Optimal Threshold Policy for Opportunistic Network Coding Under Phase Type Arrivals

by

Charith Gunasekara

A dissertation submitted to The Faculty of Graduate Studies

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

The University of Manitoba

Winnipeg, Canada

April 2016

Dedicated to my parents.

Abstract

Network coding allows each node in a network to perform some coding operations on the data packets and improve the overall throughput of communication. However, network coding cannot be done unless there are enough packets to be coded so at times it may be advantageous to wait for packets to arrive.

We consider a scenario in which two wireless nodes each with its own buffer communicate via a single access point using network coding. The access point first pairs each data packet being sent from each node and then performs the network coding operation. Packets arriving at the access point that are unable to be paired are instead loaded into one of the two buffers at the access point. In the case where one of the buffers is empty and the other is not network coding is not possible. When this happens the access point must either wait for a network coding opportunity, or transmit the unpaired packet without coding. Delaying packet transmission is associated with an increased waiting cost but also allows for an increase in the overall efficiency of wireless spectrum usage, thus a decrease in packet transmission cost. Conversely, sending packets un-coded is associated with a decrease in waiting cost but also a decrease in the overall efficiency of the wireless spectrum usage. Hence, there is a trade-off between decreasing packet delay time, and increasing the efficiency of the wireless spectrum usage.

We show that the optimal waiting policy for this system with respect to total cost, under phase-type packet arrivals, is to have a separate threshold for the buffer size that is dependent on the current phase of each arrival. We then show that the solution to this optimization problem can be obtained by treating it as a double ended push-out queueing theory problem. We develop a new technique to keep track of the packet waiting time and the number of packets waiting in the two ended push-out queue. We use the resulting queueing model to

resolve the optimal threshold policy and then analyze the performance of the system using numerical approach.

Acknowledgments

I would like to express heartiest gratitude to my advisors Professor Attahiru S. Alfa and Professor Pradeepa Yahampath for their continuous support, guidance and financial assistance throughout my research tenure for the last 5 years at the University of Manitoba.

I would also like to thank my family and all my friends for their love and support which helped me to achieve this stage.

Table of Contents

List of Figures vi

List of Acronyms vii

1 Introduction 1
  1.1 Research Motivation . . . 5
  1.2 Contribution and Outline of the Thesis . . . 5

2 Background and Literature Survey 8
  2.1 Network Coding Opportunities . . . 12
  2.2 Opportunistic Network Coding in Wireless Two Way Relay . . . 16
  2.3 Related Queueing Theoretic Analysis of Opportunistic Network Coding . . . 21
  2.4 General Arrivals and Use of Phase Type Distribution . . . 23

3 Determining the Optimal Waiting Policy 26
  3.1 System Model and Problem Statement . . . 27
  3.2 Markov Decision Process for the System . . . 28
  3.3 Proof of the Optimal Threshold Policy . . . 31
  3.4 Convexity of the Problem . . . 38
  3.5 Optimal Threshold Policy . . . 40
  3.6 Special Case of Systems with Unobservable Arrival Phase . . . 42

4 Discrete Time Markov Chain to Solve for the Optimal Thresholds 45
  4.1 Arrival Process . . . 46
  4.2 Service Process . . . 47
  4.3 Discrete Time Markov Chain . . . 48
  4.4 Solving the Discrete Time Markov Chain . . . 54
  4.5 Finding the Optimal Solution Using a Gradient Descent Algorithm . . . 57

5 Numerical Results 60

6 Conclusion and Future Work 72
  6.1 Possible Future Extensions . . . 73

A Derivation of the rest of the DTMC in (4.3) 78

Bibliography 81

List of Figures

1.1 Network Coding in a Butterfly Network 2
1.2 Network Coding at an Access Point. 4

2.1 Network Coding for Unicast Data 10
2.2 Network Coding for Multicast Data 11

3.1 Two Way Relay Access Point with Phase Type Packet Arrivals 27
3.2 Markov Decision Process 29

4.1 Proposed Two Ended PH/PH/1 Queue: Push-out Senario 46
4.2 Proposed Two Ended PH/PH/1 Queue: Coded Packet Transmission 47
4.3 Level Dependant QBD Structure 55

5.1 Average Number of Waiting Packets in Each Queue: Queue 1: Arrival 2, Queue 2: Arrival 1 66
5.2 Average Waiting Time per Packet vs Buffer Size: Queue 1: Arrival 2, Queue 2: Arrival 1 67
5.3 Average Number of Transmissions vs Buffer Size: Queue 1:Arrival 2, Queue 2: Arrival 1 68
5.4 Comparison of Average Cost: Queue 1:Arrival 1, Queue 2: Arrival 3 69
5.5 Comparision of Average Cost: (a) Arrival 1 and Arrival 2, (b) Arrival 1 and Arrival 2, (c) Arrival 1 and Arrival 4 70
5.6 Comparision of Coding Ratio (a) Arrival 1 and Arrival 2 (b) Arrival 1 and Arrival 3 (c) Arrival 1 and Arrival 4 71

6.1 TDMA Wireless Start Network 74
6.2 Time Slot Arrangement of One TDMA Cycle 74
6.3 Reduced transmittion time slots with network coding 75
6.4 Proposed New Data Queues Pairs in the AP 75
6.5 Network Throughput vs M (N=30). 76

List of Acronyms

ANC	Analog Network Coding
AP	Access Point
COPE	Opportunistic Coding Architecture
CPPA	Coded Packets Priority Access
DMAP	Discrete time Markov Arrival Process
DT	Direct Transmission
DTMC	Discrete Time Markov Chain
EM	Expected Maximization
FIFO	First In First Out
FSMH	Four-Slot Multi Hopping
GDA	Gradient Descent Algorithm
HBC	Hybrid Broadcast
HoL	Head of Line
MABC	Multiple Access Broadcast
MAC	Medium Access Protocol
MDP	Markov Decision Process

P2P	Peer-to-Peer
PH	Phase Type Distribution
PH/PH/1	Phase/ Phase/1 Server
QBD	Quasi Birth Death
TCP	Transport Control Protocol
TDBC	Time Division Broadcast
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol

Chapter 1

Introduction

The idea of network coding was first conceived, by Alhswede et al. [1], as means to improve the throughput of transmitted packets in wireless networks using bitwise XOR addition (modulo 2 addition). Since its conception, the popularity of network coding has increased considerably within the telecommunications research community. Network coding consists of intermediate nodes algebraically encoding clusters of data packets together using bitwise XOR or linear combination in a Galois field. At the same time, nodes at the destination decode these clusters to recover the original data. To generate multi-node network flows, Ahlswede et al. applied the idea of network coding at intermediate nodes. Network coding provides network nodes the ability to decode and re-encode transmitted information. Therefore, the retransmission of messages are a function of incoming messages, as opposed to traditional routing. The advantages of network coding includes increased network throughput, reduced energy consumption, and an enhanced network reliability.

Network coding can be classified into inter-session and intra-session, depending on the type of operations on sessions [2]. Intra-session network coding deals with adding redundancy into individual network flows. It allows for intermediate nodes to encode packets within the same session without the knowledge of the other nodes in the network [3] [4]. For the problem of minimum-cost multicast network coding involved with intra-session network coding in [5],

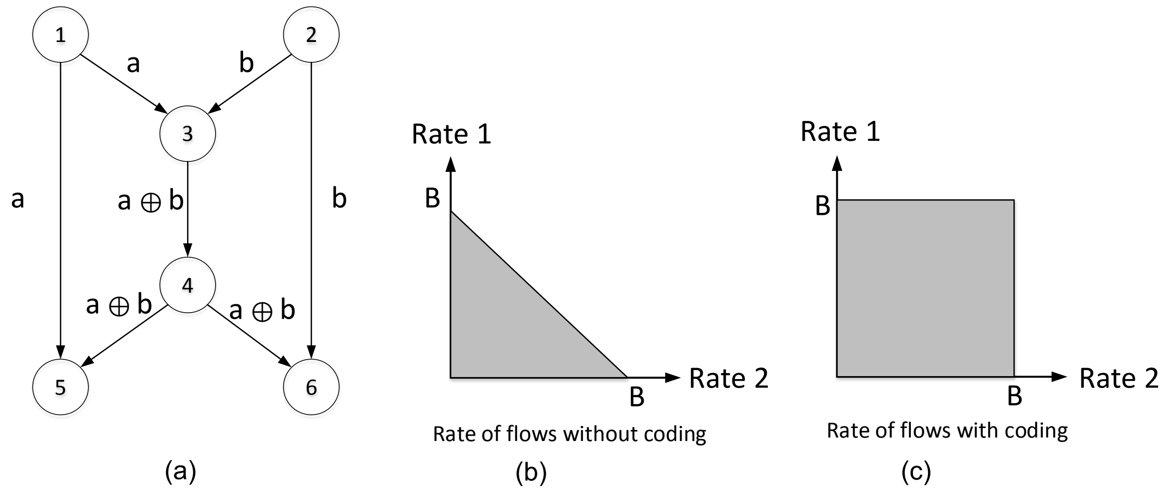


Figure 1.1: Network Coding in a Butterfly Network

Lun et al. proposed a dual subgradient method. They extended the rate control techniques in [6] [7] so that it could be applied to wireless network coding. In [8], the rate stability region for a wireless network, with or without correlated arrivals, is characterized. Intersession network coding allows coding several network flows together to reduce the number of transmissions required for the overall communication which is the main focus on this thesis. The best examples for intersession network coding are the butterfly network and opportunistic XOR coding explained below.

Originally, the concept of network coding was intended to be an alternate routing method for error free network links to improve the overall throughput of a network. Normally, store-and-forward routers send an identical copy of the received packet to the destination without modifying the body of the message. However, network coding proposes that the nodes in the network modify the packet's content before it is transmitted. Using reasoning based in information theory, Alhswede et al. [1] prove two theories important to the capacity of a network. First, the multicast capacity of a network is equal to the minimum of the maximum flows between the source and any individual destination. Secondly, traditional routing methods alone are insufficient to achieve this fundamental limit. Therefore, the nodes are required to mix the data units received from neighbouring nodes using XOR coding operations in order to achieve higher throughputs.

The initial concept of network coding is best explained by the butterfly network, shown in Figure 1.1 (a) [1], where each edge is assumed to have the same bit rate capacity, B bps. Nodes 1 and 2 send two multicast flows, a and b, to nodes 5 and 6 through direct links and also through the shared path, link 3-4. If node 3 performs traditional packet switching, link 3-4 has to be shared between streams a and b. Therefore, in this instance only node 1 or node 2 can transmit data at the maximum B bps rate, but not both at the same time. As shown in the graph in Figure 1.1 (b) the data rate of one node has to be reduced to allow data from the other nodes due to the switching bottle neck in link 3-4 using traditional routing. To overcome the bottleneck between nodes 3 and 4, node 3 combines the incoming bits through an XOR operation. Nodes 5 and 6 then use the bits they receive directly from nodes 1 and 2 to decode this XOR operation, and thus reconstruct the desired multicast flow. This allows both data streams, a and b, to be transmitted at the maximum rate of B bps as shown in the graph in Figure 1.1 (c).

Network coding, in its various forms, can be applied in every layer of the network protocol stack. However, this thesis is concerned with the use of network coding in the link layer only. Network coding applied to the link layer allows for XOR operations to be performed on data packets being transmitted in opposite directions through a wireless access point [9]. The basic principle and underlying advantages of this approach is explained in Figure 1.2. Suppose that two wireless nodes A and B are communicating with each other via an access point. Node A sends data packet X_1 to node B and, at the same time, node B sends data packet X_2 to node A via the access point. Figure 1.2 (a) shows how the access point receives these data packets from each node and then relays them to the destination nodes. Whereas the access point in the access point in Figure 1.2 (a) needs to access the wireless spectrum 4 times to complete the desired communication, the access point in Figure 1.2 (b) needs to access

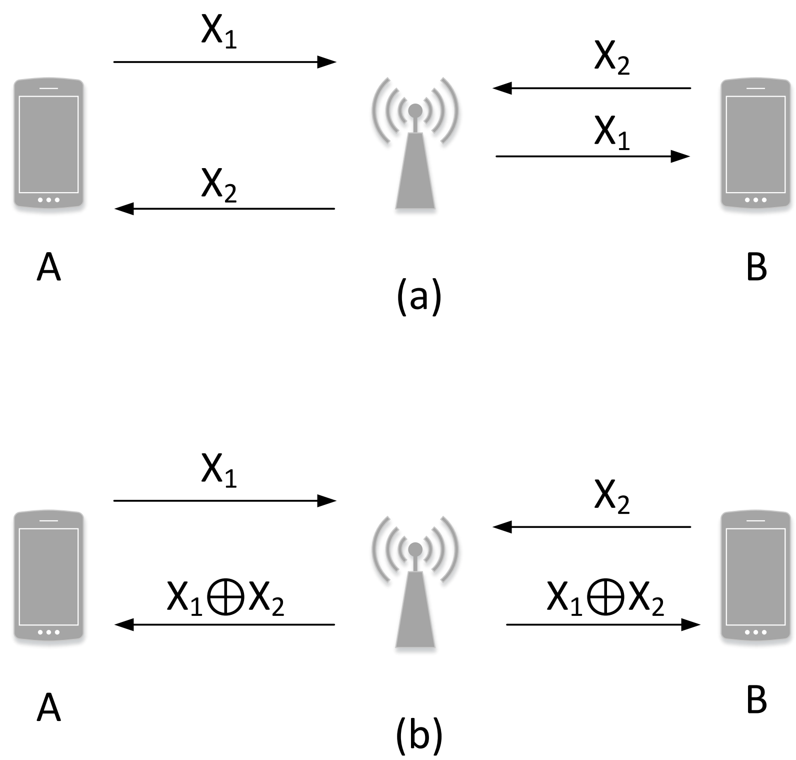


Figure 1.2: Network Coding at an Access Point.

the wireless spectrum only 3 times to broadcasts the coded $X_1 \oplus X_2$ packet. Since received packets are a function of transmitted packets, and since the transmitter knows everything about the data packets it is sending, it is able to easily decode received packets. If the access point has to make a decision about where to send data packets X_1 and X_2, it will need to access the wireless spectrum 4 times. However since each node decodes the combined $X_1 \oplus X_2$ data packets, the access point needs to only access the wireless spectrum 3 times- a reduction of 25%. However, the limiting factor to using this method is that the access point must be receiving data packets from both nodes simultaneously. When the access point is receiving data packets from one node at a time, it needs to wait indefinitely for a packet from the other node before it can apply network coding. As the waiting time for pairs of packets to join at the access point increases, the system performance decreases, and thus packet transmissions will be delayed and the overall throughput will be worse. However, if the access point does not wait long enough for pairs to join, then it will ultimately have to transmit many uncoded data packets before two packets arrive there to be coded.

1.1 Research Motivation

Network coding necessitates the need for two questions to be answered: When, and for how long, should the access point wait for packets to be paired? And, when should it transmit unpaired packets? To answer these questions, special techniques should be used to determine what decision should be made for an arbitrary situation that will minimize delays and maximize network throughput. Telecommunication systems usually have correlated inter-arrival times and can be accurately modelled using Phase type distribution [11] therefore is important to take that into consideration when we develop a system to answer these questions.

The main motivation of this thesis is to develop stochastic models as a mechanism to find the optimal decisions in network coding; when to delay the packets or transmit packets under general data packet arrival to minimize the total cost of the system.

1.2 Contribution and Outline of the Thesis

Chapter 2 presents the background of the opportunistic network coding and the previous work that has been done to analyze the trade-off between decreasing packet delay time and increasing the efficiency of the wireless spectrum usage. We also present the background of using phase type distribution in telecommunication traffic and the importance of this tool in analyzing general traffic arrivals.

In Chapter 3, we further analyze the packet delay time/wireless spectrum usage efficiency trade-off in opportunistic network coding under phase type distributions. The sequence of decisions involved with this system can be efficiently modelled and analysed by using a Markov decision process (MDP). Therefore we develop an MDP to model all possible states of the system which are dependent on whether the access point waits for packet arrivals or transmits un-coded packets. From the MDP we develop a proof that the optimal waiting policy for our system is to have a different threshold for the number of waiting packets at each of the two queues (the arrival phase vector), and that each threshold is dependent on the current phase of each queue. We then discuss the observability of the current phase of the arrival function and note that in some cases the phase vector of the arrival function may be unknown. To handle these situations we utilize a technique that probabilistically determines the arrival phase vector and consequently determines a threshold policy.

