1-1-2021

Analysis of Bundle Protocol with Turbo Code

Babita Pradhan
University of Mississippi

ANALYSIS OF BUNDLE PROTOCOL WITH TURBO CODE

A Thesis
presented in partial fulfillment of requirements
for the degree of Master of Science
in the Electrical Engineering
The University of Mississippi

by

Babita Pradhan

May, 2021

ABSTRACT

Bundle protocol is one of the main protocols for data transfer in delay/disruption networking characterized by long delay, frequent disruption, intermittent connectivity, and high error rates. Using a store and forward technique, bundle protocol stores the application data by dividing it into smaller bundles locally at each node and then forwards it to the next node when access is available. Usually, it operates as an overlay network by staying on top of any other networking architecture. As one of the main protocols for delay/disruption networking, many studies have analyzed its performance.

This thesis analyzes the bundle protocol's performance in terms of delivery time and net data transmission rate by incorporating it with a turbo code. For the lower-layer network, a user datagram protocol is used. Each bundle or fragment of a bundle will be used as an information for each turbo code frame which will be transmitted to the receiver. Among different information block lengths recommended to use for turbo code, we will only use the length of 8920 bits in order to provide reasonable throughput among all block lengths. We will be evaluating the performance in two different cases: (a) the channel bit-SNR is constant throughout the file transmission, and (b) channel bit-SNR varying in each transmission round. For varying SNR cases, one SNR, for each transmission round, will be chosen randomly from a SNR list.

Our main objective is to find the best bundle size that will maximize the net data transmission, conditioned on a given file failure probability target. This target is set in order to avoid infinite transmission of a file that may be time critical and may lose value, if delayed indefinitely. Also, spending infinite time for transmitting a single file can occupy the system for an extended time, delaying other messages that could be more important. Our analysis shows that bundle size, which is neither too small nor too large, works better for many cases.

All these studies are made with channel rates that are the same for uplink and downlink channels, i.e., symmetric channel rates.

ACKNOWLEDGEMENTS

I would like to express the most profound appreciation to my advisors, Prof Dr. Ramanarayan Viswanathan and Prof. Dr. Lei Cao, who have the attitude and the substance of genius. They continually and convincingly conveyed a spirit of adventure in regard to research and scholarship and excitement regarding teaching. Without their guidance and persistent help, this thesis would not have been possible. Special thanks to Dr. John Daigle for his advice and for serving on this thesis committee.

I also gratefully acknowledge the assistance of my friends and all other people who helped me.

University of Mississippi   Babita Pradhan

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Space communications and interplanetary networks have been the topics of study for a long time. Many works have been done for the development of reliable networking for such communications. Existing networking protocols, such as Transmission Controlled Protocol (TCP), User Datagram Protocol (UDP), are not suited for space communication as they operate in relatively small signal propagation latencies (in milliseconds), require relatively high data rates and continuous end-to-end connectivity, which is not possible in the case of space communications. Thus, a new networking technique named Delay/Disruption-Tolerant Network (DTN) was developed to mitigate these challenges. DTN [1, 2] is one of the reliable networking technologies developed for communication in the stressed communicating environment characterized by long propagation delay, disrupted links between sender and receiver and intermittent connectivity. As suggested by the name, DTN can handle long delays and unpredictable disconnected links inherent in deep-space communications. Although initially developed for interplanetary use, DTN may also be used on other applications encountering higher disruptions in connectivity. The potential earth applications are undersea communication, military and intelligence, public service and safety. A detailed description of DTN is provided in [3].

1.1 BUNDLE PROTOCOL

Bundle protocol (BP) [4] is the main protocol of DTN developed to withstand the challenging communicating environment during space communication. In BP, files are transferred by fragmenting it into smaller data units called "bundles," which can be further fragmented as per the requirement of lower networking layers. Mainly BP follows the store-and-forward message switching technique by overlaying on the top of lower-layer protocols, which can be any Internet protocol. With the store-and-forward mechanism, BP can either suspend or resume any ongoing data transfer whenever the connection is unavailable or available for data transmission, respectively. This property helps BP to withstand intermittent connectivity and make use of scheduled or opportunistic connectivity. BP also can perform custody-based retransmission. Unlike the store-and-forward mechanism, this service is optional. Whenever custody transfer is enabled, it guarantees reliable data transfer by forwarding custody acknowledgment (ACK) to the sending node. These properties of BP make it reliable to operate in space communication.

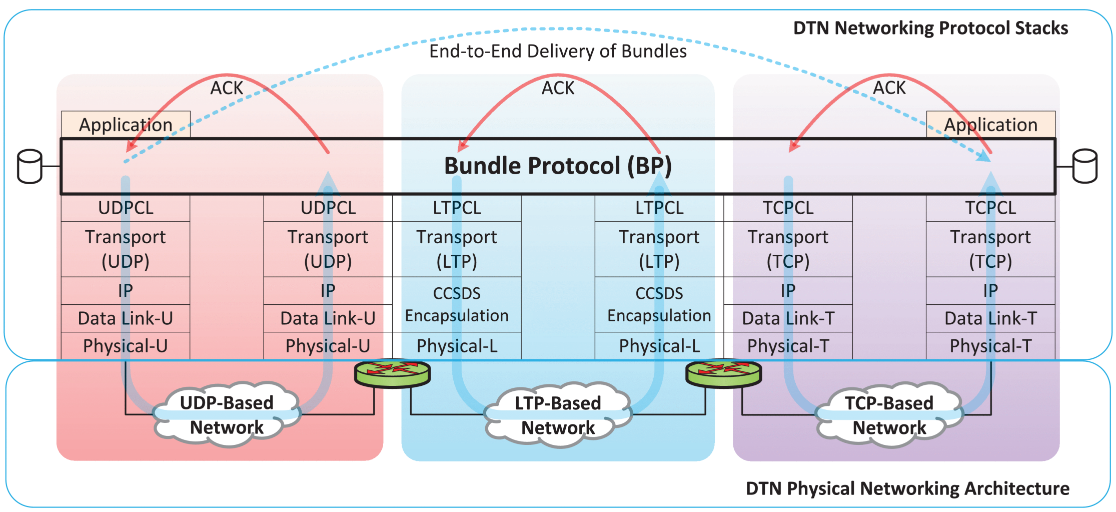


Figure 1.1. General DTN physical networking architecture and protocol stacks that implement BP with heterogeneous networks

Usually, BP stays on the application layer with some other lower layer Internet protocols for data transmission [4]. For a lower-layer network, protocols such as Transmission controlled protocol (TCP) [5], User Datagram Protocol (UDP) [6], or Licklider Transmission Protocol (LTP) is used. A general networking architecture and protocol stack for BP with heterogeneous networks is depicted in Figure 1.1[7]. As shown, BP can either operate across a heterogeneous network or any homogeneous network. The end-to-end file delivery, by residing in different networks, is possible due to the adapter's interfacing service. This adapter acts as a bridge between BP and lower layer protocol is called a "Convergence Layer Adapter" (CLA). Different CLAs can be used depending on the low-layer network protocol. Usually used CLAs are UDP based CLA (UDPCL), LTPCL used for LTP-based network and TCPCL used for TCP-based network, as shown in Figure 1.1. When UDP serves as CLA for bundles, called UDPCL [6], it transfers an unreliable networking protocol UDP to reliable data delivery architecture. Figure 1.2 shows the DTN protocol stack with UDP as the convergence layer. [4] provides the detailed architecture of BP.

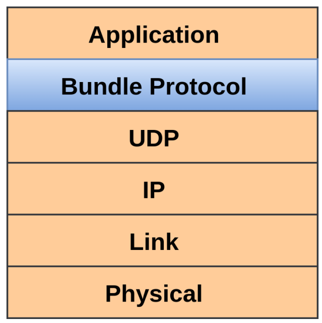


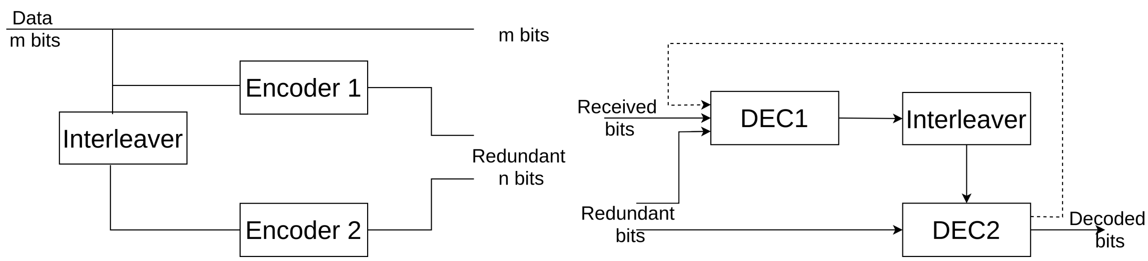
Figure 1.2. Different layers of networking showing bundle protocol over unreliable UDP transport layer

DTN protocol has been extensively investigated for space communications in recent years. Many works provided its performance analysis, including time delivery estimation, using symmetric or asymmetric channel conditions. Symmetric channel conditions mean using the same channel rates for uplink and downlink data channels. In contrast, asymmetric channels use different channel rates for uplink and downlink data transfer (usually higher data rate for downlink channels). In [8], the authors estimate bundle delivery time using a Contact Graph Routing (CGR) algorithm for space telecommunication. The paper predicts bundle routes and calculates reasonable arrival times, along with the corresponding probabilities. The authors in [9] present an analytical model for estimating the total delivery time of files using BP and study memory occupancy and release for asymmetric channel conditions. In this paper, BP's performance evaluation is focused on communication characterized by lengthy link disruptions, a too-long propagation delay, and lossy data links. It concludes that BP can deliver data over deep-space channels in the presence of lengthy disruption and delay, but these disruptions degrade performance. Similarly, [10] studies Round Time Trip (RTT) modeling for DTN protocol with LTP-based network over cislunar space channels. The authors validate their model by comparing it with experimental results that were obtained by conducting file transfer experiments on a PC-based Space Communication and Networking Test-bed (SCNT).This test-bed consists of PCs connected that act as source and destination for file transfer implementing the Interplanetary Overlay Network (ION) that was developed by the Jet Propulsion Laboratory. The paper concludes that a smaller data block size results in longer delays in ACK transmission and longer RTTs.

BP's performance over an highly asymmetrical channel is studied in [7] with the model to estimate expected file delivery time and goodput in the presence of lengthy link disruption. Similarly, in [11], analytical modeling is presented for estimating expected file delivery time over asymmetric channels. The analytical models proposed in these papers first evaluate the mean value of the number of transmission rounds taken to deliver the entire file and then calculate file delivery time. Both papers use the SCNT test-bed for validating their models. The performance evaluated for lengthy disruption concludes that it degrades performance and dominates other transmission latency. It also concludes that smaller bundle sizes are effective for file delivery rather than larger size due to increased delivery time for larger bundles. Other different studies on BP have been made in [12, 13].

1.2 TURBO CODE

Turbo codes [14] are error-correcting codes which have performance close to the Shannon theoretical limit. Turbo codes consist of a concatenation of convolutional codes and iterative decoding process. Generally, turbo encoders are formed by the parallel concatenation of two convolutional codes separated by an interleaver that forms encoding bits with a specific code rate. The main work of interleaver is to re-arrange the sequence of input symbols for encoding purposes. The simple encoder block diagram for turbo code is as shown in Figure 1.3a, which consists of two encoders connected in parallel with interleaver. If $m-$bits message signals are encoded using this encoder, it generates $(m+n)$ bits of encoded symbols with n redundant bits resulting in the code rate of $\frac{m}{(m+n)}$. These encoded bits contain an exact sequence of message bits ($m-$ bits), hence it is called a systematic codes.



(a) Turbo-encoder (b) Turbo-decoder

Figure 1.3. Block diagram for turbo encoder and decoder

The turbo decoder consists of two corresponding decoders that work in an iterative process to decode the received signal from the channel. The general block diagram for the turbo decoder is as shown in Figure 1.3b. In a turbo decoder, two decoders, DEC_1 and DEC_2 are connected in series with interleaver for decoding sequence from two encoders present in a turbo encoder. First, the received data is fed to DEC_1 with prior probabilities and it produces the extrinsic probabilities used by DEC_2 as prior probabilities to decode the signal. Then, the probabilities generated by DEC_2 is passed to DEC_1. This process of passing probability continues until the process is converged or until it reaches a maximum number of iteration.

Turbo codes are widely used in different areas from mobile communications to satellite communications, with code rates such as 1/6, 1/4, 1/3, 1/2, which may vary based on the application.

1.3 THESIS CONTRIBUTION

As mentioned above, many works have been done on performance analysis of BP under different channel conditions. All those studies are based only on analysis of BP not employing any error-correcting code. In this thesis, we are combining the error-correcting code, namely, turbo code with BP and analyze its performance. First, the file is divided into bundles and these bundles are encoded by a turbo encoder forming turbo frames. These turbo frames are then transmitted over the channel.

Unlike other studies that analyze bundles' infinite retransmissions, the thesis studies file transfer using BP and a finite number of retransmissions. Infinite transmissions are most useful when one transmits the critical data and there should not be any error in the received file. However, for live monitoring scenarios, monitoring day-to-day activities such as weather, temperature change and topology, the file transfer can be accepted with certain acceptable failure probability. Hence, transmission can be limited such that the file failure probability does not exceed an acceptable target value. Also, recent data could be more critical in these scenarios. Transmitting files with endless transmission rounds can cause recent data loss as old files occupy the available limited storage or delay recent data transfer. Considering this, we will be dealing with finite transmission during analysis in the thesis. A new analytical model is developed to get the average file delivery time for a finite number of retransmissions, which can then be extended to infinite retransmissions.

For this study, we consider two scenarios: (a) Equal SNR and (b) Unequal SNR. Equal SNR gives the performance of BP under idealistic condition of fixed SNR. Unequal SNR case depicts the real scenario of file transfer where channel condition could vary in each transmission due to change in SNR. For the unequal SNR scenario, we randomly choose SNR in each transmission round and provide an average performance under varying channel SNR conditions.

The main objective of the analysis is to find the best bundle size in different scenarios that will provide a minimum delivery time for file transmission and maximum data transmission rate under a given target file failure probability. It will be observed that under different SNR scenarios, bundle sizes that provide better performance are different and bundle sizes that are either too small or too large will not be the proper choices for file transmission.

1.4 THESIS ORGANIZATION

The rest of the thesis is organized as follows. Chapter 2 provides a description of file delivery using BP and describes two scenario assumptions that we consider for performance analysis. It also describes how BP and turbo codes are incorporated together to form frames. Moreover, system model for an analytical file delivery time is discussed. Chapter 3 discusses the formulation of time delivery, file failure probability, net data transmission, along with other relevant parameters. All equations required for the performance analysis of BP are provided in this chapter. The simulation overview is presented in Chapter 4, which is used to validate the analytical model for getting file delivery time. The results obtained from the analytical model and simulation are presented in Chapter 5 under different scenarios. The analysis of results yields appropriate bundle size for each scenario. Finally, we conclude this thesis in Chapter 6.

