}}(1-\varepsilon)\eta_I^*}{C_\xi\frac{\beta_2}{\beta_1+\beta_2}}\right] - \frac{R}{\beta_1}\frac{\beta_1+\beta_2}{\beta_2}$
TPFI (Section 5.3) FEC corrects for Γ burst errors	$L_\varepsilon{}^\Gamma \cong \frac{\not{\varpi} R}{\beta_1}\Gamma$

Table 6-3: Summary of the practical frame length expressions with respect to throughput for TPF and TPFI.

Table 6-4 summarizes the expressions for the practical frame length for TPF and TPFI with respect to delay.

Practical Frame Length with respect to Delay	
TPF (Section 5.1)	$L_\varepsilon \geq -\dfrac{\frac{RC_q}{F}\left(E[T_\varepsilon]-\tau_s-\tau_p\right)-1}{2\sqrt{\frac{\beta_1}{FC_qR}}\,W_{-1}\left(-\frac{E[T_\varepsilon]-\tau_s-\tau_p-\frac{F}{C_qR}}{\tau_s+\tau_p}\sqrt{e^{-\sqrt{\frac{FC_qR}{\beta_1}}\left(\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}\right)}}\right)}$ $L_\varepsilon \leq \dfrac{\frac{RC_q}{F}\left((1+\varepsilon)E[T]^*-\tau_s-\tau_p\right)-1}{\frac{\beta_1}{RC_q}-\frac{\beta_1+\beta_2}{F\beta_2}+\sqrt{\frac{\beta_1}{FC_qR}}}$
TPFI (Section 5.2) FEC corrects for 1 burst error	$L_\varepsilon^I \geq \dfrac{-1+\sqrt{1+\frac{4\beta_2}{\beta_1+\beta_2}\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)}}{2\frac{\beta_1}{RC_\xi}\frac{\beta_2}{\beta_1+\beta_2}}$ $L_\varepsilon^I \leq \sqrt{-\dfrac{\frac{2F(\beta_1+\beta_2)}{3\beta_2}\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}}{\mathrm{W}_{-1}\left[-\frac{2\left(\left(E[T_\varepsilon^I]-\tau_s-\tau_p\right)\frac{C_\xi R}{F}-1\right)\frac{RC_\xi}{\beta_1}e^{-\frac{2F\beta_1}{3RC_\xi}}}{\frac{3}{F}\frac{\beta_2}{\beta_1+\beta_2}\left(\frac{\left(\frac{\tau_s+\tau_p}{\beta_1}\right)^2(C_\xi R)^4(\beta_1+\beta_2)}{4F\beta_2}\right)^{2/3}}\right]}}$

Table 6-4: Summary of the practical frame length expressions with respect to delay for TPF and TPFI.

6.2 Conclusions

In this work, we designed a transport layer protocol for an OFS network. The objective of the protocol design is to guarantee the reliable delivery of data files over an all-optical end-to-end flow-switched network which is modeled as a burst-error channel. The OFS architecture is expected to serve large transactions (>100 Mbits). Thus, in our analysis, we only considered the transmission of large data files. Smaller file sizes are assumed to be transmitted via a data network with a TCP/IP architecture. The main contribution of this work is to optimize the throughput and delay performance of OFS using

file segmentation and reassembly, FEC, and frame retransmission. We find the optimum solutions are typically very gentle functions of the key parameters such as frame size and additional retransmission duration. Note also that we have not considered processing burden of the protocols in this thesis. Nonetheless, we can conjecture that if the frame sizes are larger with fewer number of frames per file, the segmentation and reassembly efforts will be smaller. Thus, we recommend choosing frame sizes and retransmission durations such that the frame sizes are larger than optimum but only sacrifice less than 10% of the throughput and delay performance.

Due to OFS session turn-on/turn-off transients and nonlinear effects in the fiber channel such as optical amplifier gain coupling and nonlinear propagation effects in the fiber, burst errors may be unavoidable even with the best compensation electronics. We modeled this channel as a Markov channel with interarrival times for the outages (burst errors) given by the turn-on/turn-off session times and outage durations given by the time the control electronics take to mitigate the transients.