Solving for the optimal threshold policy using a MDP is not ideal since this problem has an infinite horizon. Due to its high complexity of implementation solving the MDP, in Chapter 4 we present a novel queueing theoretic model which enables us to analyze this system and ultimately solve for the optimal waiting policy efficiently. Our model records the packet waiting times and the number of packets in a First-In-First-Out (FIFO) Phase-Phase-1 (PH/PH/1) push out queueing model with two ended traffic. According to the best of our knowledge this is the first time a queueing model is presented that records the packet waiting times and the number of packets in a FIFO PH/PH/1 push out queue. Recoding the age of the packets in a queueing model with finite buffers is a challenging problem in general due to the fact that some packets can leave the system without getting served when the buffers are full. Our model is capable of recording the waiting time of the Head of Line (HoL) packet and the time that the HoL packet spends in the queue before it is either served, or pushed out. Based on the inter-arrival times of the arrival functions of each queue and the time that each HoL packet spends in its respective queue, the waiting time for each packet is calculated. We then show how our queueing model can be reduced to a level dependant, Quasi-Birth-Death (QBD) structure, for efficient computing.

In Chapter 5, we generate numerical and simulation results to analyse the behaviour of the proposed optimal policy and show how it reduces the total cost compared the other coding policies in given the literature.

Finally, in Chapter 6, conclusions are summarized and we discuss the situations where the proposed policy is more efficient to be used. We also identify possible future directions of research extensions.

Chapter 2

Background and Literature Survey

Ahlswede et al. in [1] changed how the world viewed data communication over networks by introducing the concept of network coding. They showed that the algebraic combination of information at network nodes can increase the capacity of a network beyond conventional network routing. This concept characterised the beginning of a breakthrough research era in network communications.

Before network coding was introduced, the theory of coding was broadly divided into two categories: Source Coding and Channel Coding. Whereas Source Coding is the increasing of transmission efficiency by compressing information at the source nodes, Channel Coding transforms a noisy channel into a noiseless one by introducing redundant bits into the information sequence.

With the introduction of network coding, another branch of coding theory was revealed which performed packet-level algebraic coding at network nodes. The key concept of Network Coding is that each node in the network performs coding operations. Therefore, a node can transmit functions of previous messages received from incoming links onto outgoing links. Over the last decade, network coding theory and its applications have become increasingly widespread. This has encouraged the incorporation of well-established mathematical tools including algebra, graph theory, and optimization theory, into network coding. As a result, network coding has evolved into a much broader and more complex field within communications than when it was first introduced. Moreover, the strong connection between network coding applications and queueing theory has played a pivotal role in leading network coding to its current level.

Initially, network coding was used only for operations over binary fields and was later developed for applications with larger field sizes. Moving from the binary field to larger field sizes allows the nodes in the network to perform more complex mathematical operations on data packets. The concept of linear network coding was first proposed, in [12] and [13], as a fully distributed mechanism to replace traditional store-and-forward routing. These studies proposed that nodes in the network can generate linear combinations of data packets by using randomly generated coefficients. In doing so, the mechanism necessary to achieve theoretical min-cut max-flow multicast capacity, defined in [1], was brought to fruition. Since each of these algebraic combinations are modulo operations within a finite field, the mixing of packets, through linear network coding, does not increase the packet size. In order to perform the decoding procedure, the destination node performs Gaussian elimination on the incoming data packet combinations. The destination node first waits until it has received a sufficient number of linearly independent data packet combinations, along with their respective coding coefficients, before decoding them into their original form. One drawback to linear network coding is that the coding coefficients consume space in the packet headers. However, since the space occupied in the header is small in comparison to the packet payload, this overhead is often neglected.

Figure 2.1 shows an example of a typical linear network coding application for unicast data. Initially, the sender has four packets to be transmitted. Rather than transmitting the raw packets, the sender generates five linearly independent combinations of the four data packets before transmitting them. During the transmission, even if one packet is lost, the receiver can decode the four coded data packets by decoding the remaining four coded data

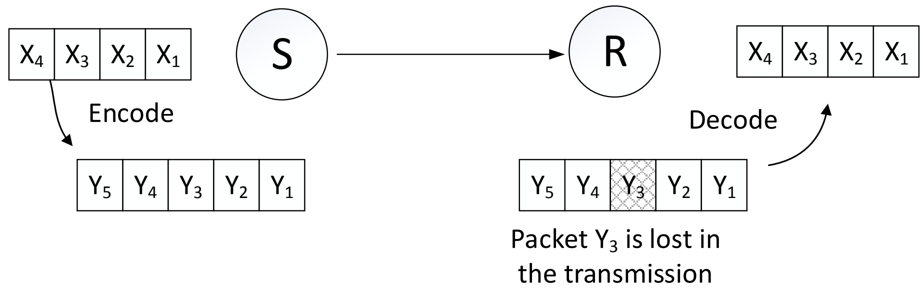


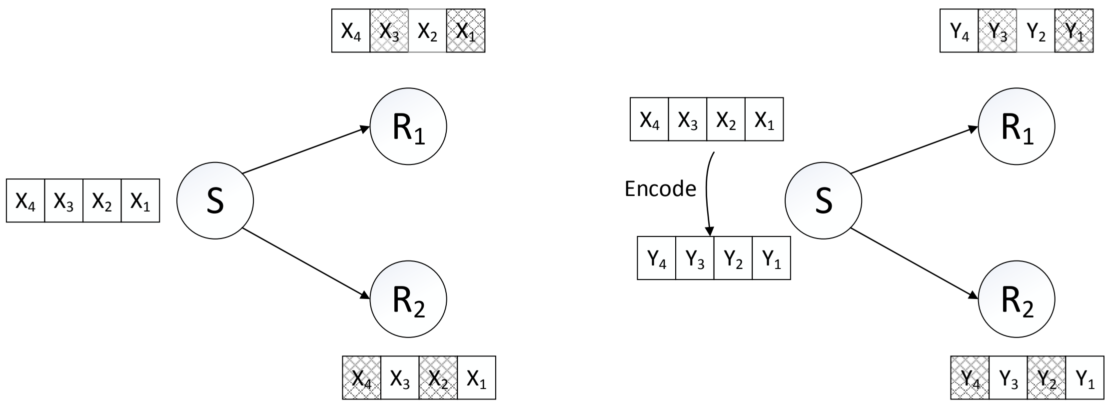
Figure 2.1: Network Coding for Unicast Data

packets.

A similar example for multicast transmission is given in Figure 2.2. In this situation the sender transmits four data packets to a pair of receivers using traditional store-and-forward routing. Suppose that, due to transmission errors, both receivers lose two packets each. As a result the sender must retransmit the same four packets again since it does not know which packets were lost. However, if in the same situation network coding is used, the sender needs only to retransmit two coded packets to the receiver nodes so they can recover the lost information. This is because the sender only needs to know the number of packets each receiver requires for packet decoding and not the specific identity of the packets lost.

Network coding in the application layer is the preferred method of implementation. This is because network coding is implemented on software level on top of existing network protocols. Therefore, there is no requirement to replace any network equipment with sophisticated routers to perform low level coding operations. Despite the fact that these higher level protocols do not give the same ideal performance as lower layer coding techniques, there are many promising applications in the literature for higher level network coding.

Peer-to-peer (P2P) file sharing is one area that can be improved by linear network coding. In [14], the authors developed and implemented a P2P file sharing algorithm based on network coding. In their algorithm, each node in the network collects linearly coded combinations of file fragments until the complete file is downloaded. This method contrasts



(a) Traditional Network (b) Network Coding

Figure 2.2: Network Coding for Multicast Data

traditional P2P sharing algorithms, like Bit Torrents [15], where peer nodes must download all file fragments until the file is complete. This is because in traditional P2P file sharing, un-coded file fragments are stored at each peer node. However, in network coded P2P file sharing systems the peer nodes store linear combinations of file fragments instead. As a consequence the downloader only needs to collect the required number of coded file fragments to complete the file. This provides the flexibility of distributed storage, adds more redundancy, and reduces the download time for each peer node. The authors also showed significant improvements in download time and tolerance for more dynamic network environments, such as seeders disconnecting before any peer node is able to download the complete file. Furthermore, some P2P streaming techniques based on network coding have shown that it can reach the theoretical maximum capacity of the network [16]. In general, file transfer protocols, based on network coding [17] [18], have shown significant improvement in reliability and transmission time over conventional protocols. Network coding has also been applied to distributed storage systems whose storages are unreliable [19] [20]. Moreover, network coding has even been applied to relatively new emerging areas such as molecular communications [21] in underwater acoustic networks [22].

In addition to the application layer, the transport layer of the network protocol stack is another area where network coding can be applied. The transport layer is responsible for reliable end-to-end delivery of transmitted data, as well as acknowledgements, and the retransmissions of lost packets. It is also responsible for flow and congestion control through its monitoring of the network traffic and the buffer capacities at network nodes. However the sending of acknowledgements for every data packet received at the receiver nodes is inefficient and consumes a considerable amount of network bandwidth. In the case of network coding applied at the transport layer, acknowledgements sent by receiver nodes do not contain information about which packets were received or not. Instead of acknowledging the receipt of individual packets, each destination node sends back requests for degrees of freedom that are required for decoding [23] [17] [18]. After receiving acknowledgements from the receivers, the source node sends another set of random linear combinations of packets to the receiver according to the required degrees of freedom requested in the acknowledgement packet. The advantage of this technique is that if some packets are lost in the communication link, the destination node is not required to request by the sequence numbers of the lost data packets, but instead request the missing number of degrees of freedom; in other words, the receiver requests the the missing number of linearly independent packets. Therefore, end-to-end data delivery using random linear network coding is more efficient in flow control, and uses less bandwidth than the traditional go-back-N protocol.

2.1 Network Coding Opportunities

Wireless networks are significantly different from wired networks due to the broadcast nature of the communication protocol. Opportunistic network coding exploits the broadcast nature of data packet transmission. Using the technique of opportunistic hearing of data packets in the transmission range, the overall throughput of the network can be increased significantly

[9]. Most network coding applications for wireless networks [24] [25], assume noiseless links with no interference due to careful scheduling, or that interfering packets are simply dropped. These assumptions simplify the wireless network coding problems for the sake of analyzing the most relevant issues.

We will now outline how a wireless architecture designed around network coding can help improve throughput. This improvement is possible because coding allows the routers to compress the transmitted information according to what packets are available at the receiver nodes. By matching what each neighbour node has with what another neighbour node requires, the network router or the access point (AP) can deliver multiple packets to different neighbours in a single transmission. This style of coding is called opportunistic network coding because coding is done on packets that can be overheard during the transmission by neighbour nodes and also from different flows.

The use of opportunistic network coding using omni-directional antennae was first investigated by Lun et al. in [26]. They analysed the minimization of communication cost and solved the problem in a distributed manner using linear programming. However, their work was mainly theoretical, and it was assumed that the network utilized multicast traffic. As a consequence, their technique has limited applicability to systems with unicast topologies. Consequently, authors such as Zhang et al. [27] and Sengupta et al. [28] have studied unicast-specific topologies. Their work demonstrated that, for the studied topologies, network coding results in better throughput than the pure forwarding of packets.

Chachulski et al. [29] presented a network coding based opportunistic routing technique that did not require centralized node coordination. They designed a practical system called MORE (MAC-independent Opportunistic Routing and Encoding), that applies network coding into the current network stack. Using this system they exploit the opportunistic overhearing, inherent to the wireless medium, in order to significantly increase the performance of the system. Using tests conducted on a wireless testbed, they showed that MORE provides both unicast and multicast traffic with significantly higher throughput than traditional routing.

Katti et al. [30] introduced MIXIT: The network meets the wireless channel, which is an architecture that performs opportunistic routing on groups of correctly received symbols. The core component of MIXIT is a novel symbol-level network coding system that also functions as an error correcting code. They utilized the physical layer of network coding to encode and decode the symbols. This allowed the network coding routers to identify the corrupted symbols and correct them before forwarding them to the next node. They addressed the main challenges in forwarding the coded packets as follows: First, the management of the packet buffers at each network node so that received symbols are stored correctly thus preventing duplicate transmission. Second, the maintenance of the error correcting mechanism. Despite the fact that network coding routers forward only the symbols that can be decoded, there is a probability that the forwarded symbols are corrupted. Therefore, symbol-level network coding co-functions as an error correcting mechanism by providing a redundant number of data packets. This redundancy allows for the correction of any corrupted symbols that pass through the network. MIXIT takes advantage of the broadcast nature and opportunistic overhearing characteristics of the wireless medium, and presents a merging technique that achieves both space and time diversity. It also forces the network and lower layers to enhance network throughput and the overall reliability of the communication. Ultimately, MIXIT conserves the distributed and unsophisticated nature of network coding necessary for practical implementation.

COPE was the first approach to implement opportunistic network coding for practical multi-hop wireless networks [9]. The authors explored the use of network coding for different network topologies, with an emphasis for two-way relay networks. COPE takes advantage of the broadcast nature of the wireless medium and performs opportunistic overhearing to reduce the number of packet transmissions. COPE also provides a general scheme for intersession wireless network coding by extending this topology to multicast traffic with multiple nodes. The authors also analyse the possibility applying their technique to other known topologies under dynamic network flows. The COPE architecture uses simple bitwise XOR operation to provide the advantages of network coding in wireless networks. These coding opportunities are affected by many factors, such as network topology, routing techniques, and network traffic patterns. Katti et al. [9] have shown that COPE cannot provide the same throughput-gain for TCP as UDP does. The main reason for this is explained by the connection-oriented nature of TCP. This can result in burst-like traffic behaviours with rate mismatch in different network edges. This is not the ideal situation to perform network coding as there is no waiting mechanism employed in COPE to tackle such situations.

To summarize, the key characteristics of the COPE coding scheme are as follows: 1) No scheduling technique exists in the scheme for synchronization. 2) Packets are not delayed; each time the node transmits a packet it occupies the head of the line (HoL) position in the queue. When multiple packets are available the access point (AP) will code them together, but if not they will be transmitted un-coded. 3) Packet reordering is not done at the nodes, and the AP transmits the packets in the same order in which they arrived at the FIFO queue.

Since COPE lacks both session and link scheduling algorithms, Cuit et al. [31] proposed a fully distributed, suboptimal scheduling algorithm, called Coding with Opportunistic Reception (COPR), as an extension to COPE. They showed that COPR can reduce routing power by exploiting multiple-reception gain in network coding. Therefore, COPR significantly improves the overall network performance when compared to COPE. Their extensions to COPE can be summarized as 1) The utilization of intersession network coding to both multicast and unicast sessions. 2) The use of an achievable rate region for single-hop wireless network coding using an interference model. 3) The use of a back pressure algorithm to do dynamic scheduling and single-hop wireless network coding. This back pressure-based scheduling system is then combined with the fixed path routing presented in COPE. 4) The reduction of the acknowledgement overhead.

2.2 Opportunistic Network Coding in Wireless Two Way Relay

The opportunistic network coding over two-way relay networks became very popular after its first introduction, as a part of the COPE protocol, since it offered significant performance improvements over traditional store and forward packet routing [9] [32].

Data packets going in opposite directions through a wireless AP are coded together in order to reduce the amount of spectrum access [9]. This reduction of wireless frequency spectrum usage also reduces the transmission cost. In the early stages of network coding, in order to achieve maximum network capacity through network coding it was assumed that packets of different flows were well synchronised. Both Chen et al. [3] and Larsson et al. [33] focused on the assumption that waiting queues at the AP are saturated (i.e., they focused on removing the assumption that the AP always has packets from both connections available). By doing so, they created a situation where the relay could choose to either transmit an un-coded packet, or wait for a coding opportunity with a newly arrived packet. However, network coding can be applied only when the wireless AP has received at least one data packet from each of the wireless nodes. When the AP receives a data packet from only one node but not from the other, the AP can either transmit the received packet without coding, or wait for a packet to be received from the other node. The former strategy leads to no delay but also reduces the spectral efficiency, while the latter strategy incurs a coding delay but improves the spectral efficiency as packets from both nodes are transmitted simultaneously. Using network coding as the only mechanism for removing packets from the AP can cause network buffer instability. This instability is caused when there is an arrival rate mismatch between the two nodes. The node with less frequent data packet arrivals will send packets to the AP at a lower rate on average. As a consequence, there will be insufficient opportunities to serve the packets of the node having the higher frequency of data packet arrivals. Furthermore, this instability occurs even when the two flows have the same rate due to the probabilistic nature of packet arrivals. Dealing with this stochastic arrival of data packets to the AP has gained special interest in opportunistic network coding, especially for cases where there are no packets available at one of the nodes.