CHAPTER 2

MODEL

This chapter will present the system model of a BP which is then used for a performance analysis. First, we will present a brief overview of BP explaining the concept of file transfer using this protocol. Then, BP incorporated with the turbo code will be discussed along with the turbo code assignment for different SNRs. In later sections, the BP file transfer scenario for equal and unequal SNR and system model for a file delivery time are discussed.

For BP analysis, we will be using the User Datagram Protocol (UDP) as the lower layer networking protocol for which the network protocol stack was shown in Figure 1.2.

2.1 BUNDLE PROTOCOL FILE TRANSFER OVERVIEW

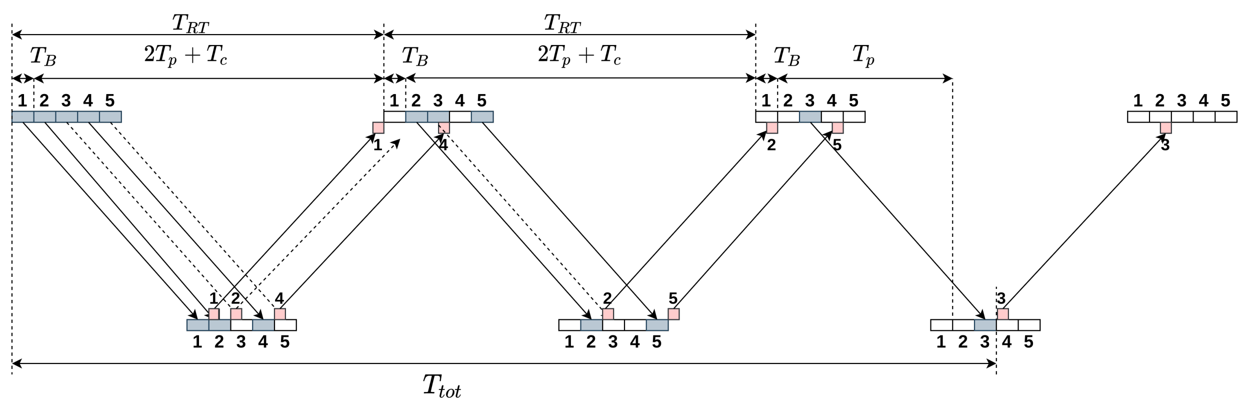


Figure 2.1. Bundle Protocol transmission scenario between sending and receiving nodes

Generally, in a BP, first, the file is divided into bundles and these bundles are transmitted. The file transmission is not considered successful until all bundles are received without any error. Two factors can fail bundle transmission: (1) bundle's corruption or loss due to channel error and (2) loss of its acknowledgment. Upon failure of a bundle, it is re-transmitted in the next transmission round. Figure 2.1 shows this protocol overview, where a file comprises of five bundles. In this figure, the dotted line represents a loss of bundle, whereas a solid line means the bundle's successful reception. Let T_B be the bundle transmission time, T_p be the one-way propagation time, T_c the acknowledgment(ACK) transmission time and T_{RT} the round trip time.

During the first transmission, all bundles are transmitted. Due to channel errors, the 3^{rd} and 5^{th} bundles are lost. For the rest of the bundles, the receiver sent ACKs stating the successful arrivals. However, ACK for the 2^{nd} bundle is lost. Now, the transmitter has to resend 2^{nd}, 3^{rd} and 5^{th} bundles in the second transmission as no ACK are received upon the expiration of corresponding Retransmission Time Out (RTO), which is usually $2T_p + T_c$. After the second transmission, again 3^{rd} bundle is lost, which is re-transmitted during the third transmission. Finally, this bundle is received successfully on the third transmission, marking the successful delivery of the entire file. The total time taken for the file to transmit successfully is denoted as T_{tot}.

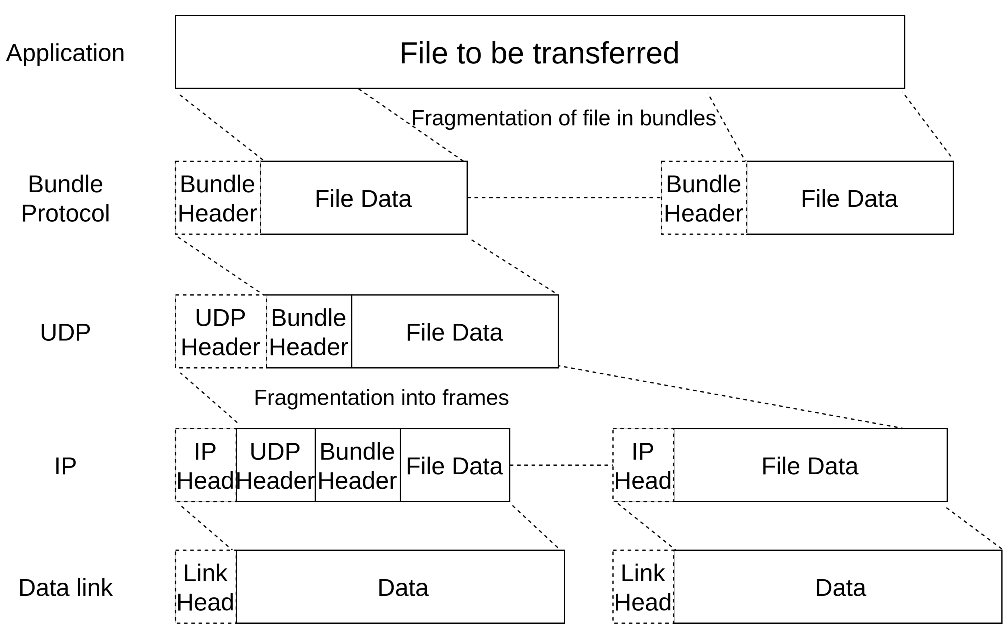


Figure 2.2. Bundle encapsulation in the different networking layer

Figure 2.2 shows the encapsulation of a bundle in all layers before transferring data to the channel. First, the application layer provides the file to the BP layer for transmission. Here, BP fragments the entire file into smaller data units known as bundles. The number of bundles required to represent an entire file depends on the bundle size used for transmission. Each bundle is provided with its bundle header. Then, the UDP layer encapsulates each bundle with its header. In the IP layer, fragmentation of bundles into smaller frames is carried. These frame numbers vary based on bundle size. The larger the bundle size, the higher will be frame numbers required. The fragmentation in the IP layer occurs only when the size of a bundle is greater than the transmission unit that can be handled. Otherwise, bundle transmission occurs without any fragmentation. This maximum frame or packet size that can be handled by the data link layer is the Maximum Transmission Unit (MTU). Upon appending the header by IP and data link layer, the information will be ready for transmission through the channel and is proceeded to its destination. The bundle we get after the encapsulation of overhead in each network layer is called an encapsulated bundle.

2.2 BUNDLE PROTOCOL WITH TURBO CODE

Bundle protocol can be incorporated with the turbo code in the link layer. One reason for using turbo code is that it helps in error correction that can minimize the retransmission. Also, the turbo code's use ensures the decoding of data with less error in the receiver within the limited transmission round.

Consultative Committee for Space Data Systems (CCSDS) has recommended the standard code rates and information block lengths for turbo codes. The recommended nominal code rates for turbo code are selected from {1/2, 1/3, 1/4, 1/6} and information block lengths are selected from {1784, 3568, 7136, 8920} bits giving the possibilities of 16 different turbo code frame arrangements. The turbo codes can be selected from these combinations to maximize throughput under the given channel-bit SNR condition. Out of four information block lengths {1784, 3568, 7136, 8920} and code rates {1/2, 1/3, 1/4, 1/6}, particular pair of block length and code rate provides the maximum throughput under the specific channel bit-SNR range [15]. The authors in [15] define the mechanism to get higher throughput by selecting an appropriate turbo encoder. The author selects the turbo encoder using the FER associated with corresponding information block length and code rate for given bit-SNR. This selection of information block length and code rate made for throughput maximization is shown in Table 2.1. Table 2.2 shows the corresponding values of FER for the corresponding SNR values. Appendix A shows the detailed information on correspondence between FER, bit-SNR, information block length, and code rate.

Table 2.1. Turbo code assignment (block length, code rate) for different channel bit-SNR range

Channel bit-SNR range	**Turbo code**
-0.5 dB to 0.15 dB	(8920,1/6)
0.2 dB to 0.35 dB	(8920,1/4)
0.4 dB to 1.0 dB	(8920,1/3)
1.1 dB to 2.2 dB	(8920,1/2)

Table 2.2. Frame Error Rate (FER) for different channel bit-SNR in dB

SNR(dB)	**FER**
-0.5	9.0909E-01
-0.4	4.7619E-01
-0.3	9.9701E-02
-0.2	6.6542E-03
-0.1	1.3089E-04
0.0	5.5200E-06
0.1	2.4000E-06
0.2	5.4510E-05
0.3	2.5700E-06
0.4	2.0755E-04
0.5	2.8730E-05

As the information block length of 8920 bits provides the maximum link throughput, we will be only using this length to analyze BP, but the code rate will vary based on the channel-bit SNR range. When used with turbo, bundle encapsulation is the same as that provided in Figure 2.2 for all network layers except encapsulation of the final data frame from the data link layer with turbo codes. Figure 2.3 depicts this encapsulation process. Each bundle or fragment of the bundle from the data link layer will serve as an information block for turbo code forming the turbo encoded frame. As the length of the information block is limited to 8920 bits, each frame from the layer just above turbo code should have a block length no more than this value. Hence, to fix this length, the frame size and bundle size in the upper layer are adjusted. For this to happen, we simply perform the backtracking. The selection of payload size in each upper layer for avoiding the zero paddings in the turbo frame is shown in Figure 2.4. Here, N_F denotes the number of frames formed during the fragmentation in IP layers. First, the bundle size of {1, 4, 8, 16, 32, 64} KB is selected and the number of frames required is calculated, which is ceiled to be integers. Using this exact frame number N, bundle size that avoids zero paddings is adjusted using $1081N_F - 8$. This backtracking gives us the bundle size of {1073, 4316, 8640, 16207, 32422, 63771} B, which we will use for analysis. Size less than 1 KB is not used as they are inappropriate for data transmission and greater than 64 KB is not used because the maximum payload UDP can handle is limited to this size.

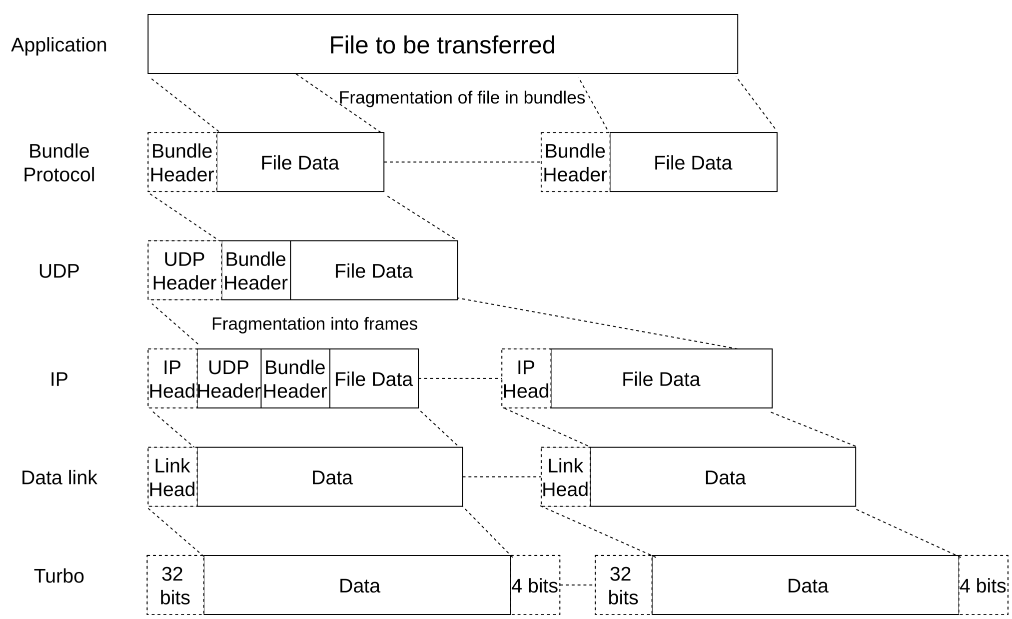


Figure 2.3. Encapsulation of Bundle in different networking layers with turbo code

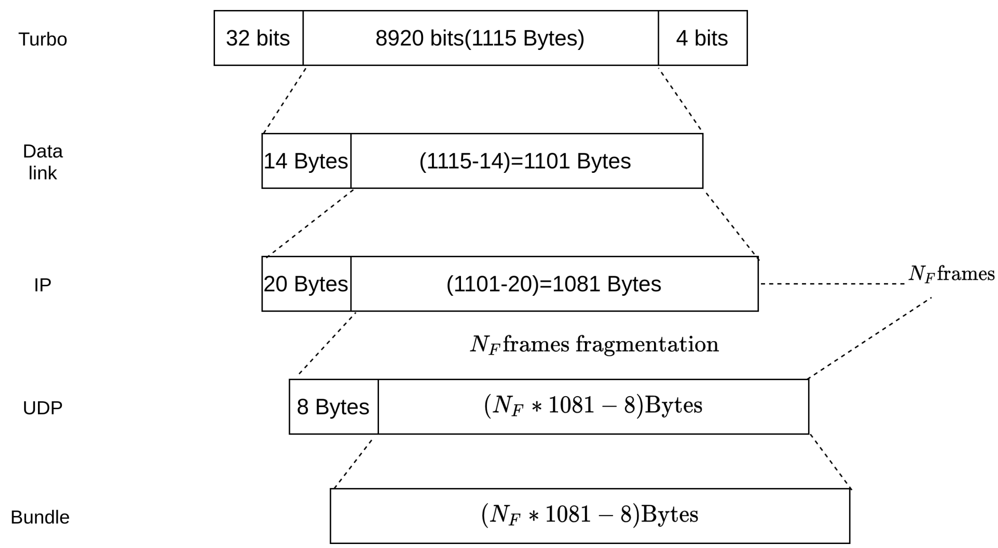


Figure 2.4. Determining Bundle size for Turbo code with information block length= 8920 bits

2.3 BUNDLE PROTOCOL OVERVIEW FOR EQUAL AND UNEQUAL SNR

Here, we will discuss the overview of file transfer using bundle protocol for an assumed bit-SNR value. Based on SNR used for the file transmission, one can select the corresponding Turbo code (information block length and code rate) and corresponding FER. We calculate the bundle transmission time and bundle failure probability from this information of turbo code. Chapter 3 discusses the detailed formulation for this calculation. The important point to note is the change in bundle transmission time with a change in SNR and the code rate during transmission.