At one extreme, we considered the Simple Transport Protocol (STP). In STP, an entire file is transmitted as one large frame. Interleaving is not used. Here we assumed that FEC can only correct for random errors but not the burst errors. In STP, the probability of an erroneous file is significant for large file sizes for any given rate of arrival of outages. As a result, both throughput and delay performance suffer with increasing file sizes. Thus, we conclude that STP is not well suited for OFS transactions. Instead, we suggest the need of a coding scheme that can correct for burst errors.

At the other extreme, we considered the Simple Transport Protocol with Interleaving (STPI). In STPI, interleaving is performed across the entire file at the channel input. We assumed that interleaving allows the channel noise affecting successive symbols in a codeword to be essentially independent [21]. This is a good assumption when the interleaver duration is longer than ~ 10 times the burst length. In order to reduce the average probability of error over the interleaved duration we will choose the frame size to be large to achieve a lower average error rate. However, after 1-3 interarrival times of the outages, the average error rate is close to an asymptote and there is little to be gained. In STPI, FEC is designed to correct for both random errors and burst errors. We invoke standard results in information theory to note that with increasing files sizes and thus code length, the

probability of an erroneous file can be made arbitrarily small [20][3]. Thus, we found that STPI resulted in throughput close to the highest throughput and highest theoretical average delay performance compared with the other transport protocols considered in this work, though this is due to the artifact that we let the code length become the size of the file. The error correcting code in this scheme with long code lengths has the effect of making the probability of having to perform a retransmission by setting up a new session to be very small and thus contribute little addition to the average delay or reduction of throughput. In practice it is dubious that code lengths of $>10^8$ has implementable decoding algorithms due to their complexity.

Interleaving across an entire file as in STPI, is impractical if the file size is very large. STPI may result in large processing delays and huge hardware burden associated with interleaving and coding over a large file. Thus, we propose the use of framing to allow for segments of a file to be processed as the file is being received. In this work, we considered two practical transport layer protocols in between the two extremes of STP and STPI: the Transport Protocol with Framing (TPF) and the Transport Protocol with Framing and Interleaving (TPFI) which have smaller code lengths equal to one frame size.

In TPF and TPFI, a large file is segmented into frames at the transmitter and reassembled at the receiver. In TPF, interleaving is not performed at the channel input. Thus, as in STP, we assume that FEC can only be used to correct for random errors. TPF corrects for burst errors via requesting for retransmission by the sender. TPF, however, can be made to have a superior throughput and delay performance compared with STP, due to a less severe performance penalty when a file is received in error. This happens only when the frame size is small so the probability of having an outage in each frame is also small. In this region, TPF is better, because when there is an outage, the fraction of bits in error for each frame is large and it makes more sense to ask for retransmission than code over the errors which has low code rate and thus an inefficient code. In TPF, only erroneous frames require retransmission while in STP, an entire file requires retransmission in the event of a burst error. In

[3] In practice, code lengths with implementable decoders are limited by hardware complexity and thus results in this thesis should only be used as bounds on performance though we believe practical systems can be designed to come close to these limits.

addition, in TPF, users have the option to reserve an additional time per session for the retransmission of erroneous frames and avoid large setup delays associated with new session requests.

In TPF, we found that the optimal frame length that maximizes throughput is small (< 1 Mbit). The same is true for the optimal frame length that minimizes delay. Segmenting a file into frames of the optimal frame length may result in a large segmentation and reassembly overhead for large files. For example, a 100 Gbit file would require the segmentation and reassembly of over ten-thousand frames. Thus, we want to design a protocol that can use larger frames. In our analysis, we found the practical frame length at which a user can segment a file in exchange for a small performance penalty ($\sim 10\%$). That frame size is of the order of 10^8 bits for typical OFS network parameters.