In opportunistic network coding, the AP can delay packet transmission, for a pre-specified amount of time, before transmitting an un-coded packet. If this waiting time is too long, then packet transmissions will be delayed significantly and the system performance will degrade. However, if the AP does not wait long enough it will have to transmit many data packets un-coded before it has an opportunity to pair two packets for network coding, thus diminishing the advantage of network coding. For each delay of packet transmission there is an associated waiting cost, and similarly, for each un-coded transmission there is an associated transmission cost due to the loss of spectral efficiency. Therefore there comes about a trade-off between the waiting cost and the transmission cost.

The waiting vs transmission tradeoff with plain routing has been studied for a single wireless link as well as multiuser scenarios [34] [35] [36]. The stability region for two-hop bi-directional communications between a pair of nodes with stochastic flows is further studied in [37] and for other network topologies in [8] [38]. The main focus of these works is mainly queue stability but looks at the maximum achievable throughput without optimizing the wireless transmission costs.

There have been a number of works studying the interaction of network coding with stochastically-varying traffic in two-way relay networks. The authors in [39] present a simulated scheduling technique that minimizes the queueing delays in opportunistic network coding for both uniform and random packet arrival models. For the case of unequal arrival rates, they propose the use of a waiting threshold at the network node with the fastest arrival rate. Due to the probabilistic nature of the packet arrivals, having a waiting a threshold for only one node can cause some packets to be delayed too long. Furthermore, ignoring the effects caused by not having a threshold for one data stream may result in suboptimal system performance. For this reason, a deadline-aware waiting technique was proposed in [40], for broadcast network-coded data, under the assumption that the nodes generate packets at a constant rate. Their technique focused mainly on reducing the transmission cost within the limits of given deadlines for the packet delivery, but did not address the issue of degrading the quality of service within the deadline. Techniques for optimization of the trade-off between transmission and coding costs have been analysed in [41] for a linear tandem network using network coding. They calculated the probabilities for not having packets for coding, but did not allow for the delaying of packets to wait for a coding opportunity. The authors in both [42] and [3] analyzed opportunistic network coding with geometric arrivals. They proposed the transmission of packets waiting for a network coding opportunity with some probability that is proportional to the data packet arrival rate. However, the main drawback of their approach was that it requires a system with an infinite memory, making it impractical. Furthermore, their approach can delay some packets for an extended amount of time as it does not utilize any waiting limit for the probabilistic transmissions. Umehara et al. [43] proposed coded packet priority access (CPPA) protocol in which coded packets have higher transmission opportunity than uncoded packets at the relay node. In [44] the authors extend this proposed method to model a network coding aware router.

A more comprehensive analysis for a waiting time based time-out mechanism was presented in [45] for avoiding excessively long waiting times at the relay, as in [44]. They also considered a two-way relay capable of network coding that receives packets from two different flows. When a packet from one flow arrives at the relay and find a packet from the other flow waiting there, the new packet is coded with the HoL packet from the other flow, and transmitted. If the arriving packet finds only packets from the same flow (or no packets at all) waiting, the packet combines the queue of the waiting packets according to a

FIFO policy. Network coding operations and packet removal from each queue are assumed to occur instantaneously, so waiting packets necessarily belong to the same flow. As soon as a waiting packet becomes the HoL packet, a timeout period is assigned to it which acts as the maximum additional sojourn time of this packet at the relay. If a packet from the other flow arrives during the timeout period, the network coding mechanism is exercised, and the waiting packet is removed from the queue. If not, a timeout occurs and the waiting packet is immediately removed from the queue and is transmitted. Timeout periods, assigned to different packets in the same flow are independently and identically distributed (I.I.D.) random variables from a distribution that is tailored to flow-related parameters. Their system model is simplified to include the factors strictly relevant to network coding and its trade-offs (e.g. the details of the final packet transmission stage are not included). The simplified model provides a tractable analysis, leading to simple closed-form expressions. These expressions illustrate how, by changing the timeout parameters, the performance and efficiency of the system is affected. The result of these analyses are subsequently employed in the selection of the timeout parameters for the traffic load present at the relay. The selection of the timeout parameters are done in such way that achieves the highest possible network coding efficiency subject to the given delay-related QoS requirements. Two delay control approaches are considered: 1) Separate timeout parameters in the two flows that can be independently tuned. 2) A simple suboptimal approach with common timeout parameters for both flows. Some closed-form results have been obtained to support the concept of delaying packets. In the presence of asymmetric traffic, they eliminate the timeout mechanism on the packets of the slower flow. By doing so, they still maintain the stable operation and the maximum network coding gain possible for this traffic pattern. Even though this approach provides some improvement over COPE, their waiting threshold is suboptimal, and thus cannot guarantee the optimal performance.

In [46] a queueing model is developed in which the decision to transmit an uncoded packet may depend on the number of packets in each queue. They derive the diversity order of several bi-directional protocols such as Direct Transmission (DT), Multiple Access BroadCast (MABC), Time Division BroadCast (TDBC) and Hybrid BroadCast (HBC). The impact of these decisions on the transmission cost vs delay trade-off is analyzed in [47]. Similarly, the scenario where packets are always transmitted un-coded if no opportunity arises is studied in [48] and [49].

Liu et al. in [48] characterize the achievable rate regions for the four-slot multi-hopping mechanism (FSMH), and the two-way opportunistic network coding scheme. For a given Poisson traffic pattern, they analyse the optimal end-to-end sum rates for both the FSMH and the two-way opportunistic network coding scheme. They first, focus on only maximizing the sum rate in one-way transmissions. They also show that, regardless of the transmission method used, the maximum network coding gain is always achieved when two-way traffic is symmetric. Consequently, the proposed opportunistic network coding scheduling algorithm is able to achieve a stable system without knowing the actual Poisson arrival rates for either the FSHM or the opportunistic network coding scheme.

Game theoretic models for network coding presented in [50], [51]. The main focus of their work is flow optimization, rather than the optimization of transmission and waiting costs. Ciftcioglu et al. in [52] present two threshold-based waiting policies: a centralized policy, and a distributed policy, where each source tries to optimize its own cost-delay trade-off. In [53], the authors consider the probabilistic transmission for all packets waiting for coding. In their method, the transmission probabilities are dependent on whether the packets are coded or not, and on the destination of the packets. Khreishah et al. [54] analyze the upper-bound of the capacity region for two-way relay network coding. The analog network coding (ANC) schemes for asynchronous two-way relay network is presented in [55] [56].

In [57], the authors propose a method based on Markov decision process (MDP) to find the waiting times that optimizes the total cost. They were the first to prove that the optimal waiting policy for this problem is to have a threshold-based policy for packets waiting in each queue for networks with Geometric packet arrivals.

2.3 Related Queueing Theoretic Analysis of Opportunistic Network Coding

The delaying of packets indefinitely in order to achieve network coding can lead to system instability. This is because the buffers which are attached to the relays with high data rates may have memory overflow issues. Therefore, in this context, the use of queueing theory to analyse the performance of network coding becomes very important. Queueing analysis can be used to mathematically analyse the delay performance of network coding and help to make the optimal decision for efficient system performance.

The authors in [49] analyze queueing systems with specific service processes which capture the queueing behavior of two-way relay network coding. They show that network coding can improve network throughput significantly if it is used opportunistically or implemented in a fully-synchronized network setting. In [58] a queueing analysis is presented to maintain the physical queue size in order to track the backlog of the degrees of freedom required at the receiver node. They minimize the buffer size required in network-coded file transfer protocols which are originally presented in [17] [18]. Sagduyu et al. [41] consider the problem of network coding in wireless queueing networks with tandem topology, and also consider multi-hop packet propagation. Their model is a discrete-time-slotted and synchronous system in which the transmission time of each packet is one time slot.

Yuan et al. [59] present a queueing model for asynchronous network coding at a single router. They assume traffic arrivals to be I.I.D. Their coding node maintains a separate FIFO queue, with a queue size K for each flow, which tries to increase the coding opportunities at the coding node by delaying packets with a maximum opportunistic delay threshold.

The opportunistic network coding presented in [3] uses queueing analysis to make decisions of coding dependent on the size of the buffer at a given node. In their queuing model, the decision to transmit un-coded packets depends on the number of packets in the buffers. The authors propose a queueing model with Bernoulli symmetric arrival processes where the relay node maintains a queue to store received packets from each source. The relay node performs network coding when both queues are non-empty. Otherwise, it sends the packets un-coded with a probability that depends on the state of the queues. They use queueing analysis to optimise this probability in order to minimize packet delays for a given a power constraint. Ding et al. in [60], extend this work to model asymmetric arrival processes as well. Using a Finite State Markov Chain they characterize the relay queue states, derive the transition probabilities and the stationary distribution to analyze the average power and delay costs. They then find the optimal power-delay trade off, and propose a heuristic discrete solution for the problem. The impact of this decision on the energy-delay trade-off is then analysed in [47]. In this work the authors present energy-efficient transmission decisions based on the state of queue buffers. By doing so they maintain stable queues at the nodes, thereby providing a cross-layer optimization trade-off between different measures of throughput and energy efficiency.

Chieochan et al. [61] propose a discrete time Markov chain (DTMC) model for a wireless, lossy, butterfly network, where a opportunistic network coding scheduling protocol [62] is employed at the relay node. They use the queueing model to solve the synchronization problem using dynamic buffer allocation at the relay station. In [4], the authors use a Continuous-Time Markov Chain (CTMC) with exponential inter-arrival times, to model the energy-delay trade-off problem. They use the queueing model to determine the buffer size and decide to transmit data packets un-coded when the buffer is full. When the length of the waiting queue exceeds a threshold, the packets are transmitted uncoded. They assume the relay node has independent Poisson arrivals and show that network coding, with no limiting measures, results in infinite delays and propose a technique that is a hybrid between network coding and traditional routing. In [63], they consider the same scenario, but develop different priority-based queues in the relay nodes buffer for different flows. Their opportunistic network coding scheme is based on traffic priority and queue length. However, non of the above queueing models provide any closed-form expressions for the optimization of the transmission-versus-waiting costs.

In our work, presented in [64], we propose an analytical method using a DTMC to model the system behaviour of an AP using network coding under packet arrivals according to a discrete time Markov arrival process (DMAP). We presented a threshold for the maximum waiting times of data packets in each queue to minimize the packet delays. We derived this by extending the work in [65] and the pair formation technique presented by Neuts and Alfa [66]. We showed how the variation of waiting time affects the amount of spectrum access and the average waiting time of packets. Based on what we formulated, an efficient waiting strategy to minimize both transmission and waiting costs is presented. Therefore we show how the delaying of packets to increase coding opportunities is more beneficial to the queue of the lower arrival rate. We also found the optimal waiting times thresholds which minimize both the number of transmissions and the average waiting time of packets in the buffer.

2.4 General Arrivals and Use of Phase Type Distribution

One disadvantage of Geometric (Poisson in continuous time) arrivals is that it forces the assumption that each time slot has the same arrival probability due to its lack of memory property. Therefore, it cannot be used to model most correlated inter-arrival processes as shown in [67]. The data packets in telecommunication systems can be more accurately modelled by correlated inter-arrival processes using Phase-type distributions [11] and most general arrival functions can be represented using such a distribution. This allows us to analyse the correlated inter-arrival times of packets having general arrival distributions.

Using phase-type distribution to model the arrival process is advantageous because most general arrival distributions can be represented (or closely approximated) by a phase type distribution [68]. Rather than assuming geometric distribution (or Poisson distribution for the continuous-time case), the data arrivals can be modelled using phase type (PH) distribution with parameters (α, T) of order n, where α and T are a row vector and a square matrix of order n, respectively [69]. Every element of T has the following property $0 \leq T_{ij} \leq 1$ and $T\mathbf{1} \leq \mathbf{1}$ with at least one of the rows strictly less than 1, where $\mathbf{1}$ is a column vector of ones of appropriate dimension. Also, the vector α is stochastic, i.e., $\alpha\mathbf{1} = 1$. We define $p_b(k)$ as the probability that there is an arrival after k time slots. Then, according to the PH distribution, we have,

$$p_b(k) = \alpha T^{(k-1)} t \tag{2.1}$$

where $t = \mathbf{1} - T\mathbf{1}$.

To model data arrivals as a phase type distribution and we need to find the exact parameters for matrix T. We can use the Expected Maximization (EM) algorithm to find the exact elements in transition matrices as presented in [68]. The EM algorithm used in [70] is an iterative method for implementing a Maximum Likehood estimation in the case of incomplete data. The complete iterative steps of the EM algorithm are presented in [68]. By following these iterative steps, we can calculate the T matrix of phase-type distributions for a given data set of an arrival pattern. While the size of the matrix is not restricted in the original algorithm, they showed that using a larger number of states provides a more accurate data-fitting for the phase-type distribution. Using numerical analyses, they showed that a dimension of six is a sufficient size to represent the phase-type distribution of data in a telecommunication system.

Therefore the phase type distribution is a very accurate tool to model general packet arrival functions in telecommunications traffic and we use it to model all the data arrivals in the following chapters and analyse the optimal waiting policy for opportunistic network coding.

Chapter 3

Determining the Optimal Waiting Policy

In this chapter, we investigate the problem of finding the optimal waiting policy for data packets in two-way relay opportunistic network coding under phase type arrivals. We first model the trade-off between waiting and transmitting packets, using a Markov decision process (MDP) to analyze the problem and to find an optimal waiting policy that gives us a minimum average cost in the long term (as the system epoch tends to infinity). By using Sennott's theorem [71] [72] for MDP optimization, we prove the existence of an average-cost optimal policy within our MDP model. We further prove that the problem is convex under sufficient conditions, thus allowing us to find the global optimal solution using the Gradient Descent Algorithm (GDA). Using the steps of value iterations algorithm for the MDP, we prove that the nature of the global optimal solution is a threshold-based policy. This threshold-based policy defines a separate threshold for each phase vector of phase-type arrival functions within each queue. In some cases the current phase of the arrival function may be unknown. For such situations we introduce a technique to probabilistically predict the arrival phase vector in order to determine a threshold policy.

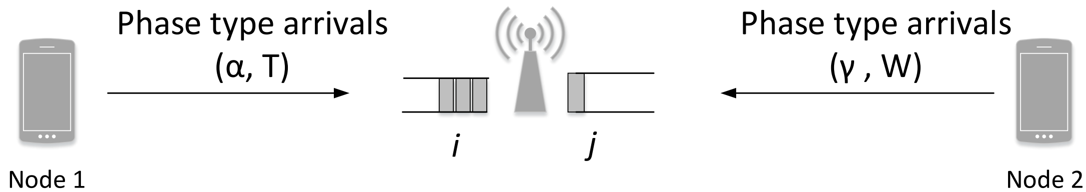


Figure 3.1: Two Way Relay Access Point with Phase Type Packet Arrivals

3.1 System Model and Problem Statement

We consider network coding in a two-way relay AP as shown in Figure 3.1. Node 1 and Node 2 each generates packets and transmits them to the AP according to discrete time phase-type distributions defined by (α, T), having n_1 number of phases, and (γ, W) having n_2 number of phases, respectively. The packets are assumed to have the same length. The whole system shares a synchronized clock to define the time unit i.e. the packet arrivals from each node happen at the same time during a time slot. The access point uses the same channel to transmit packets to each node. The AP receives data packets from both nodes, computes the bit-wise XOR addition of the packets received from each node, and then broadcasts the coded packet to both nodes. This saves the transmission cost by one unit in comparison to traditional un-coded communication (two separate transmissions). This transmission cost savings represents a 50% savings in transmission spectrum access and a 25% savings in the overall spectrum access for this communication. Furthermore, the saved downlink spectrum can also be used to send more data and thereby increase the overall throughput of the system. However, since the two wireless nodes are not synchronized and the packet transmissions from the nodes are not regular, it is possible that when the AP receives a packet from one node, the waiting queue for the other node is empty. In this case the packets in the non-empty queue wait for packets to arrive at the other queue. The AP tries to reduce the number of packet transmissions by using network coding. The AP buffers packets coming from each node into separate queues to facilitate network coding. As long as both queues contain one or more data packets, the relay node will code one packet from each queue and transmit them as a single packet. However, when a packet cannot be paired for coding, the AP can choose to buffer the packet in the queue or transmit without coding. However, in order to for the AP to make this choice it must abide by an optimal policy that yields the best choice at any instance in time. To find this policy we first analyze the trade-off between the costs associated with waiting, and the cost associated with transmitting packets un-coded. To simplify our analysis, we assume that the capacity of the transmission links are large enough so that system is stable and the queues are finite. So long as there are one or more packets in each queue to be paired, network coding can be performed without any delay. If queue 1 has i number of packets and queue 2 has j $(i \neq j)$ number of packets, then there are $|i - j|$ number of extra packets that will not immediately be coded.