Figure 2.5 shows the bundle transmission overview between the transmitter and receiver for a real scenario. As shown, during the real data transmission scenario, the SNR can vary during file transmission. The worst-case scenario can be changing SNR during each bundle transmission within a single transmission, as depicted in Figure 2.5. As a result of these varying bit-SNRs, bundle transmission time (T_B) for each bundle may also vary. For the first transmission, these different bundle transmission times are denoted by $T_{B_1}^{[1]}$, $T_{B_2}^{[1]}$ and others for corresponding bundles. Similarly, $T_{B_1}^{[2]}$ and $T_{B_1}^{[3]}$ denote the first bundle's transmission time for the second and third number of rounds, respectively, which may differ from other bundles depending on bit-SNR during that transmission. Although there is variation in T_B, the protocol followed for file transmission is the same as that explained in section 2.1. If we see Figure 2.5, we may notice some gaps between the two bundles after the first transmission. These gaps are the delay caused due to not expiring the RTO for the bundle, although the preceding bundle is already transmitted. RTO for any bundle depends on its transmission time, propagation delay and ACK transmission time.

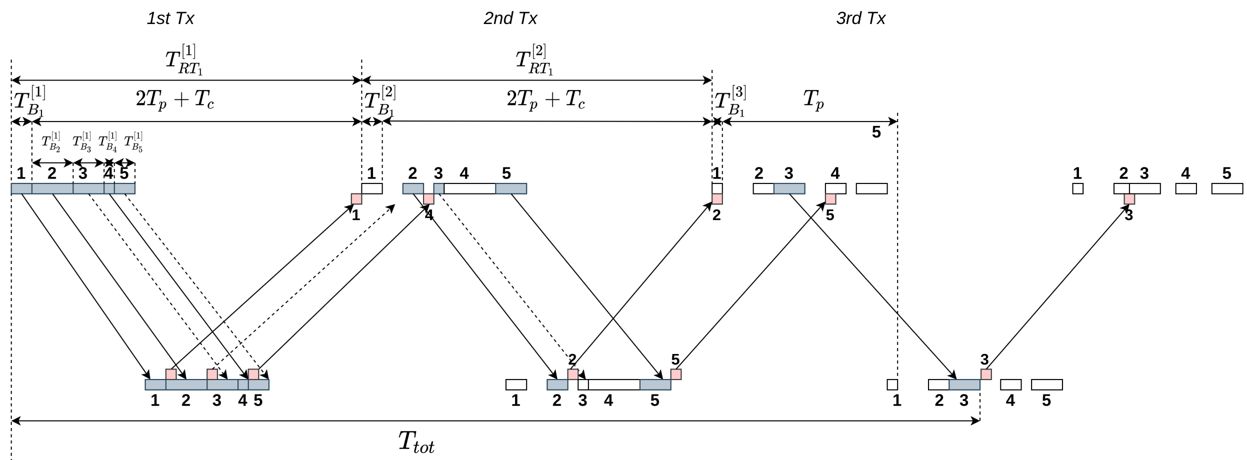


Figure 2.5. Bundle Protocol transmission scenario between sending and receiving node for a real extreme scenario where SNR can change at any time during transmission

The real scenario performance analysis seems complicated due to variation on bit-SNR during transmission at any time, which must be noted beforehand. Thus, for simplicity, two simple cases are studied here with the corresponding assumptions. These assumptions are as follows:

1. Assumption I: Equal SNR case

   In this case, we assume that all bundles transmit with a fixed SNR throughout the entire file's transmission and retransmissions. Figure 2.6 depicts this case. All bundles have equal T_B and equal SNR (represented as SNR_1) throughout the file transmission. Through this assumption of equal SNR, we study the effect of each SNR value on file transmission individually.

2. Assumption II: Unequal SNR/random case

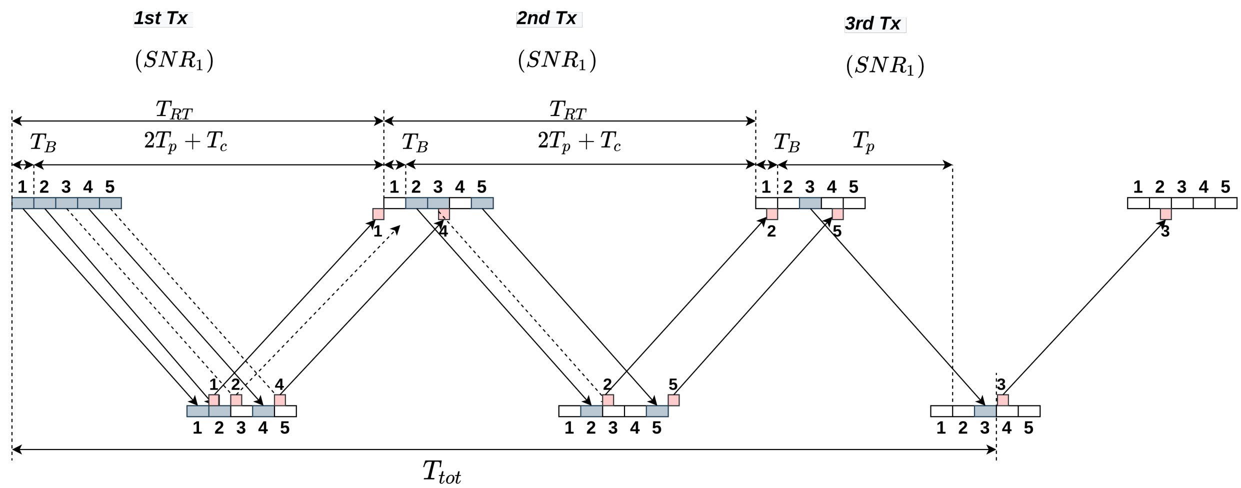


Figure 2.6. Bundle Protocol transmission scenario between sending and receiving node for constant SNR throughout a file transmission

In this case, we assume that SNR varies in each transmission round. That is, for individual transmission, there may be variation in SNR throughout the file transfer. Due to this variation, each round's bundle transmission time is changed but is the same within that transmission. Figure 2.7 depicts this scenario. During each transmission, SNR_1, SNR_2 and SNR_3 denote the variation of SNR for three respective rounds. These corresponding values are chosen randomly from SNR's given list, assuming any of them is equally probable. After choosing any random value, we transmit the bundle under that condition. In Figure 2.7, $T_B^{[1]}$, $T_B^{[2]}$ and $T_B^{[3]}$ represent bundle transmission time during each round, which may be different or the same depending on the transmission scenario. Also, there are some gaps between bundles on the 3^{rd} round. This gap is because although the first bundle transmits with $T_B^{[3]}$, which is lesser than $T_B^{[2]}$, the second bundle still has to wait for its RTO to expire, depending on the bundle transmission time of the previous round. Hence, if for k^{th} round, $T_B^{[k]} < T_B^{[k-1]}$, the gap is observed.

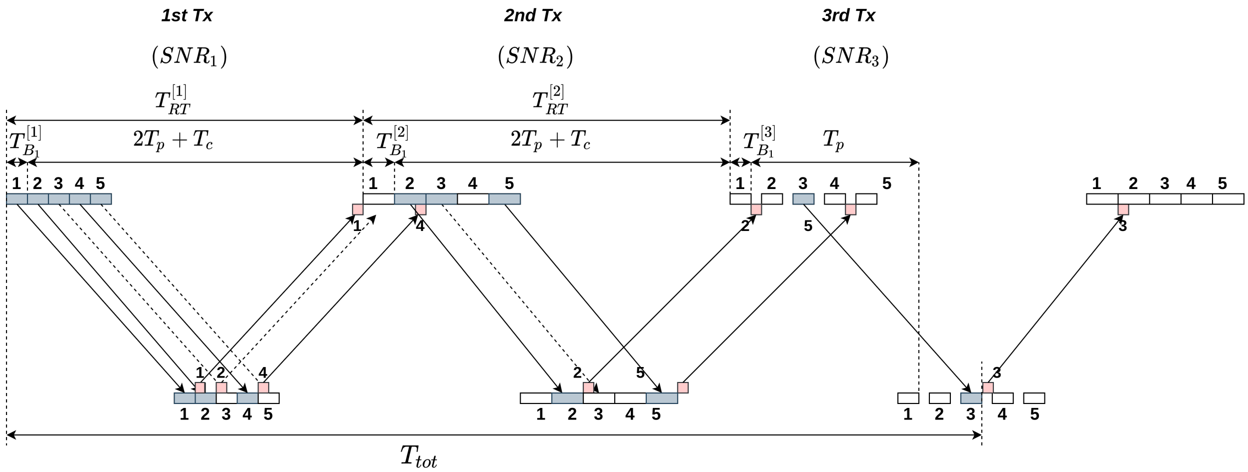


Figure 2.7. Bundle Protocol transmission scenario between sending and receiving node for randomly selected SNR in each transmission

2.4 DELIVERY TIME MODEL

In this section, the model used for computing file delivery time using bundle protocol is discussed. This model is appropriate for any number of transmissions, infinite or finite. We need this new model for delivery time computation as we will be dealing with the finite number of rounds also. To get delivery time for finite retransmissions, we will first find the last error bundle position in each transmission. This position is the point up to which one needs to transmit the bundle for the following round. The average time will be calculated based on the time to transmit bundles up to the last bundle error position for all rounds.

The file transmission scenario using bundle protocol between the sender and receiver is shown in Figure 2.8 for both symmetric and asymmetric channel conditions. The figure shows the transmission scenario for the first and second transmission rounds only. The process goes on similarly for other transmission rounds too. One can assign a limited number of retransmissions or retransmit infinitely until all bundles are transmitted successfully for getting the total delivery time for file transfer.

In Figure 2.8, for both channel conditions, it is assumed that the file has seven bundles, each bundle having the bundle transmission time of T_B and round trip time of T_{RT}. During the first transmission, all bundles are transmitted. Bundle 1, 3, 5 and 6 are corrupted due to

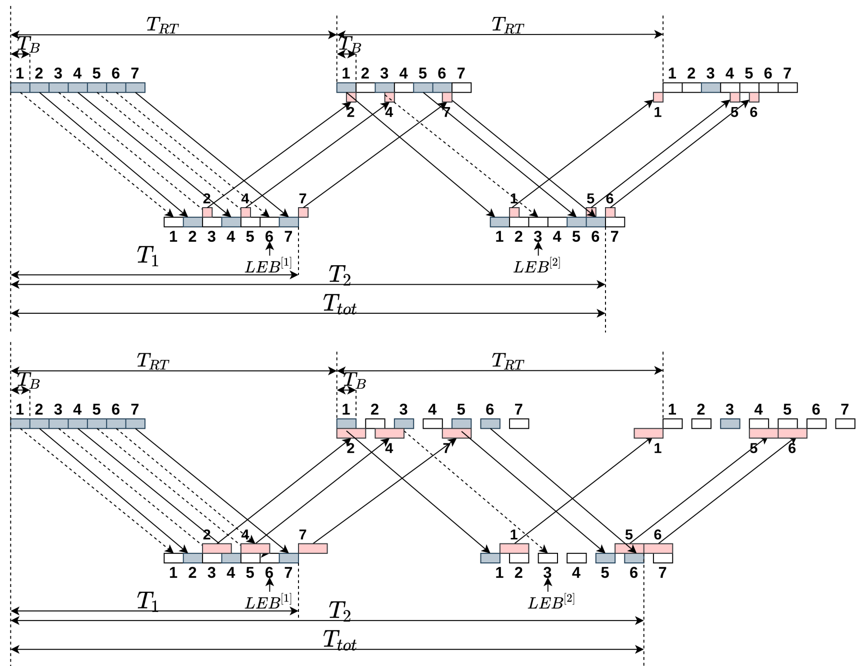


Figure 2.8. Bundle transmission scenario between sender and receiver for $N_T = 2$ for symmetric (shown at top) and asymmetric (shown at bottom) channel conditions

channel error. These IDs of corrupted bundles are also the position of an erroneous bundle for the first transmission. Also, the last error bundle after the first transmission (LEB$^{[1]}$) is the 6^{th} bundle. Acknowledgment for all successful bundles is issued by the receiver and sent to the sender, notifying bundle arrival. T_1 denotes the total time for bundle transmission during the first round. Only the corrupted bundles (1, 3, 5 and 6 bundle) will be transmitted in the second transmission round. After the second round, the last bundle error position, LEB$^{[2]}$, is 3^{rd} as it is corrupted by channel. The time for two transmission rounds (T_2) will be the total time for transmission of bundles until $LEB^{[1]}$. If single retransmission is allowed, then the total delivery time (T_{tot}) will be calculated based on T_1 and T_2.

For finite rounds, we compute the time taken to deliver bundles till that point only, but for infinite transmission, we transmit bundles infinitely until all bundles are transmitted successfully. The limited transmission round does not guarantee successful reception of all bundles at the receiver. However, we can limit the transmission number such that the file error probability is a minimum acceptable value.

If we observed asymmetrical channel case (figure 2.8 bottom), then from 2^{nd} transmission, the gap between bundles is noticed. These gaps are introduced as ACK transmission time is longer than bundle transmission time, due to which each bundle has to wait until its ACK arrives. More generally, we can say until round trip time (T_{RT}) is expired. These gaps' introduction depends on whether ACK transmission time is longer than bundle transmission time or not. For some bundle that is large enough, bundle transmission may take longer than ACK resulting in no gap.

The formulation for time delivery calculation will be discussed in section 3.1.

CHAPTER 3

FORMULATION

This chapter discusses the formulation of a file's total delivery time, file failure probability, number of transmissions round, and net data transmission rate under separate sections. Table 3.1 defines the notations used for this formulation. At first, we will discuss calculations of the bundle parameters like number of bundles (N_B), number of frames (N_F), bundle error probability (P_B), bundle transmission time (T_B) and other parameters that will be used for other formulations in different sections of the chapter.

If we assume that the file required for transmission has the size of L_{file} Bytes and each bundle has a size of L_B bytes including its header (L_{BH}), then the number of bundles that are required to transmit the entire file is given by:

$$N_B = \left\lceil \frac{L_{file}}{L_B - L_{BH}} \right\rceil \tag{3.1}$$

This calculated N_B depends on the bundle size used. The smaller the bundle size, L_B, the higher is the number of bundles, N_B, required to represent a file. As shown in Figure 2.3, there is encapsulation in each layer beyond BP and further fragmentation of bundle occurs in the IP layer, depending on the bundle size. Usually, fragmentation occurs when the bundle size exceeds the packet size that can be handled by the lower layer. For BP with turbo code, turbo frame size determines this frame/packet size for transmission. If L_{turbo}, L_{LH}, L_{IPH} and L_{UDPH} be the turbo frame, data link header, IP header and UDP header length, respectively, then the number of frames fragmented in the IP layer is given by:

$$N_F = \frac{L_B + L_{UDPH}}{L_{turbo} - L_{LH} - L_{IPH}} \tag{3.2}$$

Table 3.1. Notations

Symbols	**Definition**
L_{file}	Total length of the file
L_B	Bundle length/size
L_{BH}	Bundle header length
L_{EB}	Encapsulated bundle length
L_{UDPH}	UDP header length
L_{IPH}	IP header length
L_{LH}	Link layer header length
L_{turbo}	information block length for turbo code
L_c	Custody acknowledgment length
T_B	Bundle Transmission Time
T_p	Propagation delay
T_c	Acknowledgment transmission time
T_{tot}	Total delivery time
T_{RT}	Round trip time
N_B	Number of bundles
N_F	Number of frames
N_T	Transmission number
N_R	Number of rounds needed to achieve file failure
q	Probability of success of file delivery
P_B	Probability of bundle corruption
P_F	Frame error rate
P_{file}	Probability of file error
P_{fR}	Targeted file failure probability
r	code rate
R_D	Data channel rate
R_c	ACK channel rate
R_{net}	Net data transmission rate

As the size of turbo frames we are using is only 8920 bits, in order to get maximum throughput, Equation (3.2) can simply be written as $\frac{L_B+L_{UDPH}}{1081}$ after putting all corresponding values of turbo frames and header size for IP and data link layers. Figure 2.4 shows the exact value of these headers. Using equations (3.1) and (3.2), we find the corresponding values of L_B, N_B and N_F as provided in Table 3.2, where L_B is in Bytes and the file size is 20 MB.