In TPFI, a frame is interleaved at the channel input as in STPI. In TPFI, however, we assumed that practical interleaving and code length limit the depth of the interleaver. Thus, ideal interleaving is not possible but after a duration of about 1-3 outage interarrival times, further interleaving will have little more gains. In TPFI, interleaving combined with FEC decreases the probability that a frame is received in error. We found that the throughput and delay performance of TPFI is better than that of TPF for large frame sizes. In addition, both the optimum frame length and practical frame length is larger in TPFI than in TPF. Thus, interleaving combined with FEC allows a file to be segmented into larger frames without a significant decrease in throughput or delay performance. We also do not want the frame size to be too large as in STPI since we want to limit the amount of channel resources used for retransmission. Typically the choice of parameters to optimize throughput and delays are different. For the protocol that optimizes delay, we find that the protocol favors reserving an additional duration per OFS session for frame retransmissions, minimizing the delay due to another setup time (including waiting time of the flow sequence).

In this work, we considered the possibility of interleaving combined with FEC to correct for burst errors. Interleaving and FEC increase both OFS throughput and delay performance and enable a file to be segmented into large frames (>100 Mbits). STPI has the best average delay performance in the region of interest but when retransmission is needed (not a typical case and occurs with low probability) the extra network resources needed equals to that of an entire transaction with a correspondingly large transmission delay. Thus, TPFI is the protocol of choice when higher order

statistics of the delay is important. Also when the file size is very large, interleaving and coding of the whole file will become unwieldy and require a large interleaver which is impractical.

Finally, in this thesis we have not considered the processing burden of the transport layer protocol. When the overhead processing and retransmission protocol execution is taken into account, the optimum strategy will favor large frames even more.

Appendix A

A Derivations for Equations in Chapter 5

A.1 Derivation of (5.21)

$\tau_s = 0$ and $\tau_p = 0$	
$E_n[T] = \frac{\frac{F}{D}\frac{L}{R}}{1-p}$ $= \frac{F}{R}\frac{1}{\eta}$	(A.1)

Let τ_s be the expected setup delay, τ_p be the roundtrip propagation delay, and p be the probability of an erroneous frame. Let F bet the length of the file in bits, L be the frame length, and D be the number of message bits per frame. Let $p_{k\backslash x}$ be the probability that k outstanding frames remain after session termination out of x initial frames sent. Let $E_x[T]$ be the total expected delay (including retransmissions) to receive x frames without error. We derive (5.21) by induction on x where

$$p_{k\backslash x} = \binom{x}{x-k}(1-p)^{x-k}p^k \tag{A.2}$$

$X = 1$:

$$p_{1\backslash 1} = \binom{1}{0}(1-p)^0 p^1 = p \tag{A.3}$$

$$E_1[T] = \frac{L}{R} + p_{1\backslash 1}E_1[T] = \frac{L}{R} \tag{A.4}$$

$X = 2$:

$$p_{1\backslash 2} = \binom{2}{1}(1-p)^1 p^1 = 2(1-p)p$$
$$p_{2\backslash 2} = \binom{2}{0}(1-p)^0 p^2 = p^2 \tag{A.5}$$

$$E_2[T] = 2\frac{L}{R} + p_{1\backslash 2}E_1[T] + p_{2\backslash 2}E_2[T]$$

$$E_2[T] = \frac{L}{R}\frac{2}{1-p} \tag{A.6}$$

$\boldsymbol{X = n-1}$:

$$E_{n-1}[T] = \frac{L}{R}\frac{n-1}{1-p} \tag{A.7}$$

$\boldsymbol{X = n}$:

$$\begin{aligned} E_n[T] &= n\frac{L}{R} + p_{1\backslash n}E_1[T] + \cdots + p_{n-1\backslash n}E_{n-1}[T] \\ &\quad + p_{n\backslash n}E_n[T] \\ &= n\frac{L}{R} + p_{1\backslash n}\frac{L}{R}\frac{1}{1-p} + \cdots + p_{n-1\backslash n}\frac{L}{R}\frac{(n-1)}{1-p} + p_{n\backslash n}E_n[T] \\ &= \frac{L}{R}\left(n + \frac{n - n(1-p) - np_{n\backslash n}}{1-p}\right) \\ &= \frac{L}{R}\frac{n}{1-p}(1-p_{n\backslash n}) \\ &= \frac{L}{R}\frac{n}{1-p}. \end{aligned} \tag{A.8}$$