3.2 Markov Decision Process for the System

We develop a MDP [73] to analyze the trade-off between buffering vs transmitting packets in this system. An MDP is a sequential decision making process (Figure 3.2). In an MDP a system is defined by the state of the system at any given time. At any given state we can choose an action to take. In our case the state is defined by the number of packets waiting at each queue, and the action we take is how many packets to transmit, both coded and un-coded. For any action taken there will be an associated immediate cost. In our case, the immediate cost is the cost of transmitting packets summed with the cost of buffering the remaining packets until the next time unit. Depending on the action we take, and the probability of packet arrivals for a given time unit, the system may move to a different state with a new set of possible actions. The MDP is composed of the following factors: The set of possible states, the set of possible actions, the set of immediate cost for each action in a given state, and the set of transition probabilities between states. Furthermore, we assume

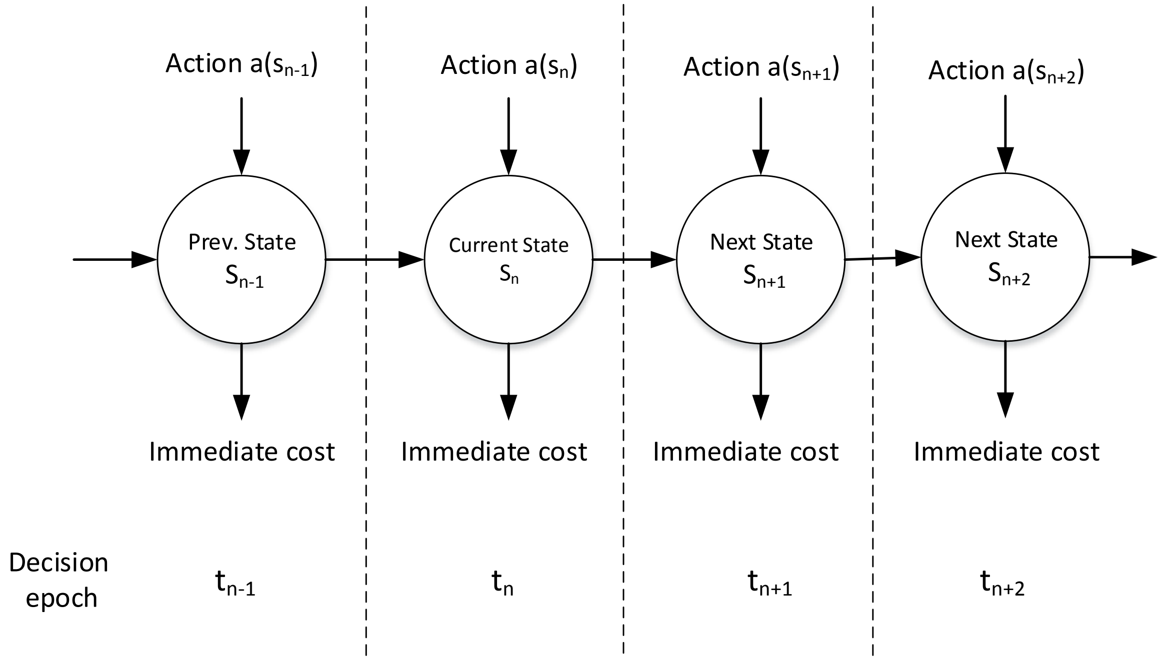


Figure 3.2: Markov Decision Process

that all necessary information is available to us in order to make the best decision at a given state. For every decision there will be an associated immediate cost, and a transition to a new state. The immediate cost we pay and the transition probabilities depend on what action was taken, and the state we were in prior to the decision. As the system moves on in time, we take a set of actions and pay a sequence of immediate costs. Our objective is to find the optimal set of actions that minimizes the sum of the immediate costs paid in the long-term.

A decision policy is a technique that provides us with a given set of actions by predicting the possible future states we will progress through. Any given policy is associated with a fixed set of immediate costs. Furthermore, the sequence of actions associated with a given policy are predetermined prior to the initialization of the system. Therefore, it is known beforehand what expected total long-term cost for a given policy will be. Moreover, the set of actions defined by a given policy is determined by the set of immediate costs, and the set of transition probabilities between states. Finally, a policy is independent of all previous costs paid, all previous actions taken and any state we previously occupied.

The state space of the MDP is defined by $(i, j, e) \in S$. Here i is the number of packets waiting in queue 1, j packets waiting in queue 2, $e = (b, d) \in E$ is the current arrival phase vector where, b $(0 \leq b \leq n_1 - 1)$ is the arrival phase of queue 1 and d $(0 \leq d \leq n_2 - 1)$ is the current arrival phase of queue 2. Here $E = \{(0,0), (0,1), (0,2), \cdots (n_1 - 1, n_2 - 1)\}$ is the state space of the phase vector. At state $s = (i, j, e)$, the system will take some action $a(s)$, where $a(s)$ is defined by the total number of packets being transmitted (both coded and un-coded). For i and j packets waiting in queue 1 and queue 2 respectively, the action space can be defined as $A = \{\min(i, j), \cdots, \max(i, j)\}$. Let c_t define the cost incurred when transmitting a packet and let c_h define the holding cost of a packet for one time unit. Since our model is a discrete time system, we need to define the order of events happening during one time slot because two events cannot occur simultaneously in a discrete time system. Therefore, we assume that a transmission action is taken at the beginning of a time unit and packet arrivals occur only after this action is taken. Note that the order of events does not affect the long term performance of the system. When the action $a(s)$ is taken at the state $s = (i, j, e)$ the total cost for that time unit is the cost of $a(s)$ packets being transmited, plus the cost of buffering the remaining packets, plus the cost of buffering the newly arrived packets until the next time unit, as shown in (3.1),

$$c(i, j, e, a(s)) = c_t a(s) + c_h((i - a(s))^+ + (j - a(s))^+) + c_h(t_b + w_d) \tag{3.1}$$

where $(x)^+ = \max(x, 0)$, t_b is the bth element in t $(t = \mathbf{1} - T\mathbf{1})$ and w_d is the dth element in w $(w = \mathbf{1} - W\mathbf{1})$ and $s = (i, j, e)$.

Our objective is to find a transmission policy that provides the optimal action at every state s in order to minimize the long-term average cost. It is important to note that the action produced by transmission policy may not be optimal in the short-term due to the probabilistic nature of the packet arrivals. However, the long-term average cost under this optimal transmission policy θ is defined as,

$$g(\theta) = \lim_{N\to\infty} N^{-1} E_{\theta}[\sum_{n=0}^{N} c(i_n, j_n, e_n, a(s))|(i_s, j_s, e_s)] \tag{3.2}$$

where (i_s, j_s, e_s) is the random state at the nth time unit. We start the system at state (i_s, j_s, e_s). The probabilities of the initial arrival phases in e_s are defined by $\alpha = [\alpha_1, \alpha_2, ..., \alpha_k, ..., \alpha_{n_1}]$ and $\gamma = [\gamma_1, \gamma_2, ..., \gamma_k, ...\gamma_{n_2}]$ for queue 1 and queue 2 respectively. Our objective is to find the optimal policy that minimizes the average cost $g(\theta)$.

We assume that the equivalent present cost in a MDP is discounted when compared to the future cost due to the time-value of cost [73]. Therefore we introduce a discount factor β $(0 \leq \beta \leq 1)$ per time unit. Thus, the total expected β discounted cost under policy θ is given by,

$$v_{\beta,\theta}(i, j, e) = E_{\theta}[\sum_{n=0}^{\infty} \beta^n c(i_n, j_n, e_n)|(i_0, j_o, e_0) = (i, j, e)] \tag{3.3}$$

The optimal discounted cost at the state (i, j, e) is given by $v_{\beta}(i, j, e)$.

3.3 Proof of the Optimal Threshold Policy

Our objective is to find a policy that minimizes the expected average cost per unit time. The number of steps for finding the optimal policy are three fold. First, we prove that the discounted cost function in our MDP has an average cost optimal policy. Second, we prove it's convexity so that we can find the solution using a GDA. The third step is to show that the optimal policy for minimizing the expected average cost is a threshold based policy dependent on the current arrival phase vector.

Using Sennott's theorem [71] [72] for the parameters given in our MDP, we derive Theorem 1.

Theorem 1

(a) There exists a finite constant $g = \lim_{\beta \uparrow 1}(1-\beta)v_\beta(i,j,e)$ for every state $s = (i,j,e) \in S$. (Note: The physical meaning is that the constant g is less than or equal to the worst case of $g(\theta)$ [74] [71]).

(b) There exists a limit function $h(i,j,e)$ satisfying the average cost optimality inequality,

$$\begin{aligned} g + h(i,j,e) \geq & \min_{a(s) \in A_{i,j,e}} \{c(i,j,e,a(s)) \\ & + \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e_n,e_{next}} v_\beta((i-a(s))^+ + k, (j-a(s))^+ + l, e_{next}))\} \end{aligned} \tag{3.4}$$

(c) Let θ^* be the stationary policy realizing the minimum of (3.3), then θ^* is an average cost optimal stationary policy that is a limit point of a sequence of discounted cost optimal stationary policies $\{\theta_{\beta_k}\}_{k \geq 1}$ where $\beta_k \uparrow 1$ (one sided limit of β_k approaches from the left).

Sennott's theorem given in section 3.4 shows that there exists a discounted average cost optimal policy for an MDP if Proposition 1, Proposition 2 and Proposition 3, stated in below, are satisfied. We prove that our MDP discounted cost function given in 3.3 satisfies these three propositions for the existence of a minimum average cost optimal policy.

Proposition 1

The optimal discounted cost function $v_\beta(i,j,e)$ is finite for every state $(i,j,e) \in S$ and β.

Proof of Proposition 1

To prove this we consider a policy θ where every packet in each queue is transmitted at every state. The cost under this policy is then defined as,

$$v_{\beta,\theta}(i,j,e) = c_t \max(i,j) + E[\sum_{n=1}^{\infty} \beta^n c_t \max(A_n^1 + A_n^2)] \tag{3.5}$$

where A_n^1 and A_n^2 are the number of packet arrivals to the queue 1 and queue 2, respectively, between $(n-1)$th and nth transmission opportunities. The optimal discounted cost $v_\beta(i,j,e)$ is the lower bound for the discounted cost $v_{\beta,\theta}(i,j,e)$ under policy θ. Therefore this policy satisfies the following inequality:

$$\begin{aligned}
v_\beta(i,j,e) &\leq v_{\beta,\theta}(i,j,e) \\
&\leq c_t \max(i,j) + E[\sum_{n=1}^{\infty} \beta^n c_t (A_n^1 + A_n^2)] \\
&= c_t \max(i,j) + \beta \alpha t + \beta^2 \alpha T t + \beta^3 \alpha T^2 t + \cdots + \beta \gamma w + \beta^2 \gamma W w + \beta^3 \gamma W^2 w + \cdots \\
&= c_t \max(i,j) + \beta(\alpha (I - \beta T)^{-1} \cdot \mathbf{1} + \gamma (I - \gamma W)^{-1} \cdot \mathbf{1}) < \infty
\end{aligned} \tag{3.6}$$

Therefore, for any initial state (i,j,e), the optimal discounted cost $v_\beta(i,j,e) < \infty$ for $0 \leq i < \infty$ and $j \leq \infty$ and $0 < \beta \leq 1$.

Proposition 1 implies that the optimal discounted cost function $v_\beta(i,j,e)$ for $(i,j,e) \in S$, satisfies the discounted cost optimality equation 3.7 [73],

$$v_\beta(i,j,e) = \min_{a(s) \in A_{i,j,e}} \{c(i,j,e,a(s)) + \beta \sum_{k,l} p_{e,e_{next}}^{k,1} p_{e,e_{next}}^{l,2} v_\beta((i - a(s))^+ + k, (j - a(s))^+ + l, e_{next})\} \tag{3.7}$$

where $p_{e,e_{next}}^{k,m}$ is the probability of arriving $k \in \{0,1\}$ packets to the queue $m \in \{1,2\}$, with the arrival phase transition from the phase vector e to e_{next}.

Here $p_{e,e_{next}}^{0,1}$, $p_{e,e_{next}}^{1,1}$, $p_{e,e_{next}}^{0,2}$ and $p_{e,e_{next}}^{1,2}$ are defined by the elements in T, αt, W, and γw respectively.

A stationary policy that satisfies equation (3.7) will be a discounted cost optimal policy.

Our objective is to find an optimal policy that gives the minimum long-term average cost. We show that there exists a long-term average cost stationary policy within our MDP and that it is the limit of the discounted cost optimal policy. □

Proposition 2

There exists a finite non-negative function M such that $h_\beta(i,j,e) \geq -M$ for $(i,j,e) \in S$ and $\beta \in (0,1)$.

Proof of Proposition 2

We prove this by showing that there exists a subset of phase vectors $E'(i,j,s) \subset E$, for each state $s = (i,j,e) \in S$, such that the discounted cost function $v_\beta(i,j,e)$ is non-decreasing in i and j, with a phase transition to the next phase vector $e' \in E'(i,j,s)$.

We therefore prove this by mathematical induction using the following steps given in the Value Iteration Algorithm [73]:

$$\begin{aligned} v_{\beta,n}(i,j,e_n) = \min_{a \in A_{i,j,e_n}} &\{c(i,j,e_n,a(s)) \\ &+ \beta \sum_{k,l,e_{n-1}} p^{k,1}_{e_n,e_{n-1}} p^{l,2}_{e_n,e_{n-1}} v_{\beta,n-1}((i-a(s))^+ + k, (j-a(s))^+ + l, e_{n-1}))\}, \\ &(i,j,e_n) \in S, e_{n-1} \in E \end{aligned} \tag{3.8}$$

The starting value when $n = 0$ is $v_{\beta,0}(i,j,e_0) = 0$, for every state (i,j,e_0). Therefore, $v_{\beta,n}(i,j,e_n)$ is non-deceasing in i and j for every phase vector $e_n \in E$ for $n = 0$.

Let $v_{\beta,n-1}(i,j,e_{n-1})$ be non-deceasing in the $n-1$ iteration, and select an optimal action $a \in A_{i+1,j,e_n}$, from (3.8), such that,

$$
\begin{aligned}
v_{\beta,n}(i+1,j,e_n) &= c(i+1,j,e_n,a(s)) \\
&\quad + \beta \sum_{k,l,e_{n-1}} p^{k,1}_{(e_n,e_{n-1})} p^{l,2}_{(e_n,e_{n-1})} v_{\beta,n-1}((i+1-a(s))^+ + k,(j-a(s))^+ + l,e_{n-1}))
\end{aligned} \tag{3.9}
$$

By taking the same action, $a(s)$, for the state (i,j,e'_n), we obtain the following expression, as derived from (3.8):

$$
\begin{aligned}
v_{\beta,n}(i,j,e'_n) &\le c(i,j,e'_n,a(s)) \\
&\quad + \beta \sum_{k,l,e_{n-1}} p^{k,1}_{(e'_n,e_{n-1})} p^{l,2}_{(e'_n,e_{n-1})} v_{\beta,n-1}((i+1-a(s))^+ + k,(j-a(s))^+ + l,e_{n-1})) \\
&\qquad e'_n \in E' \subset E
\end{aligned} \tag{3.10}
$$

Since equation (3.10) is still valid when $a(s) = i+1 > j$, $a(s) \neq A_{i,j,e_n}$, we conclude that our initial assumption, $v_{\beta,n-1}(i,j,e'_n)$ is non-deceasing in i. We then investigate the case where $e_n = e'_n$, and resolve that $c(i,j,e_n,a(s))$ is also non-deceasing in i. Therefore, using the equation in (3.1), we derive the following inequality from (3.9) and (3.10):

$$
\begin{aligned}
v_{\beta,n}(i+1,j,e_n) - v_{\beta,n}(i,j,e_n) &\ge c(i+1,j,e_n,a(s)) - c(i,j,e_n,a(s)) \\
&\quad + \beta \sum_{k,l,e_n} p^{k,1}_{(e_n,e_{n-1})} p^{l,2}_{(e_n,e_{n-1})} \{ v_{\beta,n-1}((i+1-a(s))^+ + k,(j-a(s))^+ + l,e_{n-1})) \\
&\qquad - v_{\beta,n-1}((i+1-a(s))^+ + k,(j-a(s))^+ + l,e_{n-1}))\}
\end{aligned} \tag{3.11}
$$

Therefore, given that $e'_n \in E' \subset E$

$$
\begin{aligned}
v_{\beta,n}(i+1,j,e_n) - v_{\beta,n}(i,j,e'_n) \geq c(i+1,j,e_n,a(s)) - c(i,j,e'_n,a(s)) \\
+ \beta \sum_{k,l,e_n} p^{k,1}_{(e_n,e_{n-1})} p^{l,2}_{(e_n,e_{n-1})} v_{\beta,n-1}((i+1-a(s))^+ + k, (j-a(s))^+ + l, e_{n-1})) \\
- \beta \sum_{k,l,e'_n} p^{k,1}_{(e'_n,e_{n-1})} p^{l,2}_{(e'_n,e_{n-1})} v_{\beta,n-1}((i+1-a(s))^+ + k, (j-a(s))^+ + l, e_{n-1}))
\end{aligned} \tag{3.12}
$$

Therefore, there exists a subset of phase vectors $E' \subset E$ for each phase (i,j,e) in each state $s = (i,j,e) \in S$, such that the discounted cost function $v_\beta(i,j,e)$ is non-decreasing in i for fixed j with a phase transition to the phase vector $e' \in E'(i,j,s)$.