Table 3.2. Bundle size (Bytes) with corresponding N_B and N_F for the file size of 20 MB

L_B	N_B	N_F
1073	20069	1
4316	4891	4
8640	2436	8
16207	1297	15
32422	648	30
63771	330	59

Bundle transmission time, T_B, is the time taken for a bundle to transmit from sender to receiver. This time depends on the total encapsulated length of a bundle (L_{EB}) and channel data rate (R_D) and is given by:

$$T_B = \frac{L_{EB}}{R_D} \tag{3.3}$$

For the BP with turbo code, each encapsulated bundle is equal to each turbo frame's length with its header multiplied by frame numbers. Substituting this value of encapsulated bundle in Equation (3.3) for any k^{th} transmission, the bundle transmission time is:

$$T_B^{[k]} = \frac{(L_{turbo} + 36)N_F}{r^{[k]} R_D} \tag{3.4}$$

Here, $r^{[k]}$ represents the k^{th} transmission's turbo code rate, determined by the SNR value for that transmission. When SNR varies, the rate may change and so does the bundle transmission time. For unequal SNR case, as described in chapter 2, we will get a different value of $T_B^{[k]}$ for any k^{th} round given by Equation (3.4). But for equal SNR case, this time will be the same throughout the file delivery time i.e. $T_B^{[1]} = T_B^{[2]} = ... = T_B^{[N_T]} = T_B$. Hence, for assumption I in section 2.3, $T_B^{[k]}$ can be reduced as given in Equation (3.5).

$$T_B^{[k]} = T_B = \frac{(L_{turbo} + 36)N_F}{rR_D} \tag{3.5}$$

Likewise, for the given ACK channel rate (R_c), the time for acknowledgment transmission is given by the following equation:

$$T_c = \frac{L_c + L_{UDPH} + L_{IPH} + L_{LH}}{R_c} \tag{3.6}$$

Any bundle is failed or corrupted if any frame failed to be received successfully. In the case of failure, those bundles need to be retransmitted during the next round. The error rate of any frame depends on the Frame Error Rate (FER) of the turbo frame, which can be known directly from Table 2.2 (for exact value, refer to appendix A) when knowing SNR. From this FER (P_F), we can calculate the bundle error probability (P_B) using Equation (3.7).

$$P_B = 1 - (1 - P_F)^{N_F} \tag{3.7}$$

This equation is for an equal SNR case. However, for unequal SNR cases (when SNR varies for different transmission round assumption II in section 2.3), FER varies in each round, causing a change in P_B. If $P_F^{[k]}$ is FER of turbo frames at the k^{th} transmission, then $P_B^{[k]}$ for that transmission is given by:

$$P_B^{[k]} = 1 - (1 - P_F^{[k]})^{N_F} \tag{3.8}$$

Equation (3.8) exactly equals to Equation (3.7) when $P_F^{[1]} = P_F^{[2]} = ... = P_F^{[N_T]} = P_F$,i.e., SNR is fixed for all transmission rounds.

Round-trip delay or Round Trip Time (RTT), denoted by T_{RT}, is simply a time taken by a packet to travel from source to destination and acknowledgment of that signal from destination to source. This time includes the propagation time for paths along with packet and acknowledgment transmission time. Propagation delay (T_p) is the time taken by the electromagnetic(wireless) signal to reach its destination. In our case, first, the bundle is transmitted from sender to receiver and then ACK is transmitted back by the receiver on getting the bundle successfully, suffering a total of two propagation delays. Hence, the RTT is given by:

$$T_{RT} = T_B + T_c + 2T_p \tag{3.9}$$

This time remains constant if T_B does not change but will vary if it changes, as in the case of unequal SNRs.

3.1 FILE DELIVERY TIME FOR FINITE NUMBER OF TRANSMISSION

This section discusses the formulation of delivery time for file transfer using the system model described in section 2.4. This formulation gives the delivery time of a file for any number of transmissions, finite or infinite.

3.1.1 EQUALLY LIKELY ERROR POSITIONS

Error positions refer to the bundles' positions that are not transmitted successfully during the previous transmission round. These positions are nothing but the ID or corresponding numbering of bundles from 1 to N_B. For example, if the 5^{th} bundle is corrupted, then our error position is simply 5 and it is required to be transmitted again in the next round. These error positions are equally likely over all bundles [1, 2,. . . , N_B]. If we assume m bundles failed after the 1st transmission, which is represented by set $Y^{[1]} = \{Y_i^{[1]}, i = 1, 2, \ldots, m\}$, then these error positions can be any bundle depending on the probability of P_B or $P_B^{[k]}$ (for any k^{th} transmission). The concept of equal probability for error positions also holds for other transmission rounds. We discuss both the mathematical and simulation perspective in order to show the validity of this result.

(I) Mathematics:

For any transmission, a bundle can be either a success or a fail, are represented by 0 or 1, respectively. Each bundle being successful or failed, is independent of each other. Therefore, these can be represented as independent Bernoulli trails. Let $Y_m^{[k]}$ denotes a Bernoulli random variable indicating if the bundle in the m^{th} position after the k^{th} transmission is erroneously received ($Y_m^{[k]} = 1$) or correctly received ($Y_m^{[k]} = 0$). If $Y_l^{[j]}$ is an error bundle at l^{th} position after j^{th} transmission, then for any given k, j, it is clear from the protocol and channel model that $Y_m^{[k]}$ and $Y_l^{[j]}$ are statistically independent for any $m \neq l, (m, l) \in [1, 2, \ldots N_B]$. Moreover, for any $m \in (1, 2, \ldots N_B)$ and any $k = 1, 2, \ldots,$ $(Y_m^{[k]} = 1) = \cap_{j=1}^{k}(Y_m^{[j]} = 1)$. Hence, the probability of m^{th} position error after the k^{th} transmission is given by:

$$P(Y_m^{[k]} = 1) = P_B^k \tag{3.10}$$

which is valid for all $m \in (1, 2, \ldots N_B)$. This means $Y_m^{[k]}$ are iid as Bernoulli random variables with probability P_B^k. Now, given that there are m_k bundle errors after the k^{th} transmission, because of iid Bernoulli variables, any subset of m_k positions of erroneous bundles out of N_B positions have equal probability, $\frac{1}{\binom{N_B}{m_k}}$.

(II) Simulation:

Here, we use a simulation study to show that error positions are equally likely over any transmission. For this, we transfer N_B bundles through the channel with an error probability of P_B during the first transmission and retransmit the erroneous bundles over the same channel for the next round. For each transmission, we note the positions of the error bundle. If we get frequencies of each erroneous bundle position for enough instances and obtain these counts of all positions as equal, we can conclude that error positions are equally likely.

Figure 3.1 shows the frequency of selecting any bundle position as an erroneous bundle for 100 bundles on different transmission number (N_T) with bundle error probability, $P_B = 0.345$ for 1000000 instances. The figure shows that each bundle position has nearly equal counts for being erroneous at any transmission round, conforming to bundle error positions' uniformity. The frequency of error position decreases with the increase in transmission round. This is because as we retransmit a file, the error occurrence lowers.

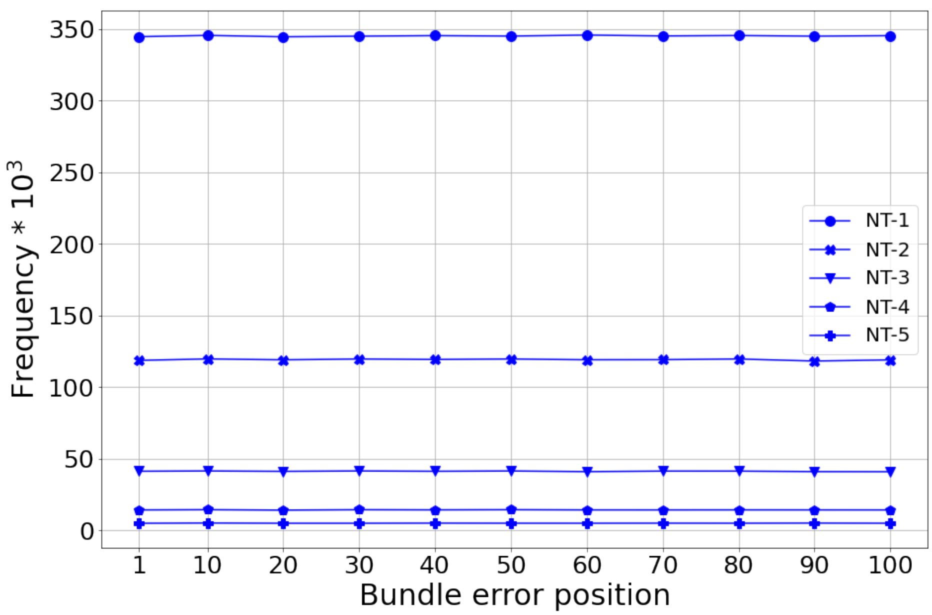


Figure 3.1. Bundle error position frequency plot for $N_B = 100$, $P_B = 0.345$ and $N_T = [1, 2, 3, 4, 5]$

Hence, we observed that the error positions equally range from 1 to N_B for any transmission round mathematically and by simulation.

We can also test if the assumption of equally likely error positions holds true or not by a simple chi-square test (goodness of fit test). For this, our null hypothesis is: error positions are equally likely and the alternative hypothesis is: error positions are not equally likely. As the error bundle can be in any position between 1 and N_B, we can treat each position as bins, giving us a total of N_B bins. For the chi-square goodness-of-fit computation, the Equation

(3.11) defines the test statistic.

$$\chi^2 = \sum_{k=1}^{N_B} \frac{(O_k - E_k)^2}{E_k} \tag{3.11}$$

Here, O_k is the observed value/sample value and E_k is the expected value.

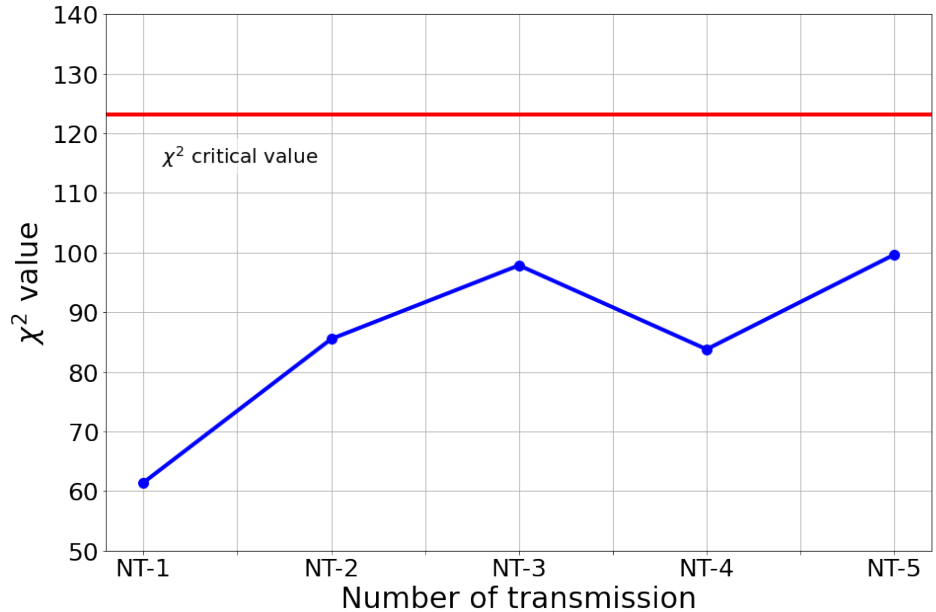


Figure 3.2. χ^2 value for $N_B = 100$ and $P_B = 0.345$ for different number of transmission

O_k is the frequency of error being in a certain position, which is just the count of total errors in the same bundle position during all instances obtained from simulation. The expected value after k_{th} transmission with N instances and error probability P_B is given by:

$$E_k = P_B^k N$$

Figure 3.2 shows that test statistics χ^2 is within critical limit $\chi^2_{cr} = 123.225$ for 1,000,000 instances. If χ^2 exceeds χ^2_{cr},then we reject the hypothesis that errors are equally likely. The observed, expected and χ^2 test statistics values for the configuration of $N_B = 100$ and $P_B = 0.345$ are as shown in appendix C. Hence, we are 95% confident that the error positions at any transmission are equally likely.

3.1.2 EXPECTED POSITION OF THE LAST ERROR BUNDLE

Before going to time delivery computation, we need to find the last error bundle (LEB) for each transmission. For the first round, this position is simply the last bundle transmitted, which is N_B. However, after the first transmission, error positions can be any between 1 and N_B depending on the number of error bundles. Let for the k^{th} transmission round, m_k be the number of error bundles with a set of error positions being $Y_i, i = 1, 2, ..., m_k$. If we assume $X_{[m_k]}$ is the last error bundle, then the expected position of this bundle X_k can be computed as shown in Equation (3.12). The numerator on line 3 of Equation (3.12), $\binom{x-1}{m_k-1}$, gives the number of combinations of the last error bundle to be on x^{th} position for m_k error bundles.

$$\begin{aligned}
X_k &= E[X_{[m_k]}] \\
&= \sum_{x=m_k}^{N_B} xP[X_{[m_k]} = x] \\
&= \sum_{x=m_k}^{N_B} x\frac{\binom{x-1}{m_k-1}}{\binom{N_B}{m_k}} \\
&= \frac{1}{\binom{N_B}{m_k}} \sum_{x=m_k}^{N_B} x\binom{x-1}{m_k-1} \\
&= \frac{1}{\binom{N_B}{m_k}} m_k \sum_{x=m_k}^{N_B} \binom{x}{m_k} \\
&= \frac{m_k}{\binom{N_B}{m_k}} \binom{N_B+1}{m_k+1} \\
&= \frac{m_k}{m_k+1}(N_B+1)
\end{aligned} \tag{3.12}$$

We know that the number of error bundles m_k can also vary from 1 to N_B for any transmission round, with the error probability at the k^{th} round being $P_k = P_B^k$ as given by Equation (3.10). Hence, again averaging the value X_k over m_k, we get,

$$E[X_k] = \sum_{m_k=1}^{N_B} \binom{N_B}{m_k} P_k^{m_k} (1 - P_k)^{N_B - m_k} X_k \tag{3.13}$$

This gives us the expected position of the last error bundle for any k^{th} transmission.