A.2 Derivation of (5.22)

$\boldsymbol{\tau_s \neq 0}$ **and** $\boldsymbol{\tau_p \neq 0}$
$$p_{i\backslash n} = \binom{\lfloor n(1+\Delta)\rfloor}{n-i}(1-p)^{n-i}p^{\lfloor n(1+\Delta)\rfloor - n + i}$$
$$p_{n\backslash n} = \binom{\lfloor n(1+\Delta)\rfloor}{0}(1-p)^{0}p^{\lfloor n(1+\Delta)\rfloor} = p^{\lfloor n(1+\Delta)\rfloor}$$ (A.9)
$$E_n[T] = \frac{\lfloor n(1+\Delta)\rfloor\frac{L}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{i\backslash n}E_i[T]}{1-p_{n\backslash n}}$$

Let τ_s be the expected setup delay, τ_p be the roundtrip propagation delay, and p be the probability of an erroneous frame. Let F bet the length of the file in bits, L be the frame length, and D be the number of message bits per frame. Let $p_{k\backslash x}$be the probability that k outstanding frames remain after session termination out of x initial frames sent. Let $E_x[T]$ be the total expected delay (including retransmissions) to receive x frames without error. We derive (5.22) by induction on x where

$$p_{k\backslash x} = \binom{\lfloor x(1+\Delta)\rfloor}{x-k}(1-p)^{x-k}p^{\lfloor x(1+\Delta)\rfloor - x + k} \tag{A.10}$$

$\boldsymbol{X = 1}$:

$$p_{1\backslash 1} = \binom{\lfloor 1+\Delta\rfloor}{0}(1-p)^{0}p^{\lfloor 1+\Delta\rfloor} = p^{\lfloor 1+\Delta\rfloor} \tag{A.11}$$

$$\begin{aligned} E_1[T] &= \frac{L}{R}(\lfloor 1+\Delta\rfloor) + \tau_s + \tau_p + p_{1\backslash 1}E_1[T] \\ &= \frac{\frac{L}{R}(\lfloor 1+\Delta\rfloor) + \tau_s + \tau_p}{1 - p_{1\backslash 1}} \\ &= \frac{\frac{L}{R}(\lfloor 1+\Delta\rfloor) + \tau_s + \tau_p}{1 - p^{\lfloor 1+\Delta\rfloor}} \end{aligned} \tag{A.12}$$

$\boldsymbol{X = 2}$:

$$\begin{aligned} p_{1\backslash 2} &= \binom{\lfloor 2(1+\Delta)\rfloor}{1}(1-p)p^{\lfloor 2(1+\Delta)\rfloor - 1} \\ &= \lfloor 2(1+\Delta)\rfloor(1-p)p^{\lfloor 2(1+\Delta)\rfloor - 1} \\ p_{2\backslash 2} &= \binom{\lfloor 2(1+\Delta)\rfloor}{0}(1-p)^{0}p^{\lfloor 2(1+\Delta)\rfloor} = p^{\lfloor 2(1+\Delta)\rfloor} \end{aligned} \tag{A.13}$$

$$\begin{aligned} E_2[T] &= \lfloor 2(1+\Delta)\rfloor\frac{L}{R} + \tau_s + \tau_p + p_{1\backslash 2}E_1[T] + p_{2\backslash 2}E_2[T] \\ &= \frac{\lfloor 2(1+\Delta)\rfloor\frac{L}{R} + \tau_s + \tau_p + p_{1\backslash 2}E_1[T]}{1 - p_{2\backslash 2}} \\ &= \frac{\lfloor 2(1+\Delta)\rfloor\frac{L}{R} + \tau_s + \tau_p + \lfloor 2(1+\Delta)\rfloor(1-p)p^{\lfloor 2(1+\Delta)\rfloor - 1}\frac{\frac{L}{R}(\lfloor 1+\Delta\rfloor) + \tau_s + \tau_p}{1 - p^{\lfloor 1+\Delta\rfloor}}}{1 - p^{\lfloor 2(1+\Delta)\rfloor}} \end{aligned} \tag{A.14}$$