We can similarly prove the above for j with fixed i. Therefore, we conclude that there exists a subset of phase vectors $E'(i,j,s) \subset E$ for each state $s = (i,j,e) \in S$, such that the discounted cost function $v_\beta(i,j,e)$ is non-decreasing in i and j.

It is important to note that for every action taken in each state, it is our expectation that we will go in to a state which will result in the minimum cost. This shows that in the worst case there exists at least one phase vector for the next state that will make the cost function non-decreasing in i and j. $\square$

Proposition 3

The MPD has a stationary policy that induces an irreducible and ergodic Markov chain with a finite average cost.

Proof of Proposition 3

To prove this, we consider a policy θ where every packet in each queue is transmitted at every state. Under this policy the state transition probability is defined by:

$$p_{(i,j,e_x)(k,l,e_y)} = p^{k,1}_{(e_x,e_y)} p^{l,2}_{(e_x,e_y)} \tag{3.13}$$

Where $p^{1,1}_{e_x,e_y}$, $p^{0,1}_{e_x,e_y}$, $p^{1,2}_{e_x,e_y}$ and $p^{0,2}_{e_x,e_y}$ are represented by $t\alpha$, T, $w\gamma$, and W.

We then derive the transition matrix P for this Markov chain $p_{(i,j,e_x)(k,l,e_y)}$ by showing that each level (i,j) of the matrix P represents the transition probabilities of the number of packets in each queue:

$$P = \begin{array}{c} \\ (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (0,1) & (1,0) & (1,1) \end{array} \\ \begin{pmatrix} T \otimes W & T \otimes w\gamma & t\alpha \otimes W & t\alpha \otimes w\gamma \\ T \otimes W & T \otimes w\gamma & t\alpha \otimes W & t\alpha \otimes w\gamma \\ T \otimes W & T \otimes w\gamma & t\alpha \otimes W & t\alpha \otimes w\gamma \\ T \otimes W & T \otimes w\gamma & t\alpha \otimes W & t\alpha \otimes w\gamma \end{pmatrix} \end{array} \tag{3.14}$$

Therefore, the Markov chain $p_{(i,j,e_x)(k,l,e_y)}$ is irreducible and ergodic.

Given that the stationary vector of this Markov chain is $\pi_{i,j,e}$, the transition matrix P is stochastic, and that the relationship is true $\sum_{i,j,e} \pi_{i,j,e} = 1$. Then the long-term average cost of this Markov chain is given by,

$$\begin{aligned} g(\theta) &= \sum_{(i,j,e)\in S} \pi_{(i,j,e)} \max(i,j) \\ &= c_t(\sum_e t\alpha \otimes w\gamma \cdot \mathbf{1} + \sum_e t\alpha \otimes W \cdot \mathbf{1} + \sum_e T \otimes w\gamma \cdot \mathbf{1}) < \infty \end{aligned} \tag{3.15}$$

Therefore, the average cost of this policy is finite. $\square$

3.4 Convexity of the Problem

In this section we prove that our problem is convex under the sufficient conditions so that it is guaranteed to have a global solution when we apply the Gradient Descent Optimization Algorithm. According to [75] a vector function is convex if it abides by the following definition:

Definition: Convexity of a function

A real valued function $f : Z^3 \to R$ is defined to be convex in i if and only if $f(i+1,j,k) - f(i,j,k)$ is non decreasing in i and similarly f is defined to be convex in j if and only if $f(i,j+1,k) - f(i,j,k)$ is non decreasing in j and f is defined to be convex in k if and only if $f(i,j,k+1) - f(i,j,k)$ is non decreasing in k [75].

Proposition 4

For $c_h \geq c_t/2$, the discounted cost function $v_\beta(i,j,e)$ is convex in i and j.

Proof of Proposition 4

We prove this by using the steps given in the Value Iteration Algorithm for the discounted cost function,

$$\begin{aligned} v_{\beta,n}(i,j,e) = \min_{a(s)\in A_{i,j,e}} & \{c_t a(s) + c_h[(i-a(s))^+ + (j-a(s))^+]\} \\ & + \beta \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e,e_{next}} v_{\beta,n-1}((i-a(s))^+ + k, (j-a(s))^+ + l, e_{next}) \end{aligned} \tag{3.16}$$

The first step of the value iteration when $n = 0$ is $v_{\beta,0}(i,j,e) = 0$, for every state (i,j,e). Therefore, $v_{\beta,n}(i,j,e)$ is convex in i and j for $n = 0$.

We then assume that $v_{\beta,n-1}(i,j,e)$ is convex in i and j, so that the function $f_{n-1}(i,j,e) = \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e,e_{next}} v_{\beta,n-1}(i+k, j+l)$ is also convex in i and j. We now rewrite (3.16) as,

$$v_{\beta,n}(i,j,e) = \min_{a(s)\in A_{i,j,e}} \{c_t a(s) + g_{n-1}((i-a(s))^+, (j-a(s)), e)\} \tag{3.17}$$

Where, $g_{n-1}(i,j,e) = c_h(i+j) + \beta f_{n-1}(i,j,e)$. Since $c_h(i+j)$ and $f_{n-1}(i,j,e)$ are convex in i and j, $g_{n-1}(i,j,e)$ is also convex in i and j.

From Lemma 1 in [57] we can state the following:

If $g_{n-1}(i,j,e)$ is convex in i and j, and if

$$\min\{g_{n-1}(1,0,e) - g_{n-1}(0,0,e), c_t\} \min\{g_{n-1}(0,1,e) - g_{n-1}(0,0,e), c_t\} \geq c_t \tag{3.18}$$

Then $c_t > 0$, $\min_{a(s)\in A_{i,j,e}} \{c_t a(s) + g_{n-1}((i-a(s))^+, (j-a(s)), e)\}$ is convex in i and j.

Since we have proved in Proposition 2 that $v_{\beta,n-1}(i,j,e)$ is non-decreasing in i and j at every stage of the Value Iteration Algorithm, we can state the following:

$$\begin{aligned} g_{n-1}(1,0,e) - g_{n-1}(0,0,e) = c_h + \beta \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e,e_{next}} (v_{\beta,n-1}(k+1, l, e_{next}) \\ - v_{\beta,n-1}(k, l+1, e_{next})) \geq c_h \end{aligned} \tag{3.19}$$

$$\begin{aligned} g_{n-1}(0,1,e) - g_{n-1}(0,0,e) = c_h + \beta \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e,e_{next}} (v_{\beta,n-1}(k+1, l, e_{next}) \\ - v_{\beta,n-1}(k, l+1, e_{next})) \geq c_h \end{aligned} \tag{3.20}$$

From (3.19) and (3.20) the condition to satisfy (3.18) is,

$$2c_h + 2\beta \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e,e_{next}} (v_{\beta,n-1}(k+1,l,e_{next}) - v_{\beta,n-1}(k,l+1,e_{next})) \geq c_t \qquad (3.21)$$

Therefore, we conclude that under the sufficient condition $c_h \geq c_t/2$, the discounted cost function $v_\beta(i,j,e)$ is convex in i and j. $\square$

3.5 Optimal Threshold Policy

In this section we prove that the optimal waiting policy for our transmission vs waiting problem is a threshold based policy dependent on the current phase vector of the arrival function.

Theorem 2

For $c_h \geq c_t$, (a) there exists two thresholds $L^\beta_{1,e}, L^\beta_{2,e} \geq 0$, for each queue that are dependent on the current phase vector $e = (b,d)$ of the arrival functions, such that the optimal action in state $(i,j,e) \in S$, in β-discounted cost problem, is given by:

$$a^*(i,j,e) = \min(i,j) + (i - \min(i,j) - L^\beta_{1,e})^+ + (j - \min(i,j) - L^\beta_{2,e}) \qquad (3.22)$$

(b) There is an average cost optimal policy $\theta(L^*_{1,e}, L^*_{2,e})$, such that $\theta(L^\beta_{1,e}, L^\beta_{2,e}) \to \theta(L^*_{1,e}, L^*_{2,e})$ as $\beta \to 1$, where $\theta(L^\beta_{1,e}, L^\beta_{2,e})$ is the β-discounted cost optimal policy described in (3.22).

Proof

$a(s) \in A_{i,j,e} = \min(i,j), \cdots, \max(i,j)$ can be written as $a(s) = \max(i,j) - (i - a(s))^+ - (j - a(s))^+$. Substituting $a(s)$ in $c(i,j,e,a(s))$ given in (3.1), we can rearrange the equation for $v_\beta(i,j,e)$ for every state $(i,j,e) \in S$ as follows:

$$v_\beta(i,j,e) = c_t \max(i,j) + c_h(t_b + w_d) + \min_{a(s)\in A_{i,j,e}} \{-(c_t - c_h)[(i-a(s))^+ + (j-a(s))^+] + \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e_n,e_{next}} v_{\beta,n-1}((i-a(s))^+ + k, (j-a(s))^+ + l, e_{next}))\} \quad (3.23)$$

From Proposition 4, $v_\beta(i,j,e)$ is convex in i and j.

Therefore, the function $f(i,j,e) = \sum_{k,l,e_{next}} p^{k,1}_{e,e_{next}} p^{l,2}_{e_n,e_{next}} v_{\beta,n-1}(i+k, j+l, e_{next}))$ is convex in i and j, and $-(c_t - c_h)(i+j)$ is also convex in i and j.

We now rewrite equation (3.23) as follows:

$$v_\beta(i,j,e) = c_t \max(i,j) + c_h(t_b + w_d) + \min_{a\in A_{i,j,e}} \{g((i-a(s))^+, (j-a(s))^+, e)\} \quad (3.24)$$

Where $g(i,j,e) = -(c_t - c_h)(i+j) + \beta f(i,j,e)$ and is convex in i and j.

We then consider the case $i \geq j$, so that the equation (3.24) can be written as,

$$\begin{aligned} v_\beta(i,j,e) &= c_t \max(i,j) + c_h(t_b + w_d) + \min_{a(s)\in\{j,\dots,i\}} g(i-a(s), 0, e) \\ &= c_t \max(i,j) + c_h(t_b + w_d) + \min_{b\in\{0,\dots,i-j\}} g_1(b,e) \end{aligned} \quad (3.25)$$

Where $b = i - a(s)$, and $g_1(b,e) = g(b,0,e)$.

Let $L^\beta_{1,e} = \text{argmin}\{g_1(b,e) : b \geq 0\}$ be a global minimum of $g_1(b,e)$ for phase vector $e = (b,d)$, then the minimum $g_1(b,e)$ for a given phase vector e is:

$$b^*(i,j,e) = \begin{cases} i-j & \text{if } 0 \leq i-j < L^\beta_{1,e} \\ L^\beta_{1,e} & \text{if } i-j \geq L^\beta_{1,e} \end{cases}$$

The optimal action in state (i, j, e) is given by:

$$a^*(i,j,e) = i - b^*(i,j,e) = \begin{cases} j & \text{if } 0 \le i - j < L_{1,e}^{\beta} \\ i - L_{1,e}^{\beta} & \text{if } i - j \ge L_{1,e}^{\beta} \end{cases}$$

Similarly, it can be proven for the case $i \le j$ that there exists a threshold $L_{2,e}^{\beta}$ for the arrival phase vector e such that the optimal action in state (i, j, e) is given by:

$$a^*(i,j,e) = i - b^*(i,j,e) = \begin{cases} i & \text{if } 0 \le j - i < L_{2,e}^{\beta} \\ j - L_{2,e}^{\beta} & \text{if } j - i \ge L_{2,e}^{\beta} \end{cases}$$

This proves that there exists an optimal threshold policy for each queue, that is dependent on the current arrival phase vector, which minimizes the β discounted average cost function. Given the above we have proven Theorem 2 (a), and Theorem 2 (b) which is implied by Theorem 1 (b), proven in section 3.4. $\square$

The intuition behind this multi-threshold policy can be explained by examining the behaviour of phase type distribution. When the data packets are arriving according to phase type distribution, the packet arrival probability changes according to the current phase of the arrival function, and as the system goes from one state to other, the arrival phase also changes so does the arrival probabilities at each queue. In other words at each time unit the arrival rates of the two nodes changes therefore to control the waiting policy the threshold levels at each queue changes.

3.6 Special Case of Systems with Unobservable Arrival Phase

In some cases the current phase of arrival is not observable. In these situations we cannot take the optimal action without first knowing the current phase or, more specifically in our case, determining the threshold policy. This is a common problem associated with phase type distribution that has not received due attention in the literature. Therefore, in this chapter we provide an approximation method for determining the current phase vector for systems with unobservable arrival phases. This method relies on selecting the relevant threshold probabilistically, according to the approximated phase of the arrival function.

We consider a phase type arrival at queue 1, defined by parameters (α, T), where the initial vector α is defined as:

$$\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_{n_1}\} \tag{3.26}$$

When the phase is not observable, we derive the following in order to find the long-term probability of the system being in a given phase:

$$\alpha^* = \alpha^*(T + t\alpha) = \{\alpha_1^*, \alpha_2^*, \cdots, \alpha_{n_1}^*\} \tag{3.27}$$

For queue 2, the arrival function is defined by parameters (γ, W), where the initial vector γ is defined as:

$$\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\} \tag{3.28}$$

Similar to queue 1, when the phase is not observable, the long-term probability of the system being in a given phase, for a given state, we derive the following:

$$\gamma^* = \gamma^*(W + w\gamma) = \{\gamma_1^*, \gamma_2^*, \cdots, \gamma_{n_2}^*\} \tag{3.29}$$

Therefore, the probability of the approximated phase vector e being $\{s_1, s_2\}$, where $1 \leq s_1 \leq n_1$ and $1 \leq s_2 \leq n_2$, is given by:

$$Pr(e_{s_1,s_2}) = \alpha_{s_1}^* \gamma_{s_2}^* \tag{3.30}$$

Given that,

$$L_{1,e}^{\beta} = \{L_{e_1}^1, L_{e_2}^1, \cdots, L_{e_{n_1 n_2}}^1\} \tag{3.31}$$

and,

$$L_{2,e}^{\beta} = \{L_{e_1}^2, L_{e_2}^2, \cdots, L_{e_{n_1 n_2}}^2\} \tag{3.32}$$

are the sets of optimal threshold values for phase vector $e \in E$ for queue 1 and queue 2, we probabilistically select two threshold values $L_{1,e}^{*\beta}$ and $L_{2,e}^{*\beta}$, according to the probability of the system being at the current phase vector $Pr(e_{s_1,s_2})$.

Chapter 4

Discrete Time Markov Chain to Solve for the Optimal Thresholds

Using the MDP, we proved in Chapter 3 that the optimal waiting policy for our problem is to use a threshold-based policy. However, solving the optimal threshold policy using the MDP is not feasible due to the infinite horizon nature of the problem. Therefore we present in this chapter a novel queuing theoretic model to solve the optimal threshold policy numerically. More specifically we develop a two-ended, FIFO PH/PH/1 push out, queuing model, to numerically determine the solutions to our optimal threshold policy. Our goal is to implement our threshold-based policy so as to minimize the total cost. Since this cost is equal to the total cost of waiting plus the total cost of transmission, we need to obtain the waiting time that each packet spends within a queue and the number of packets waiting. Since recording the individual age of every packet would make the state-space too large, we instead present a method to calculate the age of each packet. We note that each packet in the queue will eventually become the HoL packet before leaving the queue. If we record the age of the HoL packet and the time that the HoL packet leaves the queue then use this information in conjunction with, the inter-arrival times of each packet, we can calculate the individual age of every packet. There are two ways for the HoL packet to leave the queue.