3.1.3 FILE DELIVERY TIME

Since we now have the expected last error bundle position for any k^{th} transmission, the mean file delivery time can be computed. As we have to retransmit only the error bundles in each transmission and Equation (3.13) provides the last error bundle position up to which the next transmission occurs, we can compute the time till that position. One point to pay attention to is that for the $(k+1)^{\text{th}}$ transmission, the preceding k^{th} transmission round gives the expected error position. This is because bundles that are unsuccessful in previous rounds are retransmitted on the next round. We compute the delivery time of the bundle at the receiver end. So, for the k^{th} round, this time is nothing but the addition of time to transmit error bundles, propagation time, and delays for the previous round. If $T_1, T_2, \ldots, T_k$ denote the delivery time for the first round, second round and others, then delivery time for each is given by:

For 1^{st} transmission,

$$T_1 = T_p + T_B N_B$$

For 2^{nd} transmission,

$$T_2 = T_p + T_B E[X_1] + T_{RT}$$

For 3^{rd} transmission,

$$T_3 = T_p + T_B E[X_2] + 2T_{RT}$$

Similarly, for any k^{th} transmission,

$$\begin{aligned} T_k = T_p + T_B E[X_{k-1}] + (k-1)T_{RT}&, \\ k = 2, 3, 4, \ldots& \end{aligned} \tag{3.14}$$

The above equation holds for symmetric channel rate. However, sometimes ACK takes a longer time to transmit than the bundle in asymmetric channel conditions. This creates the gap between bundles causing delays as discussed in section 2.4. In this case, we should note that gap too. However, for the larger bundle size, there is no gap as $T_B > T_c$. Hence, the transmission time is either governed by T_B or T_c, whichever is the larger. Therefore, Equation (3.14) can be written for the asymmetric channel rate as follows:

$$T_k = T_p + max(T_B, T_c)E[X_{k-1}] + (k-1)T_{RT}, \quad k = 2, 3, 4, \ldots \tag{3.15}$$

After getting the time for each specified transmission, we need to find the total delivery time, T_{tot}, the average of all respective time $T_1, T_2, \ldots, T_{N_T}$ up to any N_T rounds. This time is either time for 1^{st} transmission, T_1 with success probability of q_1 or time for 2^{nd} transmission, T_2 with the probability of $(1-q_1)q_2$, and so on. Hence, the mean delivery time for N_T transmission is given by:

$$T_{tot} = T_1q_1 + T_2(1-q_1)q_2 + \cdots + T_{N_T}(1-q_1)(1-q_2)\ldots(1-q_{N_{T-1}}) \tag{3.16}$$

This equation can be rewritten as:

$$T_{tot} = \sum_{k=1}^{N_{T-1}} T_kq_k \prod_{l=1}^{k-1}(1-q_l) + \prod_{l=1}^{N_{T-1}}(1-q_l)T_{N_T} \tag{3.17}$$

where q_k is the succeeding probability of transmitted bundles for the k^{th} transmission and is computed for each erroneous bundle during the previous transmission. Hence, if we have M_{k-1} average bundle errors during the $(k-1)^{\text{th}}$ transmission, then the succeeding probability of these error bundles transmitted during the k^{th} transmission is given by:

$$q_k = (1-P_B)^{M_{k-1}} \tag{3.18}$$

The average value of error bundles M_{k-1} for $(k-1)^{\text{th}}$ transmission is given by:

$$M_{k-1} = P_B^{k-1} N_B$$

One point to note for T_{tot} is for the last transmission, we do not care about the success or failure of bundles transmitted, so the succeeding probability for that round is not multiplied. However, if we want to get mean time for successful file delivery only, then it should be multiplied. Equation (3.17) can evaluate the mean delivery time for any transmission number from limited to unlimited by varying the value of N_T. If we limit the transmission round N_T, then Equation (3.17) provides the total time taken for file delivery within given transmission rounds only but does not guarantee the file delivery success at that period. There may still be some probability that file delivery fails. But, one can limit the number of transmission rounds such that the file failure probability is very low and acceptable. However, if $N_T = \infty$, then it assures that all bundles received are error-free.

3.2 FILE FAILURE PROBABILITY (P_{file})

This section talks about the calculation of file failure probability. File failure probability (P_{file}) is the probability that at least one bundle fails to be received successfully. This error results from finite transmission rounds (N_T) instead of transmitting infinitely until all bundles are successful.

Equation (3.8) gives the bundle failure probability for k^{th} transmission ($P_B^{[k]}$) for unequal SNR condition. For N_T transmission round, this failure probability is given by $\prod_{k=1}^{N_T} P_B^{[k]}$. If the file has N_B bundles, then P_{file} is nothing but error in at least one bundle and is given by:

$$P_{file}^{[N_T]} = 1 - (1 - \prod_{k=1}^{N_T} P_B^{[k]})^{N_B} \tag{3.19}$$

For equal SNR case, all transmission has equal P_B and hence Equation (3.19) can be simply reduced as follows:

$$P_{file}^{[N_T]} = 1 - (1 - P_B^{N_T})^{N_B} \tag{3.20}$$

The value of transmission round N_T determines the file failure probability. Therefore, it is chosen in order to obtain the minimum, acceptable probability of file failure. Hence, the transmission round can be fixed to a finite number for the provided file failure probability, eliminating the requirement for infinite rounds.

3.3 NUMBER OF TRANSMISSION ROUND (N_R)

Suppose we want to maintain a specific value of file failure probability during the transmission of the file. To achieve targeted file failure, we need to know the number of transmission rounds that guarantee that file failure does not exceed that value. This section gives us the transmission round formulation for achieving targeted or required file failure probability. Such a condition arises when we have only a fixed number of retransmission rounds and we need our data to be received accurately with a certain probability.

Let us assume that file failure probability must be P_{fR} and the number of transmission rounds that satisfy this file failure probability be N_R. If we assume equal SNR at all transmission rounds, then file failure given by Equation (3.20) should be upper bounded by P_{fR} as:

$$1 - (1 - P_B^{N_R})^{N_B} \leq P_{fR}$$

By further simplification, N_R is computed as:

$$N_R \geq \frac{\log[1 - (1 - P_{fR})^{\frac{1}{N_B}}]}{\log P_B}$$

$$N_R = \left\lceil \frac{\log[1 - (1 - P_{fR})^{\frac{1}{N_B}}]}{\log P_B} \right\rceil \tag{3.21}$$

In the case when SNR is varied at each transmission, P_{fR} can be bounded as :

$$1-(1-\prod_{k=1}^{N_T} P_B^{[k]})^{N_B} \leq P_{fR}$$

For each round, we can check if the above condition is satisfied or not. The minimum number of the round that satisfies this condition will be our required transmission round, N_R to achieve the targeted file failure probability. Equation (3.22) gives this value of N_R.

$$N_R = \min\left\{ N_T \in \mathbb{N} \quad | \quad 1-(1-\prod_{k=1}^{N_T} P_B^{[k]})^{N_B} \leq P_{fR} \right\} \tag{3.22}$$

As there is randomness associated with unequal SNR, we can compute N_R's average value after getting it for each instance. This computed round N_R guarantees that the file failure would not exceed the required failure probability.

3.4 NET DATA TRANSMISSION RATE (R_{net})

Net data transmission rate simply means the speed of data flow from the sender to the receiver or one node to another. It is also known as the throughput of the system. The higher the data rate, the better is the system as it provides us high information within the given time. These rates are usually measured in bits per second (bps) or bytes per second (Bps).

Here, we will calculate the data transmission rate for transferring the file using different bundle sizes under BP. The bundle size that provides a higher data rate under the given scenario is best suited for file transfer as they provide higher information bits in one second. Generally, the data transmission rate is calculated by dividing the total data transferred by the time taken to transfer it. In our case, file size is the data we are transmitting. If we consider an infinite transmission round, all bits are transmitted successfully, but as we allow only the fixed number of retransmissions, the whole file may not be transmitted successfully. Hence, we will only consider the successful bits for the calculation of the net data transmission rate. Equation (3.23) provides the required net data transmission rate.

$$R_{net} = \frac{L_{file}(1 - P_{file})}{T_{tot}} \tag{3.23}$$

CHAPTER 4

SIMULATION

4.1 FLOWCHART

Figure 4.1 depicts the flowchart of the simulation process. The simulation process for a single instance is as explained below:

1. At first, we create a queue with all bundles that are required to be transmitted. Each bundle is provided with three attributes: bundle id, which is simply the bundle position, status and time. Each bundle's status attribute can be True or False, which means transmission success or failure, respectively. Time attribute provides the total period a bundle takes in order to reach a receiver.

2. At the beginning of transmission, the number of rounds required for file transmission is assigned to be 1 and other attributes for each bundle are assigned as follows:

   id = position

   status = False

   Time = id $*T_B$ $+T_p$

3. Two conditions can terminate iteration: (1)If the queue is empty, i.e., all bundle has been transmitted successfully and (2)If the number of rounds exceeds the provided maximum number of transmission.

4. First, we check the queue to see if it is empty or not. If it is not empty, then some bundles are required to be transmitted for the next round. Upon completion of each queue cycle, the number of rounds increases.

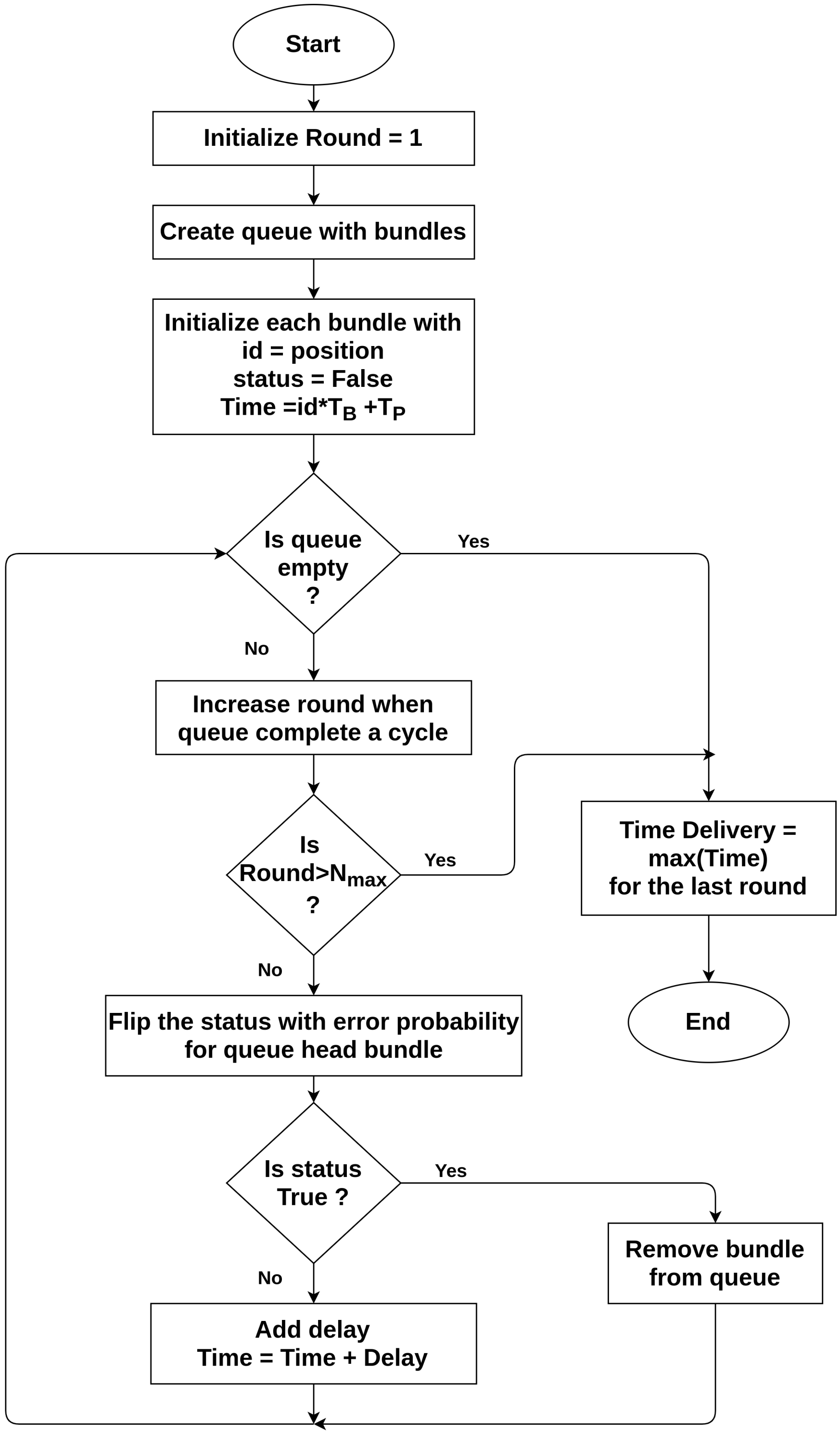


Figure 4.1. Simulation Flowchart

5. Next, we check the number of rounds to see if it has exceeded the maximum number of transmissions allowed. If an infinite number of transmission is allowed, then the process terminates only when the queue is empty. Otherwise, it can terminate before the queue got emptied.

6. If both conditions are not satisfied, then the queue bundles are passed through the channel model. Here, the channel flips the status of the bundle with the error probability of P_B. Furthermore, on checking the status, if it equals False, then the delay is added to that bundle's corresponding time. On the contrary, the bundle is removed from the queue as the retransmission is not required.

7. After the iteration terminates, we calculate the total delivery time. This time is given by the maximum value of the remaining bundles' attribute time in the last round.

4.2 PLATFORM

The simulation was performed on a Core i7 CPU with 16 GB memory using python as a programming language.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 PARAMETER SETUP

For analysis of file transfer using BP, the UDP was used as a lower layer and turbo code was implemented on the link layer. The protocol stack used is as provided in Figure 1.2. The file size was 20 MB, transferred by using six different bundle sizes. These bundle sizes (L_B), along with their corresponding number of bundles (N_B) and the number of frames (N_F), are provided in Table 3.2. Here, N_F represents the frame partitioning of each bundle in the IP layer. The turbo frame of length used for transmission is (8920+36) bits.

We configured the channel condition to be symmetric with an equal uplink and downlink channel rate of 2 Mbps. The propagation delay was set to 10 minutes, which is common for the cis-Martian channel. Each turbo frame has its FER, which varies based on channel bit-SNR used for file transfer. The evaluation is performed with only 7 different bit-SNR which are -0.1, 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5 dB respectively. Appendix A and Table 2.1 show their respective FER values and the code rate. For random (unequal) SNR, one is chosen randomly from these SNR for each transmission, assuming all are equally likely. The results from the random case are for an average of 1000000 instances.

For validation of the analytical model for a file delivery time, simulation results are used, which are obtained by averaging 1000000 instances for each configuration. Appendix B provides the values of bundle error probabilities (P_B) and bundle transmission time (T_B) for each bundle size for all bit-SNR cases that were directly used as simulation parameters. Table 5.1 shows the summarized parameters setup.