$\boldsymbol{X = n}$:

$$p_{k\backslash n} = \binom{\lfloor n(1+\Delta)\rfloor}{n-k}(1-p)^{n-k}p^{\lfloor n(1+\Delta)\rfloor - n + k}$$
$$p_{n\backslash n} = \binom{\lfloor n(1+\Delta)\rfloor}{0}(1-p)^{0}p^{\lfloor x(1+\Delta)\rfloor} = p^{\lfloor x(1+\Delta)\rfloor} \quad \text{(A.15)}$$

$$E_n[T] = \frac{\lfloor n(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} p_{i\backslash n} E_i[T]}{1 - p_{n\backslash n}}$$

$$= \frac{1}{1-p_{n\backslash n}}\Bigg(\lfloor n(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p + \sum_{i=1,\ n>1}^{n-1} \Bigg(\Big(\lfloor i(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p \Big) \frac{p_{i\backslash n}}{1-p_{i\backslash i}} \Bigg(1 + \sum_{j=1,\ i>1}^{i-1} \frac{p_{j\backslash i}}{1-p_{j\backslash j}} \Bigg(1 + \sum_{k=1,\ j>1}^{j-1} \frac{p_{k\backslash j}}{1-p_{k\backslash k}}(\dots) \Bigg) \Bigg) \Bigg) \Bigg) \quad \text{(A.16)}$$

$$= \frac{1}{1-p_{n\backslash n}}\Bigg(\lfloor n(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p + \sum_{i=1}^{n-1} \Big(\lfloor i(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p \Big) \frac{p_{i\backslash n}}{1-p_{i\backslash i}} + \sum_{i=2}^{n-1} \Big(\lfloor i(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p \Big) \frac{p_{i\backslash n}}{1-p_{i\backslash i}} \sum_{j=1}^{i-1} \frac{p_{j\backslash i}}{1-p_{j\backslash j}} + \sum_{i=3}^{n-1} \Big(\lfloor i(1+\Delta)\rfloor \frac{L}{R} + \tau_s + \tau_p \Big) \frac{p_{i\backslash n}}{1-p_{i\backslash i}} \sum_{j=2}^{i-1} \frac{p_{j\backslash i}}{1-p_{j\backslash j}} \sum_{k=1}^{j-1} \frac{p_{k\backslash j}}{1-p_{k\backslash k}} + \cdots \Bigg).$$

A.3 Derivation of E[X_1]

Let n be the total number of frames, p be the probability of an erroneous frame, θ be the probability of a failed initial transmission and Δ be maximum fraction of retransmission frames to total frames sent.

$$\begin{aligned} E[X_1] &= \sum_{k=0}^{n-1}(n-k)\binom{\lfloor n(1+\Delta)\rfloor}{k}(1-p)^k\, p^{\lfloor n(1+\Delta)\rfloor-k} \\ &\cong \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{n-1}(n-x)e^{-\frac{(x-\mu)^2}{2\sigma^2}}dx \\ &= n\vartheta_1 - \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{n-1}xe^{-\frac{(x-\mu)^2}{2\sigma^2}}dx \\ &= n\vartheta_1\big(1-(1+\Delta)(1-p)\big) + \sigma\sqrt{\frac{2}{\pi}}e^{-\frac{n-1}{2\sigma^2}} \end{aligned} \tag{A.17}$$

where

$$\begin{aligned} \mu &= n(1+\Delta)(1-p) \\ \sigma^2 &= np(1-p)(1+\Delta). \end{aligned} \tag{A.18}$$

Bibliography

[1] Katherine Xiaoyan Lin, "Green optical network design: power optimization of wide area and metropolitan area networks," Massachusetts Institute of Technology, Master's Thesis 2011.