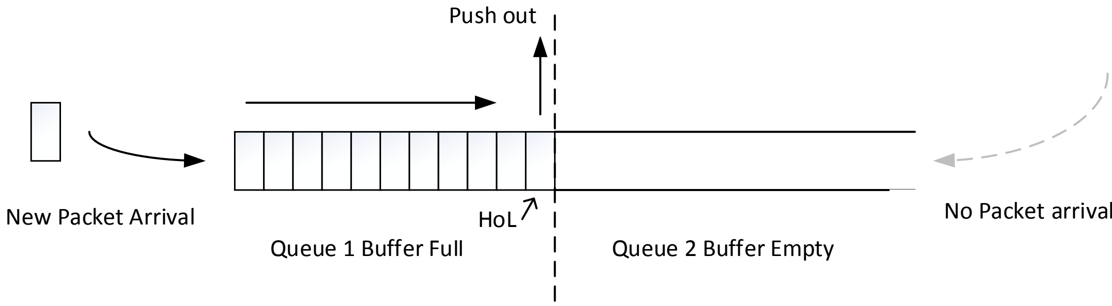


Figure 4.1: Proposed Two Ended PH/PH/1 Queue: Push-out Senario

The first way, as shown in Figure 4.1, is for the packet to be coded and served upon the arrival of a pairing packet in the other queue. The second way is for the packet to be pushed out of the queue upon meeting the waiting threshold. According to the best of our knowledge this is the first time that a method for analyzing the waiting times is used in a PH/PH/1 push out queueing model.

We then show how our DTMC model can be reducible to a level dependant QBD structure for efficient computing of the stationary vector. We then solve for this DTMC using the level dependant QBD structure in order to find the stationary distribution. Using the stationary distribution we then calculate the probability distribution of the age of each packet and the number of packets in the queues. Next, we derive the total cost by calculating the transmission and waiting costs. Finally, we present a GDA to solve for the optimal thresholds which minimizes the total cost.

4.1 Arrival Process

The arrival of data packets to each queue is modelled as a discrete time phase-type arrival function. The arrival process of queue 1 is defined by the parameters (α, T) with n_1 number of states. Similarly, the arrival process of queue 2 is defined by parameters (γ, W) and n_2

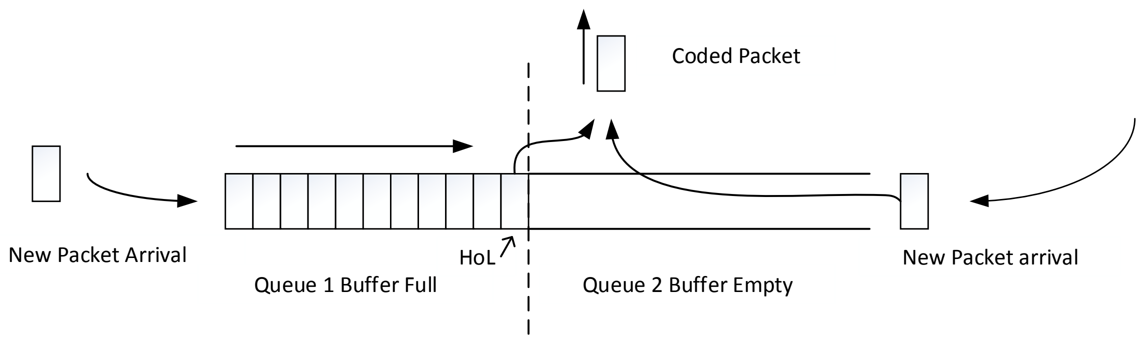


Figure 4.2: Proposed Two Ended PH/PH/1 Queue: Coded Packet Transmission

number of states.

The long-term average arrival probability of queue 1 can be found by,

$$p = \alpha^* t \tag{4.1}$$

where $\alpha^* = \alpha^*(T + t\alpha)$

Similarly, the long-term average arrival probability of queue 2 can be found by,

$$q = \gamma^* w \tag{4.2}$$

where $\gamma^* = \gamma^*(W + w\gamma)$

4.2 Service Process

Let us assume that both queues are empty when the system is initialized. If there are packet arrivals at both queues during a particular time unit, then both packets can be paired, coded together, and transmitted within one time unit. In the case where there is a packet arrival at only one queue, we wait for a packet arrival at the other queue so that both packets can be coded together. Then, it is clear that the service process of one queue is equivalent to the arrival process of the other queue, i.e., the service process is also a phase type distribution. In the case where a packet does not arrive in the other queue by the time the waiting threshold is reached, then we transmit the packet uncoded to avoid further delay. Similarly, if the current phase vector is e and the number of packets waiting in queue k $k = \{1, 2\}$ exceeds the threshold $L_{k,e}$, then the packets waiting in queue k are transmitted uncoded.

4.3 Discrete Time Markov Chain

We derive the DTMC shown in (4.3) to model the system. As this is a discrete time queueing model, we need to define the order of events happening within a time unit. We define one time unit such that arrivals occur at the beginning of the time unit and packet transmission at the end of the time unit. It is important to note that this order does not affect the steady state performance of the system.

Please note that we are recycling the previously used variables i, n, k, l to refer new set of parameters in this chapter.

This is a five dimensional DTMC therefore in order to model this system we need to record 5 variables in the DTMC: The number of packets waiting in each queue, the current arrival phase of each queue, the age of the HoL packets, and the phase of the HoL packet when it arrived. Therefore, the state space of the DTMC given by (4.3) is (n, i, s_1, s_s, s_2), where

- n is the number of packets in the non-empty queue (Note that one queue is always empty when the other queue is growing with waiting packets for a coding opportunity). We denote the number of waiting packets in queue 1 with positive numbers and that of queue 2 with negative numbers.
- i is the age of the HoL packet of the non-empty queue.

$$
P = \begin{array}{cc}
 & \begin{array}{ccccccccccccc} L_{2,e_{n^*}} & \cdots & L_{2,e_1} & \cdots & -2 & -1 & 0 & 1 & 2 & \cdots & L_{1,e_1} & \cdots & L_{1,e_{n^*}} \end{array} \\
\begin{array}{c} L_{2,e_{n^*}} \\ \vdots \\ L_{2,e_1} \\ \vdots \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ \vdots \\ L_{1,e_1} \\ \vdots \\ L_{1,e_{n^*}} \end{array} &
\begin{pmatrix}
A_0^{L_{2,e_{n^*}}} & A_1^{L_{2,e_{n^*}}} & \cdots & A_{n_2-1}^{L_{2,e_{n^*}}} & & & & & & & & & \\
\vdots & \vdots & \vdots & \vdots & & & & & & & & & \\
A_0^{L_{2,e1}} & A_1^{L_{2,e1}} & \cdots & A_{n_2-1}^{L_{2,e1}} & & & & & & & & & \\
 & & \ddots & \ddots & \ddots & & & & & & & & \\
 & & & A_{-1}^{-2} & A_0^{-2} & A_1^{-2} & & & & & & & \\
 & & & & A_{-1}^{-1} & A_0^{-1} & A_1^{-1} & & & & & & \\
 & & & & & A_{-1}^{0} & A_0^{0} & A_1^{0} & & & & & \\
 & & & & & & A_{-1}^{1} & A_0^{1} & A_1^{1} & & & & \\
 & & & & & & & A_{-1}^{2} & A_0^{2} & A_1^{2} & & & \\
 & & & & & & & & \ddots & \ddots & & \ddots & \\
 & & & & & & & & & A_{-(n_1+1)}^{L_{1,e_1}} & \cdots & A_{-1}^{L_{1,e_1}} & A_0^{L_{1,e_1}} \\
 & & & & & & & & & \vdots & \vdots & \vdots & \vdots \\
 & & & & & & & & & A_{-(n_1+1)}^{L_{1,e_{n^*}}} & \cdots & A_0^{L_{1,e_{n^*}}} & A_0^{L_{1,e_{n^*}}}
\end{pmatrix}
\end{array} \tag{4.3}
$$

- s_1 $(1 < s_1 \leq n_1)$ and s_2 $(1 < s_2 \leq n_2)$ are the phases of arrival processes of queue 1 and queue 2 respectively.
- s_s is the phase of the HoL packet of the non-empty queue, where $1 < s_s \leq n_1$ if queue 1 is non-empty and $1 < s_s \leq n_2$ if queue 2 is non empty.
- $n^* = e_{n_1} e_{n_2}$

Each level n of the Markov chain represents the number of packets waiting in the waiting queues and each level has sub levels in it to represent the age of the HoL packet of the non-empty queue.

Note that when $n = 0$ both queues are empty and hence there is no need to use s_s at level 0. In this case $i = 0$ and the state space reduces to (s_1, s_2).

Now we derive the elements of the DTMC as follows. Staying in the level 0 after a transition is defined by A_0^0. This happens when there are no arrivals to any queue or when there are arrivals to both queues then they are coded together and transmitted at the same time unit. In this situation the age of queues remains at zero,

$$A_0^0 = \begin{array}{cc} & 0 \\ 0 & \begin{pmatrix} T \otimes W + t\alpha \otimes w\gamma \end{pmatrix} \end{array} \tag{4.4}$$

Transition from level 0 to 1 happens when there is an arrival to queue 1 but not to queue 2; the packet arriving at queue 1 waits for a coding opportunity. The age of queue 1 now becomes 1. This is denoted by A_1^0. The transition from level 0 to 1 introduces the new state s_s to the state space at level 1. s_s is the phase of the HoL packet of the waiting queue (which is queue 1 in this case). There is only one packet waiting in this situation therefore $s_s = s_1$. This is represented by Z_1.

$$A_1^0 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & \cdots \end{array} \\ 0 & \begin{pmatrix} Z_1 \otimes W & 0 & 0 & \cdots \end{pmatrix} \end{array} \tag{4.5}$$

where,

$$Z_1 = X_1 \otimes Z_{R_1} + X_2 \otimes Z_{R_1} + ...X_i \otimes Z_{R_i} + ... + X_{n_1} \otimes Z_{R_{n_1}}, \tag{4.6}$$

Z_{R_i} is a row vector of zeros of length n_1^2 and the i^2th element is one

and,

$$[X_1, X_2, ..., X_k, ..., X_{n_1}] = t\alpha \tag{4.7}$$

Similarly a transition from level 0 to -1 happens when there is an arrival to queue 2 but not to queue 1. This situation is denoted by A_{-1}^0,

$$A_{-1}^0 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & \cdots \end{array} \\ 0 & \begin{pmatrix} T \otimes Z_{-1} & 0 & 0 & \cdots \end{pmatrix} \end{array} \tag{4.8}$$

where

$$Z_{-1} = F_1 \otimes Z_{L_1} + F_2 \otimes Z_{L_2} + ...F_i \otimes Z_{L_i} + ... + F_{n_2} Z_{L_{n_2}} \tag{4.9}$$

Z_{L_i} is a row vector of zeros of length n_2^2 and the i^2th element is one,

and,

$$[F_1, F_2, ..., F_k, ..., F_{n_2}] = w\gamma \tag{4.10}$$

Staying in the level 1 after a transition is denoted by A_0^1. Here there is only one packet in the queue and that packet's age can be increased by one level with no arrivals to both queues. Note that the phase of HoL packet(the only packet in this case) is unchanged in this situation. This is represented by I. When there are arrivals to both queues, the HoL packet in queue 1 gets served with the new arrival to queue 2. The age of queue 1 becomes the age of new packet which is 1 after the next transition. Now the new HoL packet is the packet that just arrived to queue 1 therefore the phase s_s which was equal to s_1 will now change according to the phase changes of the arrival to queue 1. This is represented by $\bar{T}$ which is the normalized matrix of $t\alpha$ such that $\bar{T}\mathbf{1} = \bar{T}$. $\bar{T}$ can be found by dividing each element in each row of $t\alpha$ by the sum of the elements in that row.

$$A_0^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & \cdots \end{array} \\ \begin{pmatrix} t\alpha \otimes \bar{T} \otimes w\gamma & T \otimes I \otimes W & 0 & 0 & 0 \\ t\alpha \otimes \bar{T} \otimes w\gamma & 0 & T \otimes I \otimes W & 0 & 0 \\ t\alpha \otimes \bar{T} \otimes w\gamma & 0 & 0 & T \otimes I \otimes W & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix} \end{array} \tag{4.11}$$

A_{-1}^1 represents the transition from level 1 to 0. This happens when there is an arrival to queue 2 but not to queue 1. In this situation the only packet waiting in queue 1 gets served with an arrival to queue 2. In this transition the phase s_s disappears at level 0 as defined by the Kronecker product with $\mathbf{1}$, where $\mathbf{1}$ is a column vector of ones with length n_1.

$$A_{-1}^{1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \end{array} \begin{array}{c} 0 \\ \begin{pmatrix} T \otimes \mathbf{1} \otimes w\gamma \\ T \otimes \mathbf{1} \otimes w\gamma \\ T \otimes \mathbf{1} \otimes w\gamma \\ \vdots \end{pmatrix} \end{array} \tag{4.12}$$

A_1^n represents the transition from level n to $n+1$. This happens when there is an arrival to queue 1 but not to queue 2. Queue 2 is waiting for a coding opportunity therefore the age of queue 1 is incremented by one. As there is no arrival to queue 2 the phase of the HoL packet is unchanged.

$$A_1^n = \begin{array}{c} \\ n \\ n+1 \\ n+2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} n+1 & n+2 & n+3 & \cdots \end{array} \\ \begin{pmatrix} t\alpha \otimes I \otimes W & 0 & 0 & 0 \\ 0 & t\alpha \otimes I \otimes W & 0 & 0 \\ 0 & 0 & t\alpha \otimes I \otimes W & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \end{array} \tag{4.13}$$

Staying at the same level n after a transition is defined by A_0^n.

$$A_0^n = \begin{array}{c} \\ n \\ n+1 \\ n+2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} n & n+1 & n+2 & \cdots & \cdots \end{array} \\ \begin{pmatrix} B_1 & C_0 & 0 & 0 & 0 \\ B_2 & C_1 & C_0 & 0 & 0 \\ B_3 & C_2 & C_1 & C_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array} \tag{4.14}$$

Here B_l represents the following situation: there are arrivals to both queues and there was no packet arrival to queue 1 during the next $l-1$ time units after the arrival of the HoL packet. The age of queue 1 now becomes n which is now the age of the next packet after the

HoL packet.

$$B_l = t\alpha \otimes T^{l-1} \otimes w\gamma \tag{4.15}$$

C_l represents the situation in which there is more than one packet waiting in queue 1. There are arrivals to both queues and the HoL packet of queue 1 is coded and transmitted with the arriving packet to queue 2. There was a packet arrival to queue 1 l time units after the HoL packet. This packet now becomes the HoL packet. Therefore the age of queue 1 is now decremented by $l-1$.

$$C_l = t\alpha \otimes T^{l-1} t\alpha \otimes w\gamma \tag{4.16}$$

C_0 represents the situation where there is a packet waiting in queue 1 and there is no arrival to any queue. Therefore the age of the queue 1 is incremented by one. In this case the phase of the HoL packet of queue 1 is not changed.

$$C_0 = T \otimes I \otimes W \tag{4.17}$$

Transition from level n to $n-1$ happens when there is an arrival to queue 2 but not to queue 1. This is defined by A_{-1}^{n},

$$A_{-1}^{n} = \begin{array}{c} \\ n \\ n+1 \\ n+2 \\ n+3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} n-1 & n & n+1 & n+2 & \cdots \end{array} \\ \begin{pmatrix} J_1 & 0 & 0 & 0 & \cdots \\ J_2 & H_1 & 0 & 0 & \cdots \\ J_3 & H_2 & H_1 & 0 & \cdots \\ J_4 & H_3 & H_2 & H_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array} \tag{4.18}$$

Here, J_l and H_l are defined in a similar manner to B_l and C_l respectively.

$$J_l = T \otimes T^{l-1} \otimes w\gamma \tag{4.19}$$

$$H_l = T \otimes T^{l-1} t\alpha \otimes w\gamma \tag{4.20}$$

The derivations of the remaining elements of the DTMC are given in the Appendix.

4.4 Solving the Discrete Time Markov Chain

To solve this Markov chain we approximate it by a finite state Markov chain by limiting the age of waiting packets to a finite value so that the sum of stationary probability values beyond that is negligibly small. This a realistic assumption as the age of the waiting packets are controlled by the waiting thresholds of each queue which prevents packets to be delayed for a prolonged time.

In order to fit this Markov chain into a known structure we block it in to different regions so that it gives a QBD structure. Figure 4.3 shows a simplified version of the Markov chain for given threshold values to demonstrate how fragmenting it into blocks can make a QBD structure. With this block structure we have a new set of composite levels 0 to L_1. So we label these blocks as $B_0^{(level)}$, $B_1^{(level)}$ and $B_{-1}^{(level)}$. However this QBD structure is level dependant. So we have to solve it by a recursive approach.