Table 5.1. Parameter used for simulation and theoretical analysis

Parameters	**Setup values**
File size (L_{file})	20 MB
Bundle Size (L_B)	1073,4316,8640,16207,32422,63771 B
Bundle header length (L_{BH})	28 B
Header lengths:$L_{UDPH}, L_{IPH}, L_{LH}$	8, 20, 14 B
Turbo frame length (L_{turbo})	8920 bits
Data channel rate(R_{Data})	2 Mbps
Channel bit-SNR	-0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5 dB
Propagation delay (T_p)	10 minutes
Simulation instances	1,000,000

5.2 FILE FAILURE

This section discusses the change of file failure probability (P_{file}) with the changes in bundle sizes, number of transmission and bit-SNRs. For evaluation of P_{file}, Equations (3.20) and (3.19) are used for fixed and random (unequal) SNR cases, respectively.

Figure 5.1 presents the P_{file} curve for transmitting the 20 MB file using BP with turbo for different bundle sizes with different SNR values evaluated at $N_T = 3$. In the figure, SNR random means unequal SNR case discussed in Assumption II in section 2.3. Tables 5.2 and 5.3 give the exact P_{file} values obtained under these scenarios. Figure 5.1 shows only the P_{file} curves for selected SNR. However, other SNRs also show similar nature of file failure.

Table 5.2. File failure probability obtained for different SNR at $N_T = 3$

L_B	**-0.1**	**0.0**	**0.1**	**0.2**	**0.3**
1073	4.500330e-08	4.456213e-12	0.000000e+00	3.250808e-09	0.000000e+00
4316	7.015215e-07	5.267198e-11	4.344081e-12	5.068727e-08	5.430101e-12
8640	2.792989e-06	2.098695e-10	1.730882e-11	2.018957e-07	2.109513e-11
16207	9.788980e-06	7.361072e-10	6.047829e-11	7.081822e-07	7.430190e-11
32422	3.901015e-05	2.942087e-09	2.418705e-10	2.827073e-06	2.969784e-10
63771	1.502499e-04	1.139407e-08	9.367440e-10	1.092539e-05	1.150193e-09

The curve shows that failure probability increases with an increase in bundle sizes. As bundle size increases, bundle error probability will also increase, resulting in bundle failure growth that results in higher file failure probability. The file failure is 0 in the simulation, for the 1073 B bundle at SNR = 0.3, which means that the file transmitted successfully at

Table 5.3. File failure probability obtained for different SNR at $N_T = 3$ contd.

L_B	0.4	0.5	random
1073	1.794294e-07	4.768148e-10	4.343934e-09
4316	2.796016e-06	7.421862e-09	6.771130e-08
8640	1.112671e-05	2.956806e-08	2.695552e-07
16207	3.896556e-05	1.037430e-07	9.445892e-07
32422	1.550080e-04	4.143838e-07	3.762926e-06
63771	5.949421e-04	1.603213e-06	1.448287e-05

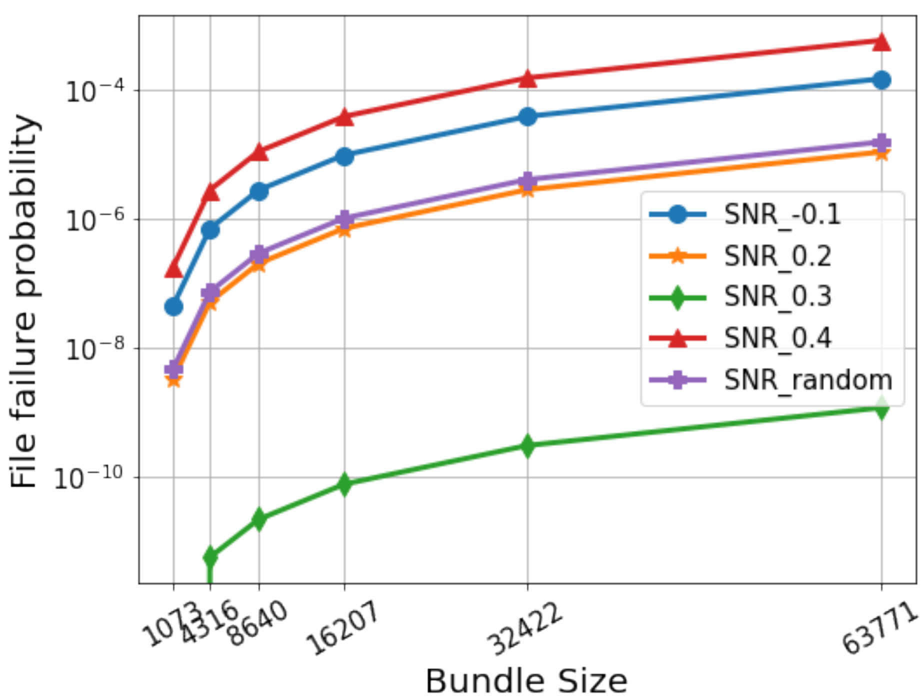


Figure 5.1. File failure obtained for different Bundle size with different SNR for $N_T = 3$

$N_T = 3$ without any error. If we observe the particular bundle size as 32422 B bundle, it has the highest failure at 0.4 dB and the lowest at 0.3 dB. This variation on file failure is due to FER's variation, resulting in the different bundle error probabilities. The curve suggests that 0.3 dB has the lowest error rate among all. However, for the overall SNR range we have evaluated, 0.1 dB provides the lowest P_{file} at $N_T = 3$.

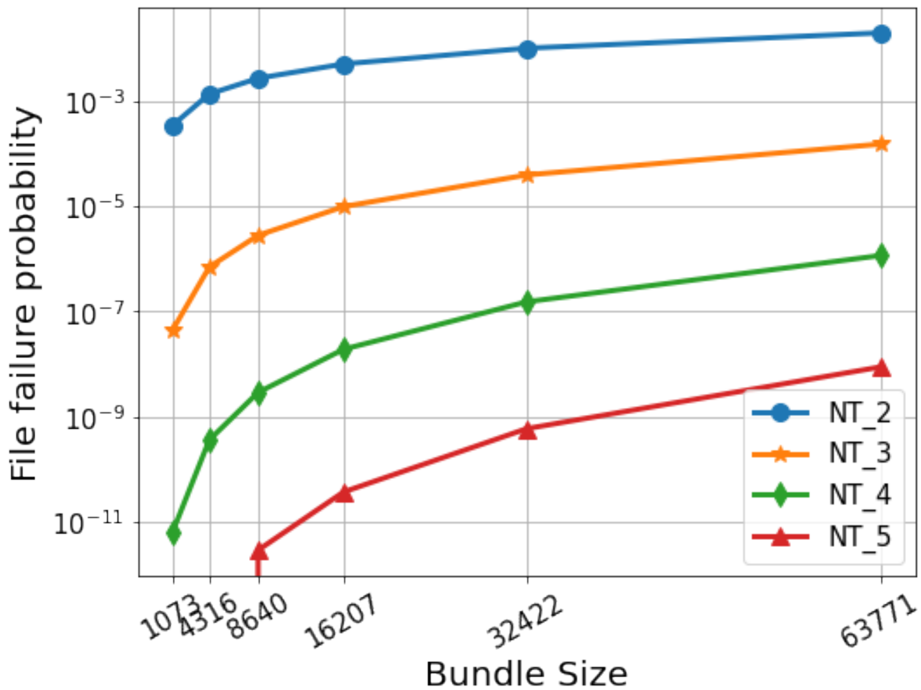


Figure 5.2. File failure obtained for different Bundle size with different N_T and SNR = -0.1

Figure 5.2 shows the effect of the number of retransmission rounds (N_T) in file failure for a constant channel bit-SNR. The curve shows that the file failure increases with the bundle size for any particular N_T due to an increase in bundle error probability with increasing size. Also, we observed a decrease in file failure with the rise of N_T. This decrement is obvious because as the number of transmission increases, the bundle failed in the previous transmission may be successfully received in the next round, decreasing file failure probability.

5.3 ACHIEVING SPECIFIED FAILURE PROBABILITY P_{fR}

Here, we are analyzing the performance of BP for a given failure probability (P_{fR}). Our main objective is to obtain the best bundle size that maximizes the net data transmission rate. We evaluate this with $P_{fR} = 10^{-3}$, which means we need to maintain file failure less than that. First, we will get the number of rounds required to achieve specified file failure probability and then compute delivery time for these obtained rounds. Lastly, we will analyze the data transmission rate under this condition of failure probability.

5.3.1 N_R REQUIRED TO ACHIEVE P_{fR}

This section talks about the results for the number of transmission rounds (N_R) that is required to achieve the specified file failure (P_{fR}). Equations (3.21) and (3.22), explained in Chapter 3 for two different assumptions of SNR scenario (constant and random), are used to evaluate N_R. For the random variation of SNR in different transmission (Assumption II), one SNR is chosen randomly from the given range and the average analysis is performed for 1000000 different instances.

Table 5.4. N_R required to achieve $P_{fR} = 10^{-3}$ for different SNR

L_B	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	random
1073	2	2	2	2	2	2	2	2
4316	3	2	2	2	2	3	2	3
8640	3	2	2	2	2	3	2	3
16207	3	2	2	2	2	3	2	3
32422	3	2	2	3	2	3	2	3
63771	3	2	2	3	2	3	2	3

Figure 5.3 shows the plot for bundle size vs. N_R to achieve $P_{fR} = 10^{-3}$ for different SNR. Table 5.4 shows the corresponding values of N_R required to achieve the specified file failure ($P_{fR} = 10^{-3}$) for all SNR and bundle sizes. Only two or three transmission rounds are enough to maintain required file failure under all SNR cases for all bundle sizes. These values for N_R are the minimum value of the transmission rounds that satisfy file failure requirements. The plot shows that the number of transmission round usually increase with

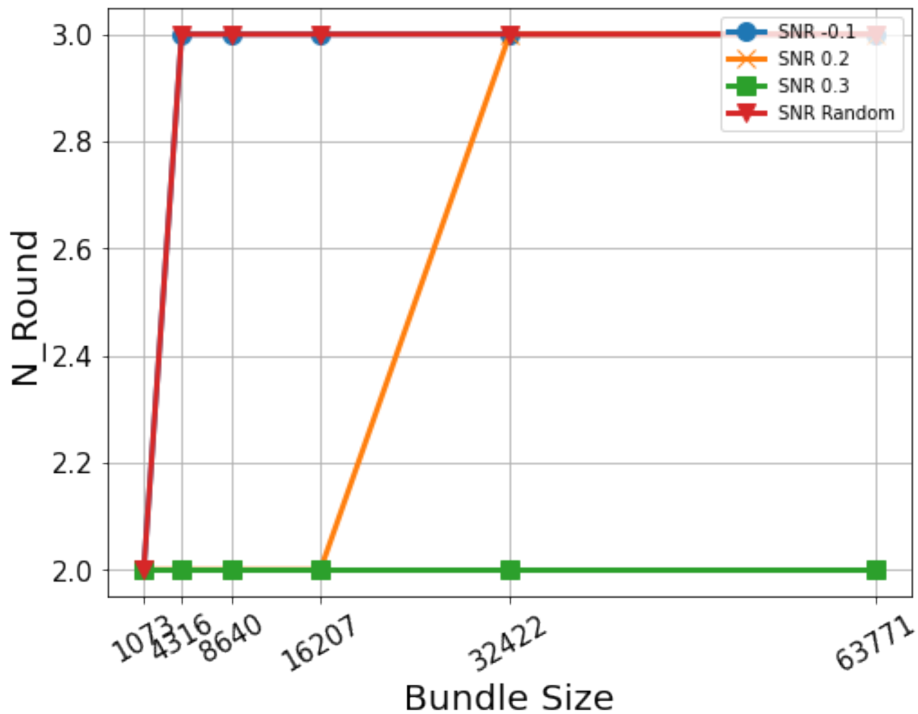


Figure 5.3. Number of rounds (N_R) required to obtain $P_{fR} = 10^{-3}$ for different bundle size with different SNR

bundle size. The relation between bundle error and bundle size is directly associated with this increment. With an increase in bundle size, the bundle error probability increases, which results in the requirement of retransmission to achieve the specified file failure. If we see a scenario $SNR = 0.3$ dB shown in Figure 5.3, then the equal N_R is required for all bundle sizes, representing no variation in number of rounds with a bundle size. If we see constant bundle size, then the change in the number of rounds is governed by bundle error probability, which varies based on FER. The higher the FER, the higher will be the number of rounds required. For the lowest bundle size (as 1073 B), only two transmission rounds are enough for all SNR cases to achieve a specified file failure.

These numbers of rounds obtained above are ceiled to be an integer. Hence, the actual file failure obtained may be lesser than the specified P_{fR}. As our targeted P_{fR} is just the upper bound beyond which file failure can not exceed, these lesser file failures are acceptable. The actual file failure achieved for N_R computed is listed in Tables 5.5 and 5.6.

As all failure values are upper bounded by targeted P_{fR}, we can say that computed N_R can achieve our targeted file failure probability.

Table 5.5. Actual P_{file} achieved for N_R given in table 5.4 for different SNR

L_B	-0.1	0.0	0.1	0.2	0.3
1073	0.000344	6.12e-07	1.16e-07	5.96e-05	1.33e-07
4316	7.02e-07	2.38e-06	4.51e-07	0.000232	5.17e-07
8640	2.79e-06	4.75e-06	8.98e-07	0.000463	1.03e-06
16207	9.79e-06	8.89e-06	1.68e-06	0.000866	1.93e-06
32422	3.9e-05	1.78e-05	3.36e-06	2.83e-06	3.85e-06
63771	0.00015	3.5e-05	6.62e-06	1.09e-05	7.59e-06

Table 5.6. Actual P_{file} achieved for N_R given in table 5.4 for different SNR contd.