[2] Vincent W.S. Chan; Guy Weichenberg; Muriel Medard, "Optical Flow Switching," in *Proceedings of the Workshop on Optical Burst Switching (WOBS)*, 2006.

[3] "Approaching the Zettabyte Era," Cisco, San Jose, CA, White Paper 2008.

[4] "Cisco Visual Networking Index: Usage," Cisco, San Jose, CA, White Paper 2010.

[5] Guy Weichenberg, "Design and Analysis of Optical Flow-Switched Networks," *IEEE/OSA Journal of Optical Communications and Networking* , vol. 1, no. 3, pp. B81-B97, August 2009.

[6] Guy E. Weichenberg, "Design and analysis of optical flow switched networks," Cambridge, MA, Ph.D. Thesis 2009.

[7] Norman Abramson, "The ALOHA System - Another alternative for computer communications," in *Fall Joint Computer Conference*, 1970, pp. 281-285.

[8] Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach*. Waltham, Massachusetts: Morgan Kaufmann, 2003.

[9] Etty J. Lee, "Free-Space Optical Networks - Fade and Interference Mitigation and Network Congestion Control," Cambridge, MA, Ph.D. Thesis 2010.

[10] Dina Katabi, "Decoupling Congestion Control and Bandwidth Allocation Policy With Application to High Bandwidth-Delay Product Networks," Cambridge, MA, Ph.D. Thesis 2003.

[11] Lei Zhang, "Fast scheduling for Optical Flow Switching," Cambridge, MA, S.M. Thesis 2010.

[12] Vincent W.S. Chan, "Optical Flow Switching Networks," *Proceedings of the IEEE*, 2012, Invited.

[13] Daniel C. Kilper, Sethumadhavan Chandrasekhar, Christopher A. White, and Bruno Lavigne, "Channel Power Transients in Erbium Doped Fiber Amplified Reconfigurable Transmission Systems," *Bell Labs Technical Journal*, vol. 14, no. 4, pp. 73-84, Feb 2010.

[14] Atul Srivastava John Zyskind, "Amplifier Issues for Physical Layer Network Control," in *Optically Amplified WDM Networks*. Burlington, United States of America: Academic Press, 2011, ch. 8, pp. 221-252.

[15] Joseph Junio, Yan Pan, Daniel C. Kilper, and Vincent W.S. Chan, "Channel Power Excursions from Single-Step Channel Provisioning," , Cambridge, MA, Holmdel, NJ.

[16] Y. Pan, D.C. Kilper, and V.W.S.Chan J. Junio, "Channel Power Excursions from Single-Step Channel Provisioning," *JOCN*, 2012.

[17] E.N. Gilbert, "Capacity of a Burst-Noise Channel".

[18] Rajiv Ramaswami, Kumar N. Sivarajan, and Galen H. Sasaki, *Optical Networks*. Burlington, MA: Elsevier Inc., 2010.

[19] W.W. Peterson, D.T. Brown, "Cyclic Codes for Error Detection," *Proceedings of the IRE*, pp. 228-

235, Januaray 1961.

[20] Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, United States of America: John Wiley & Sons, Inc., 2006.

[21] Robert G. Gallager, *Information Theory and Reliable Communication*. Cambridge, United States of America: John Wiley & Sons, Inc., 1968.

[22] Andrew J. Viterbi, Jim K. Omura, *Principles of Digital Communication and Coding*, J.W. Maisel Frank J Cerra, Ed. United States of America: McGraw-Hill, Inc., 1979.

[23] Mordechai Mushkin, Israel Bar-David, "Capacity and Coding for the Gilbert-Elliott Channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277-1290, Nov 1989.

[24] Lei Zhang, Delay and Throughput Analysis for M/G/1 Queue.

[25] Vincent Chan, 6.02 Lecture Notes, 2012.

[26] Harry L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York, United States of America: John Wiley & Sons, 1968.