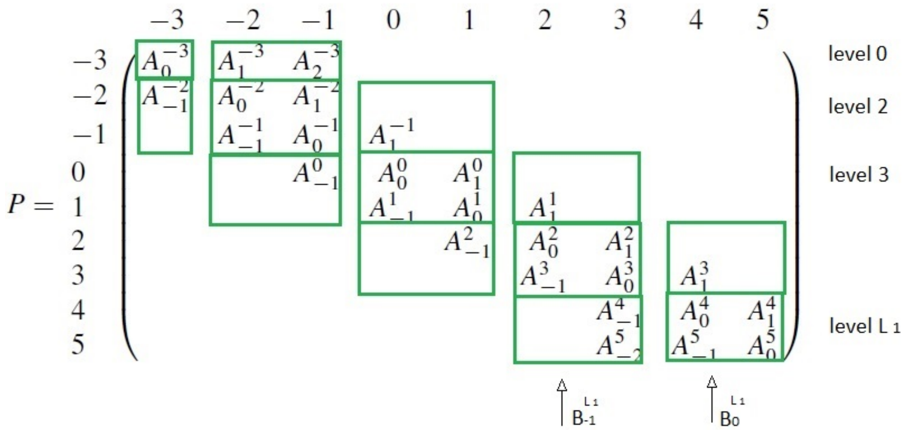


Figure 4.3: Level Dependant QBD Structure

The stationary vector $\mathbf{x} = \{x_0, \cdots, x_n, \cdots, x_{L_1}\}$

$$x_n = x_{n-1} R^{(n)} \tag{4.21}$$

where $R^{(n)}$ is the expected number of visits from level $n-1$ to n.

$$R^{(n)} = B_1^{(n-1)} (I - U^{(n)})^{-1} \tag{4.22}$$

$$U^{(n)} = B_0^{(n)} + B_1^{(n)} (I - U^{(n+1)})^{-1} B_{-1}^{(n+1)} \tag{4.23}$$

$U^{(n)}$ represents the transition probabilities from level n to level n before a visit to level $n-1$.

$(I - U^{(n)})^{-1} B_{-1}^{(n)}$ records the first passage probability from level n to level $(n-1)$.

The boundary condition $U^{(L_1)}$ is given by,

$$U^{(L_1)} = B_0^{(L_1)} \tag{4.24}$$

After finding $U^{(n)}$ for all the levels, we can find all $R^{(n)}$ values for $0 < n \leq L_1$. Now if we know the stationary vector x_0 of level 0 we can find the complete stationary vector $\mathbf{x} = \{x_0, \cdots, x_n, \cdots, x_{L_1}\}$ for all the levels as follows.

$$x_n = x_0 \prod_{1 \leq k \leq n} R^{(k)} \tag{4.25}$$

$$x_0 = x_0 (B_0^{(0)} + R^{(1)} B_1^{(0)}) \tag{4.26}$$

normalized by , $x_0 \sum_{n \geq 0} \prod_{1 \leq k \leq n} R^{(k)} \mathbf{1} = 1$

This gives us the stationary vector $\mathbf{x} = \{x_0, \cdots, x_n, \cdots, x_{L_1}\}$ which can be now decomposed in to the original levels $\{-L_{2,e_{n^*}}, \cdots, L_{1,e_{n^*}}\}$. Note that the stationary vector is five dimensional according to our state space definition.

Once we find the stationary vector $\mathbf{x}$, we can calculate the average waiting time W by,

$$W = \sum_i |i| y(i) \tag{4.27}$$

where $y(i)$ is the age distribution of packets that can be found by summing up all stationary probabilities of different phases and levels of each age,

$$y(i) = \begin{cases} \sum_{n,s_1,s_s,s_2} \mathbf{x}(n, i, s_1, s_s, s_2) & , \quad i \neq 0 \\ \sum_{s_1,s_2} \mathbf{x}_0(s_1, s_2) & , \quad i = 0 \end{cases} \tag{4.28}$$

The average number of transmissions can be found by the sum of the product of the stationary vector of each level with the probability of packet transmission events at that level,

$$\begin{aligned} Y = \mathbf{x}(0,0)(\alpha t \otimes \gamma w)\mathbf{1} + \sum_{n=1}^{L_{1,e_1}-1} \sum_i \mathbf{x}(n,i)((T \otimes I \otimes \gamma w)\mathbf{1}) \\ + \sum_{n=-1}^{-L_{2,e_1}+1} \sum_i \mathbf{x}(n,i)((\alpha t \otimes I \otimes W)\mathbf{1}) \\ + \sum_{i,e} \mathbf{x}(L_{1,e}, i)\mathbf{1} + \sum_{i,e} \mathbf{x}(-L_{2,e}, i)\mathbf{1} \end{aligned} \tag{4.29}$$

where,

$$\mathbf{x}(n,i) = \begin{cases} \{\mathbf{x}(n,i,0,0,0),...,\mathbf{x}(n,i,s_1,s_s,s_2),...,\mathbf{x}(n,i,n_1-1,n_1-1,n_2-1)\} & , \quad n > 0 \\ \{\mathbf{x}(n,i,0,0,0),...,\mathbf{x}(n,i,s_1,s_s,s_2),...,\mathbf{x}(n,i,n_1-1,n_2-1,n_2-1)\} & , \quad n < 0 \end{cases} \tag{4.30}$$

The total average cost per given threshold values can be found by,

$$c(L_{1,e}, L_{2,e}, e) = c_h W + c_t Y \tag{4.31}$$

where c_h is the holding cost per packet per time unit and c_t is the transmission cost per packet.

Our objective is to find the thresholds $L_{1,e}$ and $L_{2,e}$ for all phase vectors e to minimize the total average cost.

$$\{L_{1,e}, L_{2,e}\} = \underset{L_{1,e},L_{2,e}}{\operatorname{argmin}} \; c(L_{1,e}, L_{2,e}, e) \tag{4.32}$$

4.5 Finding the Optimal Solution Using a Gradient Descent Algorithm

We have proven that our optimization problem is convex given the sufficient condition of $c_h > c_t/2$. Therefore, we use a GDA 1) to obtain the global optimal solution. In the case that $c_h < c_t/2$, using GDA 1 is not guaranteed to produce the global optimal solution for the threshold policy. However, in our analysis every case we tested where $c_h < c_t/2$, the problem was convex, and thus the GDA 1 provided the global optimal solution.

Therefore we make the assumption that the problem is convex for all values of c_h/c_t in the absence of a counter example to suggest otherwise. However, in the Numerical Results section of this thesis, we show that our solution is most efficient in terms of total cost for all regions of c_h/c_t compared to other policies. The values of the function need to be calculated for different $L_{1,e}, L_{2,e}$ values at each phase vector e, by generating the DTMC given in (4.3) and finding the stationary vector. Since the minimum cost is dependent on the threshold values $L_{1,e}, L_{2,e}$, we use The Gradient Descent Algorithm 1 to solve this minimization problem. The algorithm starts from the origin $L_{1,e} = 0, L_{2,e} = 0$. For every phase vector e a search is over one axis using a Gradient Descent Search while values on the other axis remain fixed. This search is run iteratively by resuming from the previous stopping point on each axis. The iterations continue until the minimum point is found for each phase vector e. Here, m is the gradient of the cost function at the given point and α is a parameter which can be adjusted increase the rate of convergence. The value selected for α defines the step size $\lceil m\alpha \rceil$ for searching. If α is too large then the step sizes will be too big and the minimum point may be skipped during the search. However if α is too small, the step size will be too close to unity level which defeats the purpose of using the algorithm. Therefore, a good step size is chosen in accordance with the system characteristics.

Algorithm 1 Finds the Minimum Cost

```
L_{1,e} ← 0, L_{2,e}, min ← ∞
for doe ∈ (s_1, s_2) :
    loop_{L_{2,e}} : Similar to Loop_{L_{1,e}}. L_{2,e} changes instead of L_{1,e}. L_{1,e} and e are fixed at
    previously assigned values
        loopL_{1,e} :
            m ← C_{(L_{1,e},L_{2,e},e)} − C_{(L_{1,e}+1,L_{2,e},e)}
            if m > 1 then
                L_{1,e} ← L_{1,e} + ⌈mα⌉
                if min ≥ C_{(L_{1,e},L_{2,e},e)} then
                    min ← C_{(L_{1,e},L_{2,e},e)}
                else
                    L_{1,e} ← L_{1,e} − ⌈mα⌉
                    BREAK loop_{L_{1,e}}
                end if
                goto loopL_{1,e}
            else
                if m < 1 then
                    L_{1,e} ← L_{1,e} − 1
                    if min ≥ C_{(L_{1,e},L_{2,e},e)} then
                        min ← C_{(L_{1,e},L_{2,e},e)}
                    else
                        L_{1,e} ← L_{1,e} + 1
                        BREAK loop_{L_{1,e}}
                    end if
                    goto loopT_1
                else
                    if m = 0 then
                        min ← C_{(L_{1,e},L_{2,e},e)}
                        BREAK loop_{L_{1,e}}
                    end if
                end if
            end if
        end loop
    end loop
end for
return min, L_{1,e}, L_{1,e}
```

Chapter 5

Numerical Results

In this chapter, we investigate the behaviour of our system by analyzing the numerical results obtained by using the proposed model.

We start by defining the arrival functions used to generate the results. To do this, we created test phase type function matrices, Arrival 1, Arrival 2, Arrival 3 and Arrival 4, whose outputs are different long-term arrival probabilities. We then used these test phase type arrival functions to generate our numerical results.

Arrival 1:

$$\alpha = [0.5, 0.5], T = \begin{bmatrix} 0.5 & 0.3 \\ 0.2 & 0.6 \end{bmatrix}.$$

Arrival 2:

$$\alpha = [0.5, 0.5], T = \begin{bmatrix} 0.25 & 0.35 \\ 0.25 & 0.55 \end{bmatrix}.$$

Arrival 3:

$$\alpha = [0.5, 0.5], T = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}.$$

Arrival 4:

$$\alpha = [0.5, 0.5], T = \begin{bmatrix} 0.22 & 0.45 \\ 0.25 & 0.13 \end{bmatrix}.$$

We plot the probability distribution of the number of waiting packets in each queue as shown in Figure. 5.1. Here, both queues have a fixed threshold, $L = 10$, and the long term arrival probability of Queue 1 $>$ Queue 2. We then observe how each queue builds up with packet arrivals and transmissions. Note that the graphs have been plotted using continuous line graphs (as opposed to discrete) to avoid appearing cluttered. Figure 5.1 shows us how the queue with a higher arrival rate will have more packets waiting than the queue sending packets at a slower rate.

Figure 5.3 reveals how the average number of transmissions per time unit decreases when the waiting threshold size of each queue increases. Note that the transmission rate is 0.4 packets per time unit when the threshold levels are set to zero, and that this rate is equivalent to the case where there is no buffering of packets.

We observe that the rate of packet transmissions will converge to the rate of the arrival function with the greatest average long-term arrival probability. The reason for this is that the packets in the queue with the lower arrival rate are more likely to find packets in the other queue with the higher arrival rate for coding. As the excess packets in the queue with the higher arrival rate are transmitted un-coded, the transmission rate converges to the rate of the queue of the higher arrival rate. As the buffer size increases, the average number of transmissions converges to the rate of the arrival function with the greatest average long-term arrival probability. The reason for this is when we have larger buffers, the packets are able to wait long enough to have the maximum number of coding opportunities. In Figure 5.2 we show how the average waiting time increases as we increase the waiting thresholds of each queue. Therefore as the buffer size increases, the waiting time increases. However as demonstrated in Figure. 5.3 as the buffer size increases, the average number of transmissions decreases. These two trends demonstrate the trade-off between waiting vs. transmission. Our objective is to find the optimal point that minimizes both the total waiting and transmission costs. To do this we generate and compare numerical results for the optimal total cost according to the following four policies :

Method 1: Multi-Threshold Policy

This is the phase dependent threshold policy we have proposed for phase type arrivals. Each queue has a waiting threshold dependent on the current phase vector of the arrival function. We expect this policy to be optimal.

Method 2: Probabilistic Threshold Policy

In the case where the arrival phase is not observable, we present this policy to probabilistically select the waiting policy.

Method 3: No-Waiting Policy

Under this policy, the access point transmits all the packets at every time unit without delaying. If there are packets coming from both nodes the access point will code them, but will immediately transmit any packets that are unable to find a coding opportunity. Note that this policy is expected to be effective when the holding cost c_h is high.

Method 4: One-Sided Waiting Policy

This policy is based on the long term probability of packet arrivals to queues 1 and 2 respectively. Under this policy, upon transmission of coded packets, all the remaining packets in the queue with the higher arrival probability are sent un-coded. This is the policy that can achieve the highest coding opportunities without the system being unstable since the highest rate of coded packets cannot be more than the rate of the arrival function with the lowest average long-term arrival probability. This is similar to the policy proposed in [39]

Method 5: Geomatric Approximation

In this policy we take the mean of the arrivals and consider it as a Geomatric arrival and use the single threhold policy proposed in [57]

Since the latter four policies are specific threshold policies, we expect that they will not perform better than our multi-threshold policy which we have proven to be optimal. To demonstrate this, we will investigate the efficiency of these threshold-specific policies in comparison to our proposed policy.

In addition to using results obtained from our proposed model, we also present the results from a simulation of our proposed model. Within MATLAB we simulated the two queues under phase type arrivals as well as the coded and un-coded packet transmissions as packets arrive. By storing the age of each waiting pack we calculated the total cost of the system. We show that our simulation-based results verify the analytical results generated from our queuing model.

Next, we present our numerical results to compare the long-run minimum average costs of our system model under different arrivals. We compare the average costs of the policies normalized by the transmission cost c_t over different possible values of c_h/c_t.

From our numerical results we observe that the phase dependent multi-threshold policy achieves the lowest average cost when compared to methods 2, 3 and 4, as shown in

Figures 5.5. Both the Multi-Threshold Policy and the One-Sided Waiting Policy converge at extremely low c_h/c_t ratios. This is because when the holding cost c_h is very low, both policies have a large enough waiting thresholds to achieve the maximum coding opportunities. The Approximated Threshold Policy shows slightly higher cost when compared to these two policies at extremely low c_h/c_t because it chooses the threshold probabilistically when the phase is not observable. Since there is lack of information about the current phase, this policy cannot present the optimal decision. Figure 5.4 shows a comparison between our proposed threshold method vs the Geomatric approximation method. The Geomatric approximation of the mean of the arrivals gives higher cost compared to the multi-threshold and probabilistic threshold methods. The reason for this is, we do not take the arrival phase into account when deciding the thresholds using the mean of the arrivals. The arrival probabilities in Phase type distributions depend on the current phase, therefore considering the current phase for determining the threshold policy helps us to make better decisions compared to Geomatric approximation where we only consider the over all mean of the arrivals to decide the thresholds.

At extremely high c_h/c_t ratios, both the Multi-Threshold Policy and the Approximated Threshold Policy converge to the cost given by the No-Waiting Policy. The reason for this is that at very high holding costs, transmitting packets with little to no waiting time is cost-effective due to the relatively high holding costs. When the difference between the long-term arrival probabilities of each queue increases, both the Multi-Threshold and Probabilistic Threshold policies show significant cost reduction within a lager range of c_h/c_t. This is because when the difference between rates of packet arrival increases, storing packets in a buffer can increase the probability of having more coding opportunities. Furthermore, when the two queues have nearly the same packet arrivals rates, there is a high probability of packet pair formation for coding with less waiting. Waiting longer would only be beneficial when the holding cost is relatively low.

Another important parameter for our system is the Coding Ratio which represents the efficiency of each policy at generating network coding opportunities. The Coding Ratio is defined as the long-run proportion of coded packets divided by the total number packet transmissions. Figure 5.6 present the Coding Ratio attained by each policy over different values of c_h/c_t . From these figures, we observe that the coding ratio of the Multi-Threshold Policy and the Approximated Threshold Policy are consistently between the coding ratios of the No-Waiting Policy and the One-Sided Waiting Policy. The No-Waiting Policy has the lowest Coding Ratio since it never holds a packet to wait for a possible network coding opportunity. The One-Sided Waiting Policy consistently gives the maximum Coding Ratio as it can achieve the most coding opportunities in a stable system. Any policy that has a longer waiting policy than the One-Sided Waiting Policy may result in a case with an infinite buffering of packets and thus an unstable system. Since the maximum Coding Ratio is correlated with a higher waiting cost, and the minimum coding ration is correlated with increased transmission cost, the optimal policy must lie between these two limits. As the c_h/c_t decreases, the Coding Ratio of the Multi-Threshold Policy converges to the Coding-Ratio of the One-Sided Waiting Policy. This is because when the holding cost is very low, it most efficient for the system to wait until the maximum number of coding opportunities is achieved. Similarly, when c_h/c_t is very high, the Coding Ratio of the Multi-Threshold Policy converges to the Coding Ratio of the No-Waiting Policy, which is the minimum Coding Ratio. This is because when the holding cost is very high, it is cost-effective to transmit packets un-coded than waiting for coding opportunities.