L_B	0.4	0.5	random
1073	0.000864	1.66e-05	7.66e-05
4316	2.8e-06	6.46e-05	0.000115
8640	1.11e-05	0.000129	0.000112
16207	3.9e-05	0.000241	9.55e-05
32422	0.000155	0.000481	0.000157
63771	0.000595	0.000946	0.00019

5.3.2 DELIVERY TIME TO ACHIEVE P_{fR}

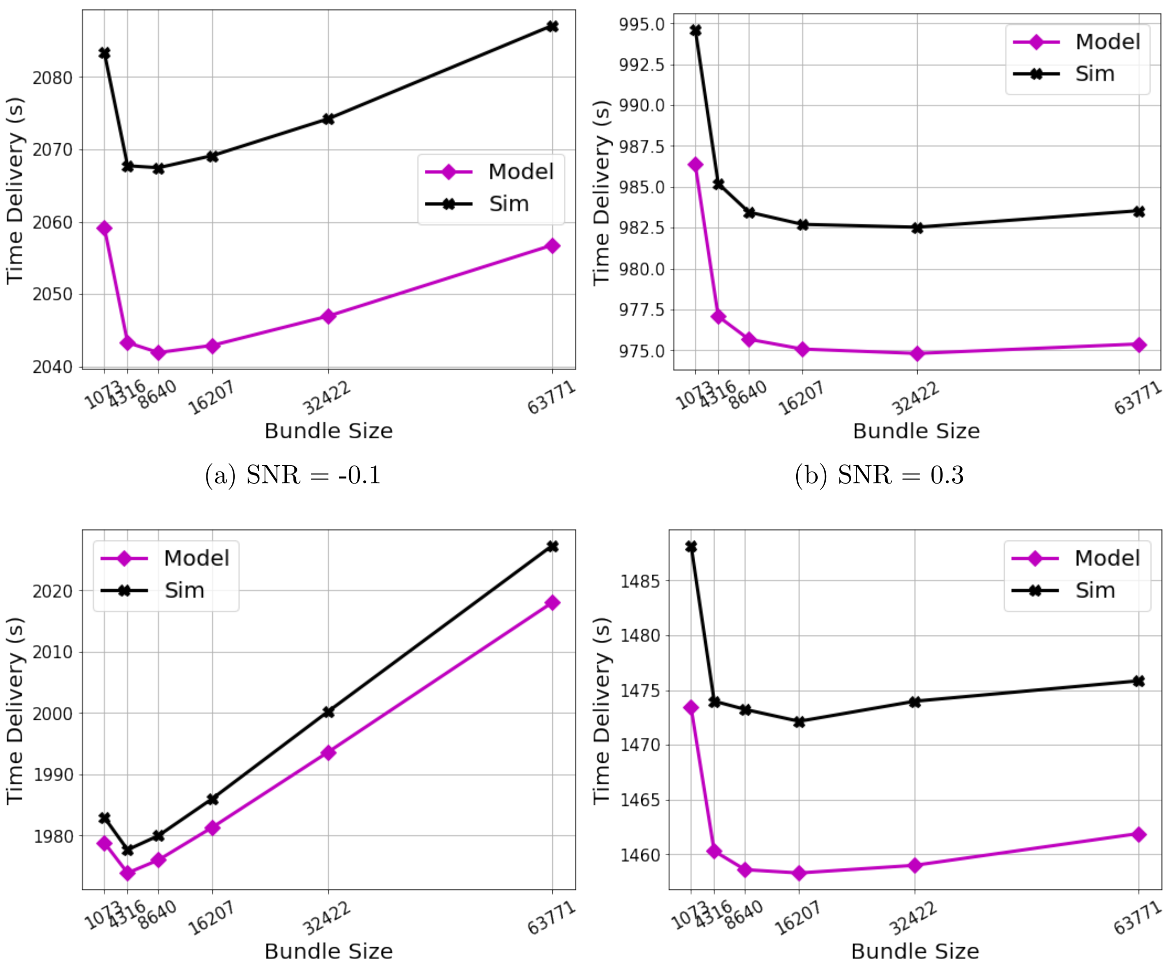
Here, we will analyze the delivery time taken by different bundles under different conditions to achieve our required file failure of $P_{fR} = 10^{-3}$. We use Equation (3.17) for these file delivery time computation. Each evaluation is performed for the number of rounds (N_R) obtained in section 5.3.1, which satisfied the required file failure probability condition. Therefore, the delivery time we obtained here is the average time a file takes for transmitting with finite transmission rounds that are required to achieve targeted file failure probability.

Table 5.7 provides the exact delivery time values for transmitting a file of 20 MB with different SNR scenarios to obtain $P_{fR} = 10^{-3}$. Figure 5.4 shows the time delivery plots for some selected SNR conditions in enlarged form to see the behavior. Both theoretical (model) and simulation values are plotted in the figure. The percentage difference of file delivery time of model with respect to simulation is plotted in Figure 5.5. The figure shows low percentage difference values for selected SNRs. The percentage difference between theoretical and simulation delivery time ranges from 0.191-2.825 %, which are low and acceptable, validating our analytical model to be correct.

Table 5.7. Time delivery values obtained for $P_{fR} = 10^{-3}$ with different SNR

L_B	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	random
1073	2059.17	1189.04	1147.06	1602.44	986.36	1978.82	1297.39	1473.4
4316	2043.28	1175.52	1133.89	1586.51	977.07	1973.9	1284.26	1460.3
8640	2041.9	1173.49	1131.9	1584.14	975.67	1976.02	1282.3	1458.6
16207	2042.9	1172.62	1131.05	1583.19	975.07	1981.36	1281.48	1458.3
32422	2046.95	1172.26	1130.68	1584.15	974.8	1993.64	1281.18	1459.0
63771	2056.76	1173.14	1131.49	1586.71	975.38	2018.01	1282.16	1461.9

The plot's overall nature shows decrease in delivery time as we move from smaller bundle size (1073 B) to medium bundle sizes (4316 B, 8640 B) and then again increase in time for other higher bundle sizes (32422 B, 63771 B). Three factors can justify this nature: (a) bundle error probability, P_B, (b) the number of bundles, N_B, needed to be transmitted and (c) bundle transmission time, T_B. A larger bundle size means fewer bundles but larger error probability, causing more bundle failure. Additionally, these bundles have a higher transmission time than others. Due to these reasons, large sizes of bundle takes a longer time to deliver. On the contrary, if we consider a smaller size as 1073 B, then these have vast numbers of bundles required to be transmitted. Hence, even though bundle error probability is lesser, it usually takes longer to deliver the file. This case is valid for all SNR scenarios shown in Figures 5.4a, 5.4b and 5.4c. Figure 5.4b (SNR = 0.3 dB) shows that delivery time for the larger bundle size is lesser than that obtained for the smallest bundle size, unlike Figure 5.4c (SNR = 0.4 dB) for which delivery time for the larger bundle size is much higher. This variation is a result of the number of transmission rounds. For 0.3 SNR, all computations are for two transmission rounds, whereas for 0.4 dB, the number of transmission increases from two to three with increase in the bundle size. Figure 5.4d is the delivery time obtained for a random SNR scenario. The nature of the plot is also similar to other fixed SNR cases. As a result of uniform choice among listed channel conditions, the



(a) SNR = -0.1

(b) SNR = 0.3

(c) SNR = 0.4

(d) SNR = Random

Figure 5.4. Time delivery plots for different SNR to achieve required file failure probability $P_{fR} = 10^{-3}$ showing results from simulation and model

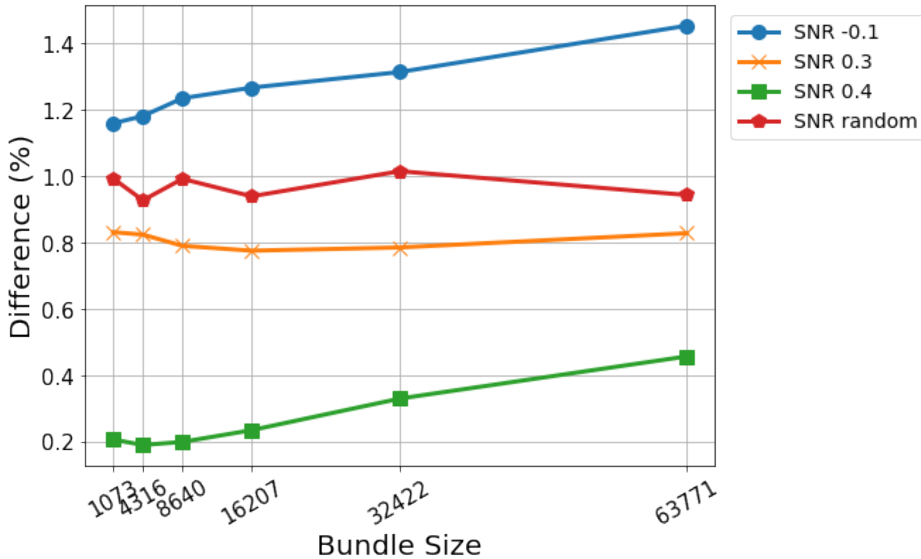


Figure 5.5. File delivery time percentage difference of model with respect to simulation under different SNR

delivery time obtained for a random case is the average of all delivery times of the fixed SNR cases.

The important point to consider is the bundle size that gives minimum delivery time for the given condition of file failure probability. If we go through each SNR case, for SNR $= -0.1$ dB, bundle size 8640 B is best for file transmission as it takes less time than other bundle sizes. Similarly, for SNR $= 0.0, 0.1, 0.5, 0.3$ dB, bundle size 32422 B seems to be good. Whereas, bundle size 16207 B takes minimum delivery time than others for 0.2 dB SNR and random SNR case and 4316 B takes minimum delivery time for 0.4 dB SNR.

Furthermore, if we compare different SNR cases for the same bundle size (as 1073 B), the delivery time decreases as SNR varies from -0.1 to 0.1 dB and then increases for 0.2 dB and again decreases for 0.3 dB. This increment and decrement in delivery time are highly governed by variation in FER and code rate for each turbo frame given in Tables 2.2 and 2.1 (appendix A). As FER increases, bundle error probability also increases, resulting in a rise in delivery time and vice-versa. For SNR -0.1 and 0.4 dBs, even though the FER is almost equal, causing bundle error probability and number of rounds required to be the same, the delivery time is still different. This variation is due to a change in bundle transmission time as the former has a turbo code rate (r) of 1/6 and the latter has a code rate of 1/3.

For our study of SNR = [-0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, random], 0.3 dB SNR has the smallest time delivery compared to others making it best channel bit-SNR for file transmission.

5.3.3 NET DATA TRANSMISSION RATE R_{net} ACHIEVED FOR P_{fR}

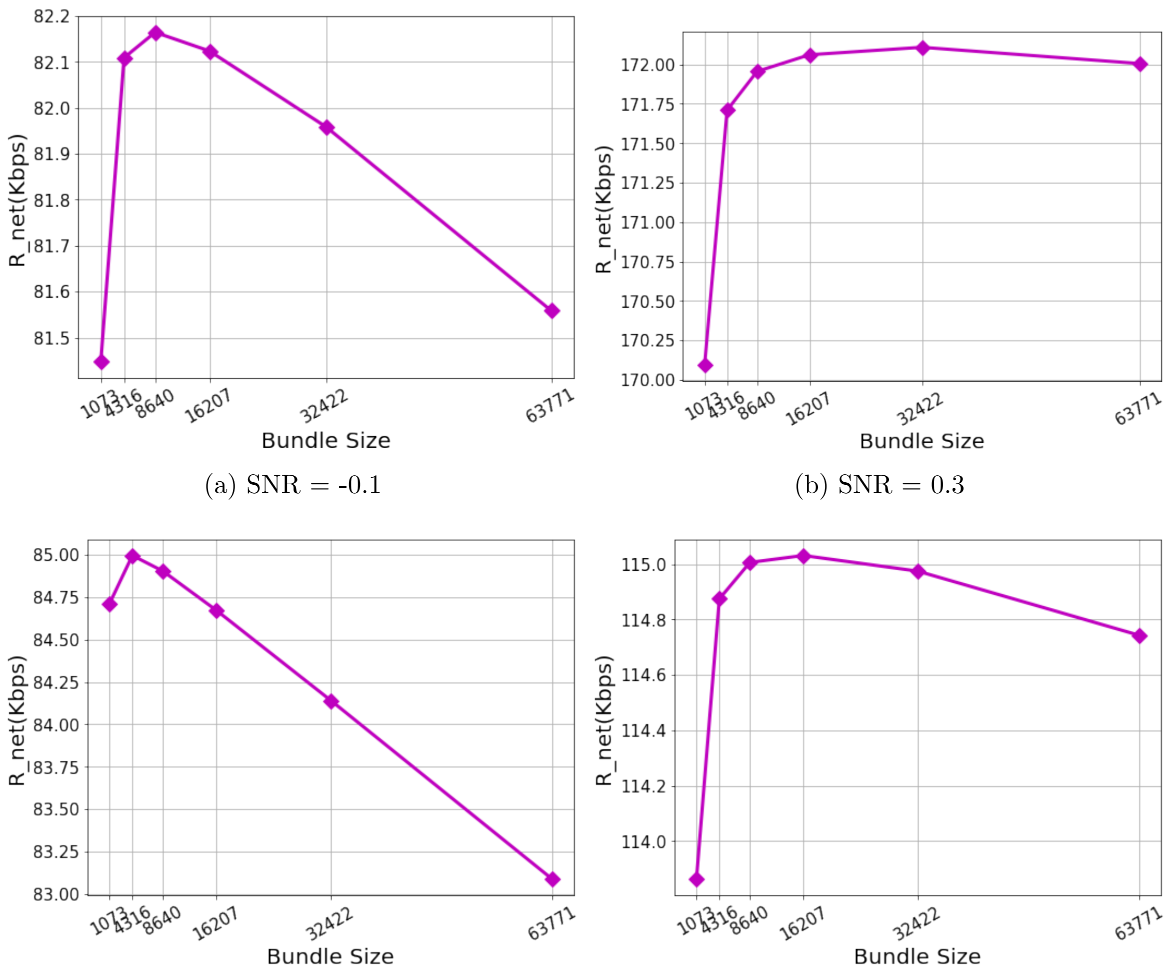
This section discusses the results for net data transmission rate (R_{net}) obtained for different SNR scenarios to maintain the file failure probability of 10^{-3}. We compute R_{net} using Equation (3.23) using the delivery time values from Tables 5.7 and file failure probability from Tables 5.5 and 5.6. Instead of the targeted file failure, $P_{fR} = 10^{-3}$, we use actual file failure values for the net data transmission rate computation. Because the targeted file failure only acts as the upper bound for the system but is not actually achieved during transmission.

Table 5.8 gives the computed net data transmission rate values for each SNR in Kbps. Figure 5.6 shows the plot obtained for these values for SNR -0.1, 0.3, 0.4 and random, respectively. This value of R_{net} mainly depends on two factors: file failure probability and delivery time. Less file failure and delivery time result in a higher data transmission rate.

Table 5.8. Net data transmission rate (R_{net}) (Kbps) values obtained for $P_{fR} = 10^{-3}$ with different SNR

L_B	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	random
1073	81.448	141.099	146.263	104.692	170.092	84.711	129.313	113.862
4316	82.109	142.721	147.962	105.725	171.709	84.995	130.629	114.875
8640	82.165	142.968	148.222	105.858	171.956	84.903	130.820	115.006
16207	82.124	143.073	148.333	105.879	172.061	84.672	130.889	115.032
32422	81.959	143.116	148.381	105.906	172.109	84.141	130.888	114.975
63771	81.559	143.006	148.274	105.735	172.006	83.088	130.727	114.743

Higher the value of data transmission rate, better is the performance. Hence, for each scenario, the bundle size that maximizes the data transmission rate is appropriate. Figure 5.6a shows bundle size 8640 B provides the maximum R_{net} for SNR = -0.1 dB, making it best for file transmission. For SNR = 0.4 dB, bundle size 4316 B has the maximum



(a) SNR = -0.1

(b) SNR = 0.3

(c) SNR = 0.4

(d) SNR = Random

Figure 5.6. Net data transmission rate (R_{net}) plot for different SNR cases to obtain the targeted file failure probability of $P_{fR} = 10^{-3}$

R_{net}. Similarly, for SNR = 0.0, 0.1, 0.3, 0.5, the maximum is obtained for the bundle size 32422 B, whereas for SNR 0.2 and random, 16207 B bundle size provides the maximum R_{net}. These best bundle sizes that maximize the net data transmission rate for each SNR case are summarized in Table 5.9. Hence, this study fulfills the main objective of analysis.

Table 5.9. Best bundle size for file transmission to achieve $P_{fR} = 10^{-3}$ for different SNR case

SNR	**Bundle size**
-0.1	8640 B
0.0	32422 B
0.1	32422 B
0.2	16207 B
0.3	32422 B
0.4	4316 B
0.5	32422 B
random	16207 B

CHAPTER 6

CONCLUSION

In this thesis, we studied the performance of BP when it is employed with turbo code in the link layer. We have shown from file failure analysis that failure probability is relatively low even at a low number of transmission rounds (as 2^{nd}, 3^{rd}). It indicated that the use of error-correcting turbo code could enhance the performance of BP. Next, we have studied the analytical model for mean file delivery time useful for any number of transmission rounds. We have validated this model by simulation.

The main objective of the thesis, that is, finding the best bundle size to maximize the net data transmission rate, was fulfilled. For this, first, we computed the delivery time with the finite number of rounds required to achieve our targeted file failure probability. Then, we found the net data transmission rate. We have analyzed this under two different scenarios, (a) Equal SNR and (b) Unequal (random) SNR. For an equal SNR case, it was observed that the bundle size that maximizes the net data transmission rate varies with SNR chosen for transmission. In the case of unequal SNR, we found bundle size 16207 B to perform well, providing maximum data transmission rate. In general, we can conclude that middle bundle sizes would provide better performance for all scenarios than the smallest and the highest bundle size.

We had these conclusions based on certain assumptions stated in this thesis. However, in a real scenario, channel SNR could change at any time duration, not necessarily change with each transmission round. Hence, we can further expand this analysis by predicting the real-time channel condition, which could be more accurate. Moreover, thesis studies are carried out for symmetric channel conditions and can be expanded for asymmetric channel cases. We have also limited the study within the selected range of channel bit-SNR only, leaving the way for further analysis with wide ranges of SNR for the future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[2] D. G. Book, "Rationale, Scenarios, and Requirements for DTN in Space," 2010.

[3] F. Warthman, "Delay-tolerant networks (DTNs) a tutorial," *http://ipnsig.org/reports/DTNTutorial11.pdf*, 2003.

[4] K. Scott and S. Burleigh, "Bundle protocol specification," 2007.

[5] M. Demmer, J. Ott, and S. Perreault, "Delay tolerant networking TCP convergence layer protocol," *IETF Draft*, 2008.

[6] H. Kruse, S. Jero, and S. Ostermann, "Datagram Convergence Layers for the Delay-and Disruption-Tolerant Networking (DTN) Bundle Protocol and Licklider Transmission Protocol (LTP)," *RFC*, vol. 7122, pp. 1–11, 2014.

[7] A. Sabbagh, R. Wang, K. Zhao, and D. Bian, "Bundle protocol over highly asymmetric deep-space channels," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2478–2489, 2017.

[8] N. Bezirgiannidis, S. Burleigh, and V. Tsaoussidis, "Delivery time estimation for space bundles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1897–1910, 2013.

[9] K. Zhao, R. Wang, S. C. Burleigh, A. Sabbagh, W. Wu, and M. De Sanctis, "Performance of bundle protocol for deep-space communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2347–2361, 2016.

[10] Q. Yu, R. Wang, K. Zhao, W. Li, X. Sun, J. Hu, and X. Ji, "Modeling RTT for DTN protocol over asymmetric cislunar space channels," *IEEE Systems Journal*, vol. 10, no. 2, pp. 556–567, 2014.

[11] B. Cao, R. Wang, A. Sabbagh, S. Peng, K. Zhao, J. A. Fraire, G. Yang, and Y. Wang, "Expected File-Delivery Time of DTN Protocol over Asymmetric Space Internetwork Channels," in *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 147–151, IEEE, 2018.

[12] N. Alessi, C. Caini, T. de Cola, S. Martin, and J. P. Mayer, "DTN Performance in Complex Deep-Space Networks," in *2018 9th Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, pp. 1–7, IEEE, 2018.

[13] F. Noviani, D. Stiawan, S. D. Siswanti, T. W. Septian, M. A. Riyadi, F. Aljaber, and R. Budiarto, "Analysis of custody transfer on moving bundle protocol of wireless router in delay tolerant network (DTN)," in *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 50–53, IEEE, 2017.

[14] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[15] R. Adhikary, J. N. Daigle, and L. Cao, "Protocol design for content transfer in deep space network," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

APPENDICES

APPENDIX A

FER TABLE FOR TURBO CODE

Here, Table A-3 through A-6 provides Frame Error Rate (FER) for turbo code for different bit-SNR (E_b/N_0) and different code rates. This FER is for the ideal case of a system without any data loss. These results were obtained by using 10 decoding iterations per frame performed at JPL. We have used these values of FER during performance evaluation in the thesis based on channel bit-SNR and turbo code.

Table A-3. Rate 1/2 Turbo Code Baseline Data

E_b/N_0	Turbo Code			
	(1784, 1/2)	(3568, 1/2)	(7136, 1/2)	(8920, 1/2)
0.4		1.0000E+00		
0.5		—		1.0000E+00
0.6	7.5000E–01	8.0000E–01		8.8496E–01
0.7	—	—	5.5266E–01	5.9524E–01
0.8	3.8931E–01	2.6247E–01	1.8382E–01	1.8939E–01
0.9	—	—	2.5046E–02	2.0309E–02
1.0	7.5529E–02	2.2411E–02	1.4271E–03	8.4691E–04
1.1	—	—	7.7270E–05	3.5650E–05
1.2	7.9605E–03	3.1980E–04	7.6700E–06	1.5510E–05
1.3	—	—		1.2280E–05
1.4	3.2503E–04	4.0800E–06		6.7700E–06
1.5	—	—		
1.6	1.1620E–05	1.5800E–06		
1.7	—	—		
1.8	3.7500E–06	5.8000E–07		
1.9	—	—		
2.0	2.2500E–06	6.0000E–08		
2.1	—	—		
2.2	7.5000E–07	1.1000E–07		

Table A-4. Rate 1/3 Turbo Code Baseline Data

E_b/N_0	Turbo Code			
	(1784, 1/3)	(3568, 1/3)	(7136, 1/3)	(8920, 1/3)
–0.4	9.9020E–01			
–0.3	—			
–0.2	9.0090E–01			
–0.1	—			
0.0	6.8493E–01			8.3333E–01
0.1	—		4.3328E–01	4.9505E–01
0.2	2.9762E–01	1.8065E–01	1.0761E–01	9.7752E–02
0.3	—	5.1557E–02	1.0989E–02	8.9847E–03
0.4	4.7174E–02	9.0463E–03	4.4099E–04	2.0755E–04
0.5	—	9.5734E–04	1.0050E–05	2.8730E–05
0.6	4.4583E–03	4.1120E–05		1.4360E–05
0.7	—	4.5100E–06		1.1490E–05
0.8	9.2350E–05			
0.9	—			
1.0	1.9100E–06			

Table A-5. Rate 1/4 Turbo Code Baseline Data

E_b/N_0	Turbo Code			
	(1784, 1/4)	(3568, 1/4)	(7136, 1/4)	(8920, 1/4)
–0.4				
–0.3				9.9010E–01
–0.2				8.4746E–01
–0.1			3.3866E–01	3.7594E–01
0.0	2.3810E–01	1.3508E–01	6.7147E–02	7.3260E–02
0.1	1.4006E–01	3.1327E–02	5.5659E–03	2.9790E–03
0.2	3.8865E–02	4.1032E–03	2.9471E–04	5.4510E–05
0.3	9.9325E–03	4.9503E–04	1.0723E–04	2.5700E–06
0.4	2.1765E–03	6.0170E–05		2.0300E–06
0.5	4.9670E–04			1.7100E–06
0.6	7.7840E–05			7.8000E–07
0.7	1.0430E–05			
0.8	3.1900E–06			
0.9	1.7100E–06			
1.0	9.7000E–07			
1.1	5.1000E–07			
1.2	6.6000E–07			

Table A-6. Rate 1/6 Turbo Code Baseline Data

E_b/N_0	Turbo Code			
	(1784, 1/6)	(3568, 1/6)	(7136, 1/6)	(8920, 1/6)
–0.50				9.0909E–01
–0.45				7.2464E–01
–0.40			4.7659E–01	4.7619E–01
–0.35			—	2.8653E–01
–0.30	2.7855E–01		1.1924E–01	9.9701E–02
–0.25	—		—	3.2362E–02
–0.20	1.4793E–01	4.8632E–02	1.2559E–02	6.6542E–03
–0.15	—	—	—	1.1703E–03
–0.10	5.1203E–02	7.2787E–03	6.4147E–04	1.3089E–04
–0.05	—	—	—	1.6310E–05
0.0	1.1990E–02	9.2768E–04	4.5750E–05	5.5200E–06
0.05	—	—		4.3200E–06
0.10	3.5388E–03	5.9720E–05		2.4000E–06
0.15	—	—		
0.20	5.8113E–04	9.6500E–06		
0.25	—			
0.30	5.7830E–05			
0.35	—			
0.40	9.9500E–06			
0.45	—			
0.50	2.3400E–06			

APPENDIX B

BUNDLE PARAMETERS

This appendix provides the values of the number of bundles (N_B), bundle error probability (P_B) and bundle transmission time (T_B) for given bundle size (L_B) under given channel bit-SNR. Table B.1 through B.7 shows these exact values for the 20 MB file. Equations (3.1), (3.7) and (3.5) are used for the computation of N_B, P_B and T_B, respectively. These values can be fed directly during simulation for computing the file delivery time. All L_B are in bytes and T_B is in seconds.

Table B.1. Corresponding values of P_B and T_B for different bundle size for SNR= -0.1

L_B	N_B	P_B	T_B
1073	20069	0.00013089	0.0256233
4316	4891	0.00052345	0.1024932
8640	2436	0.00104664	0.2049865
16207	1297	0.00196155	0.3843498
32422	648	0.00391925	0.7686996
63771	330	0.00769326	1.5117759

Table B.2. Corresponding values of P_B and T_B for different bundle size for SNR= 0.0

L_B	N_B	P_B	T_B
1073	20069	5.51999E-06	0.0256233
4316	4891	2.20798E-05	0.1024932
8640	2436	4.415914E-05	0.2049865
16207	1297	8.27968E-05	0.3843498
32422	648	0.00016558	0.7686996
63771	330	0.00032562	1.5117759

Table B.3. Corresponding values of P_B and T_B for different bundle size for SNR= 0.1

L_B	N_B	P_B	T_B
1073	20069	0.00013089	0.0256233
4316	4891	0.00052345	0.1024932
8640	2436	0.00104664	0.2049865
16207	1297	0.00196155	0.3843498
32422	648	0.00391925	0.7686996
63771	330	0.00769326	1.5117759

Table B.4. Corresponding values of P_B and T_B for different bundle size for SNR= 0.2

L_B	N_B	P_B	T_B
1073	20069	5.451E-05	0.01708221
4316	4891	0.00021802	0.06832885
8640	2436	0.00043599	0.13665771
16207	1297	0.00081733	0.25623321
32422	648	0.001634	0.51246643
63771	330	0.00321101	1.00785064

Table B.5. Corresponding values of P_B and T_B for different bundle size for SNR= 0.3

L_B	N_B	P_B	T_B
1073	20069	2.57E-06	0.01708221
4316	4891	1.027996E-05	0.06832885
8640	2436	2.055981E-05	0.13665771
16207	1297	3.85493E-05	0.25623321
32422	648	7.70971E-05	0.51246643
63771	330	0.0001516186	1.00785064

Table B.6. Corresponding values of P_B and T_B for different bundle size for SNR = 0.4

L_B	N_B	P_B	T_B
1073	20069	0.00020755	0.01281166
4316	4891	0.00082994	0.05124664
8640	2436	0.00165919	0.10249328
16207	1297	0.00310873	0.19217491
32422	648	0.00620779	0.38434982
63771	330	0.01217203	0.75588798

Table B.7. Corresponding values of P_B and T_B for different bundle size for SNR = 0.5

L_B	N_B	P_B	T_B
1073	20069	2.873E-05	0.01281166
4316	4891	0.000114915	0.05124664
8640	2436	0.000229816	0.10249328
16207	1297	0.000430863	0.19217491
32422	648	0.000861541	0.38434982
63771	330	0.001693658	0.75588798

APPENDIX C

ERROR POSITION: EXPECTED, OBSERVED AND TEST STATISTICS VALUE

The appendix provides values of observed and expected error position and the test statistics obtained for the goodness of fit test performed in section 3.1 to show that error positions at any transmission are equally likely (uniform distribution). Table C.1 shows the sampled or observed data of error positions count obtained by Simulation for a different number of transmissions ($N_T = [1, 2, 3, 4, 5]$). The simulation was performed for 100 bundles with an error probability of 0.345 with 1000000 instances. Table only shows the first 25 bundles values. Table C.2 shows the expected value for error positions for the same setup. χ^2 test statistics obtained from Equation (3.11) is provided in Table C.3 for corresponding observed and expected error positions.

Table C.1. observed data of error position for different number of transmission (N_T)

error position $\backslash N_T$	1	2	3	4	5
1	345213	119010	41166	14109	4822
2	344744	119211	41014	13906	4768
3	345170	119343	41190	14232	5017
4	344644	118691	40807	14108	4933
5	346620	119525	41096	14064	4814
6	344781	118994	40862	14179	4833
7	345184	119164	40903	14158	4921
8	344378	118749	41124	13932	4830
9	345208	119200	41411	14160	4912
10	345110	119162	41220	13981	4844
11	344904	118727	41014	14080	4873
12	345661	119479	41345	14179	4832
13	345084	119621	41490	14277	4920
14	345671	119201	41382	14329	4931
15	345008	118780	40932	14133	4916
16	345057	119013	40875	14054	4911
17	344976	119380	41253	14226	4963
18	345309	119069	41149	14172	4839
19	344888	119075	41019	13983	4715
20	345240	119234	41256	14103	4776
21	345217	119444	40995	14205	4815
22	344443	118529	40780	14050	4962
23	344116	118720	40978	14121	4875
24	345395	118901	41092	14217	4946
25	345113	119107	41211	14140	4761

Table C.2. Expected values for different transmission rounds

Number of transmissions				
1	2	3	4	5
345000	119025	41063.625	14166.95	4887.59

Table C.3. Test Statistics

Number of transmissions	χ^2 test statistics
1	61.434267
2	85.503323
3	97.851963
4	83.774592
5	99.661003

VITA

Babita Pradhan, born in Bhaktapur, Nepal in 1993 received the B.E degree in Electronics and Communication Engineering from Pulchowk Campus, Institute of Engineering, Nepal in 2015. Starting August 2018, she is working towards the M.S degree in electrical engineering at The University of Mississippi, University, MS. During her M.S. program, she worked as a teaching assistant for Dr. Ramanarayan Viswanathan.