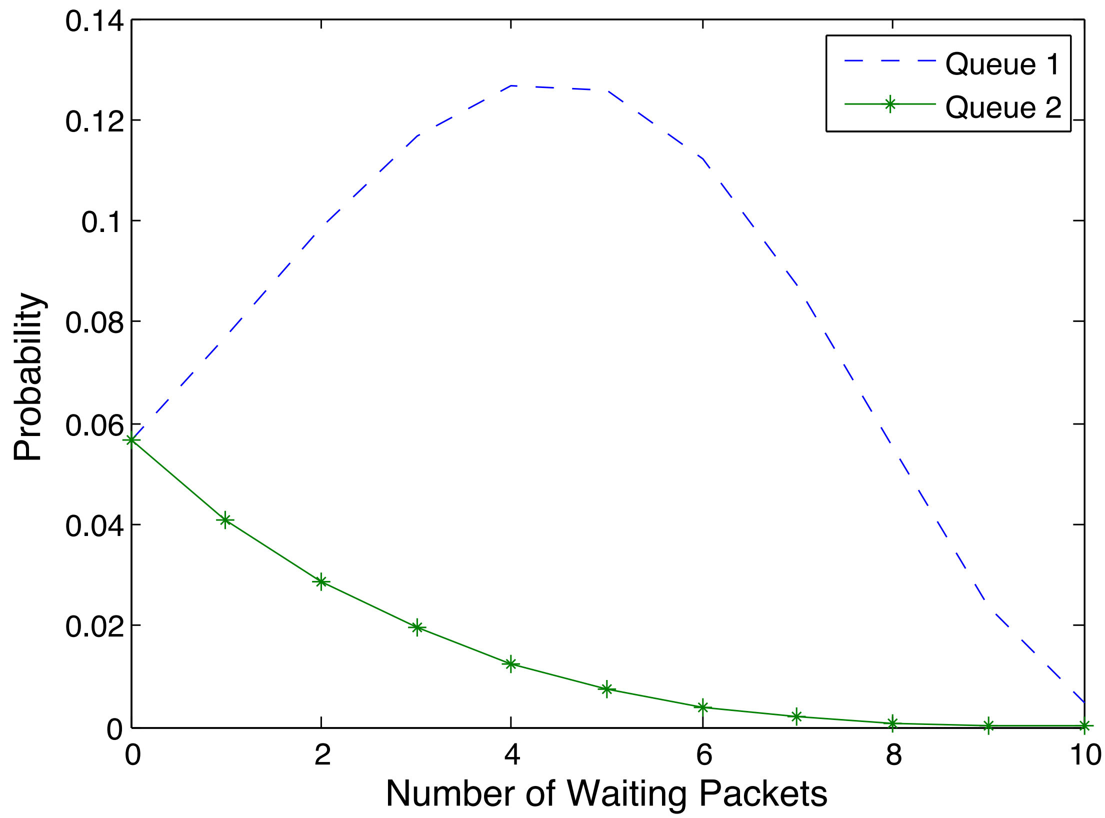


Figure 5.1: Average Number of Waiting Packets in Each Queue: Queue 1: Arrival 2, Queue 2: Arrival 1

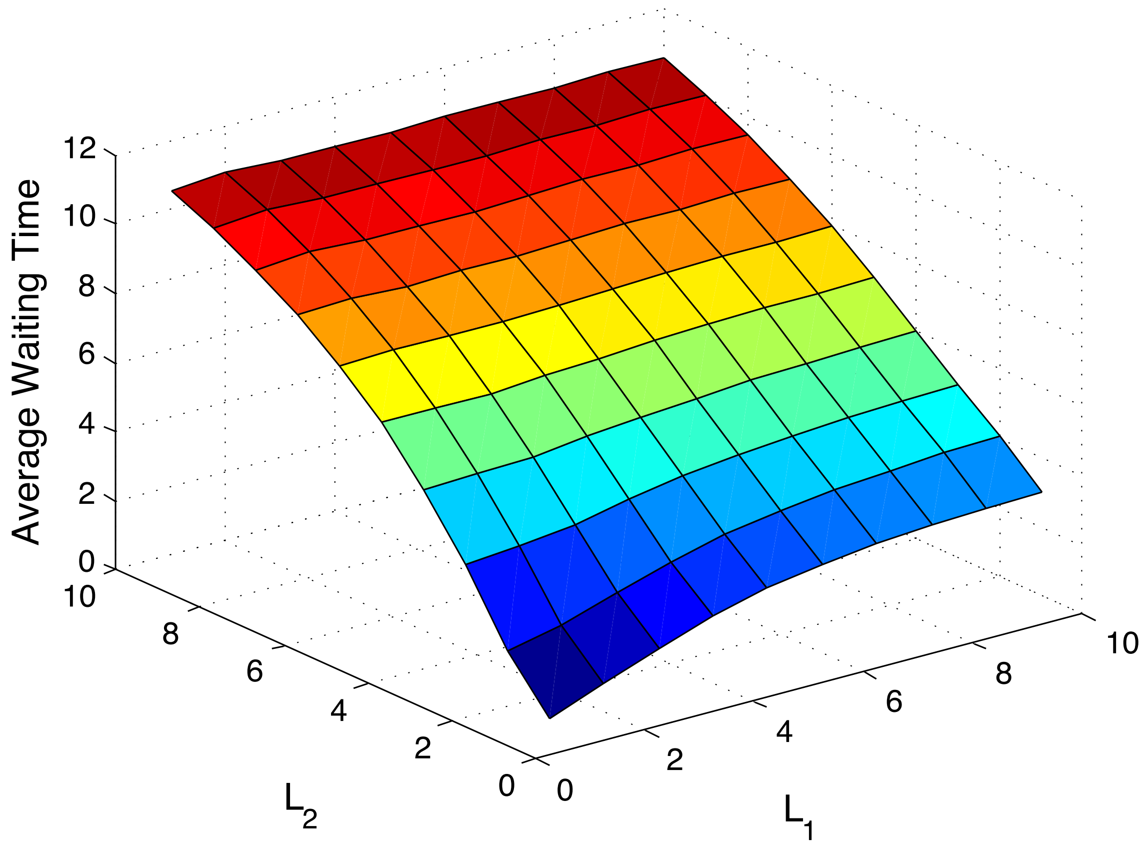


Figure 5.2: Average Waiting Time per Packet vs Buffer Size: Queue 1: Arrival 2, Queue 2: Arrival 1

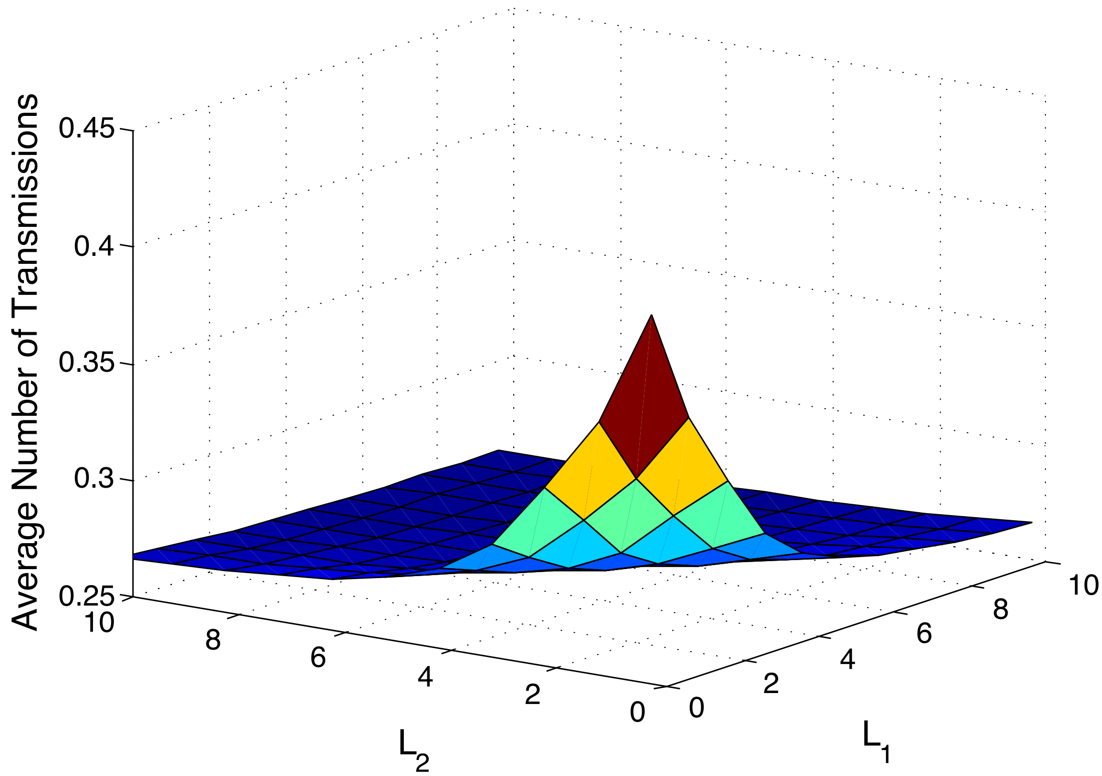


Figure 5.3: Average Number of Transmissions vs Buffer Size: Queue 1:Arrival 2, Queue 2: Arrival 1

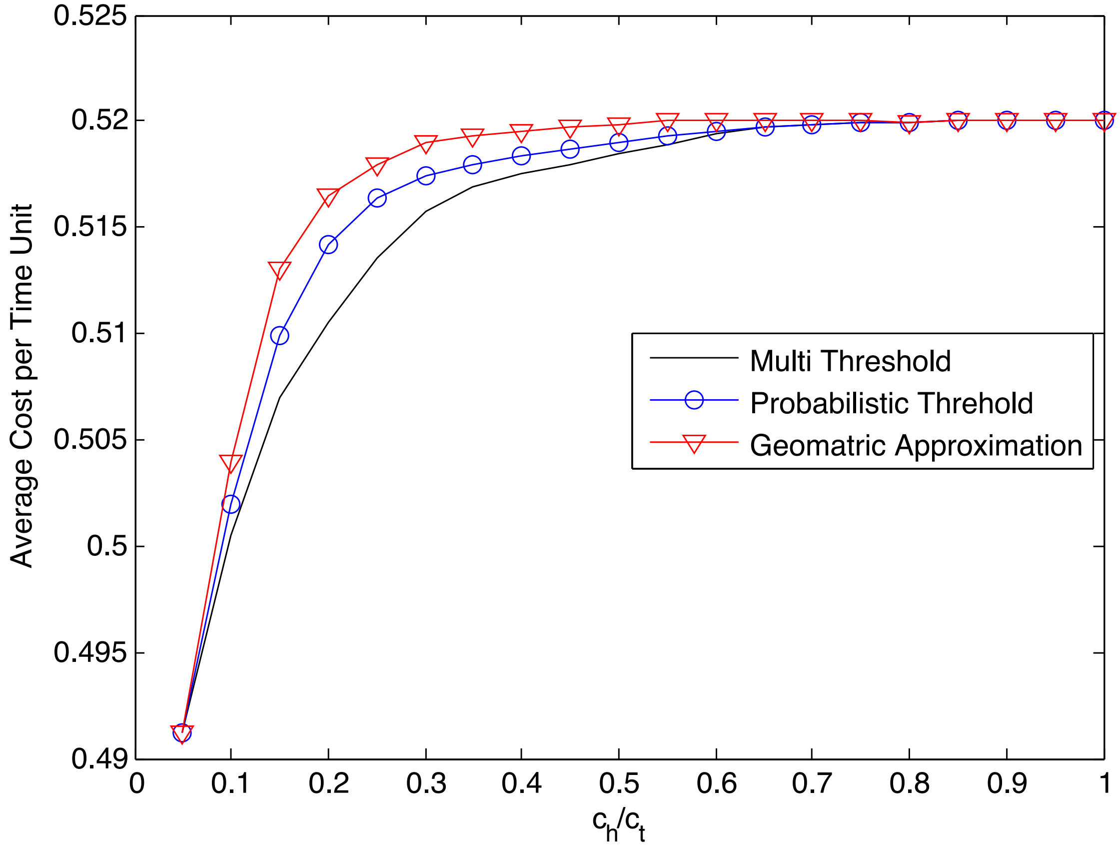
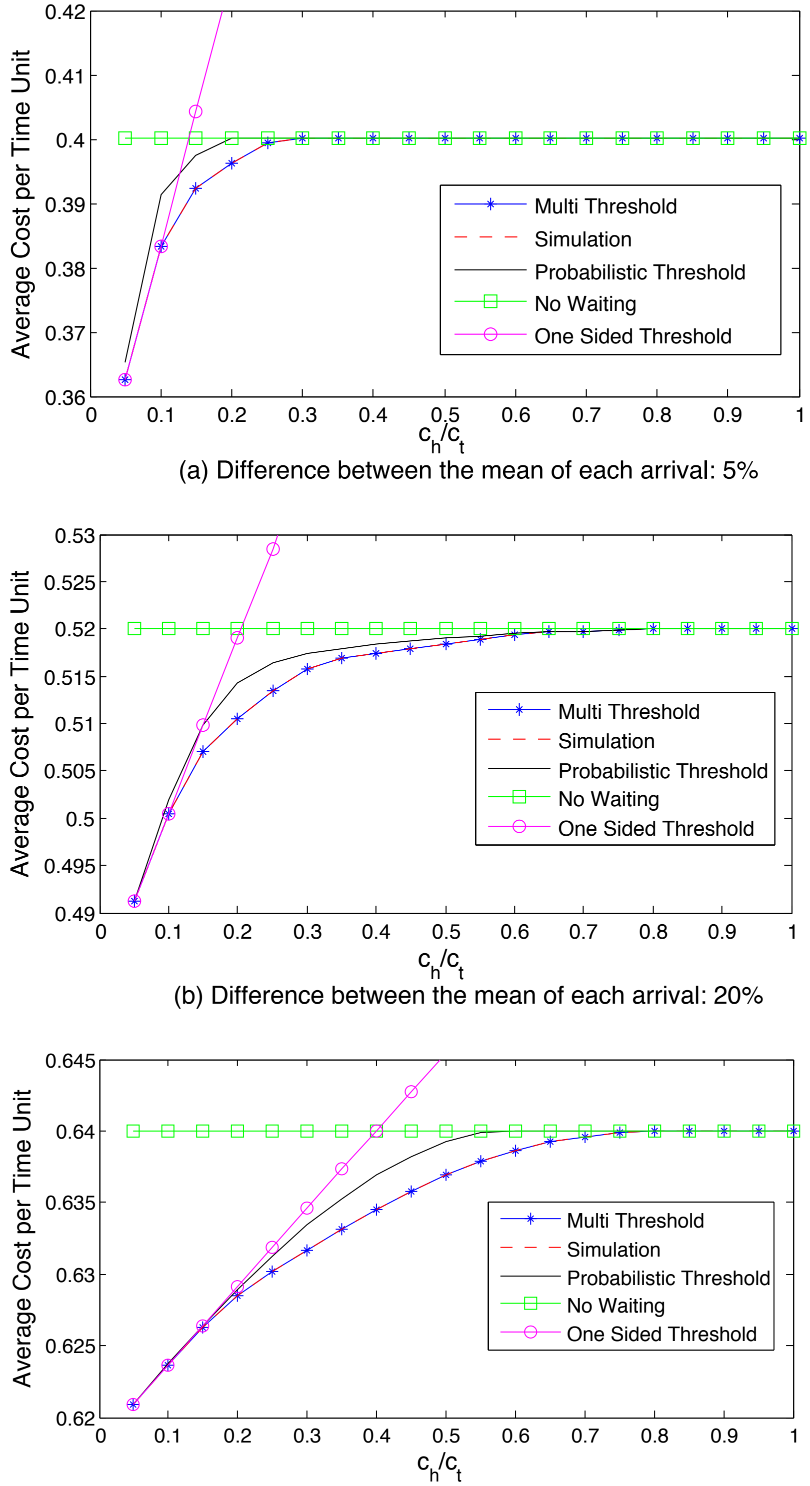


Figure 5.4: Comparison of Average Cost: Queue 1:Arrival 1, Queue 2: Arrival 3

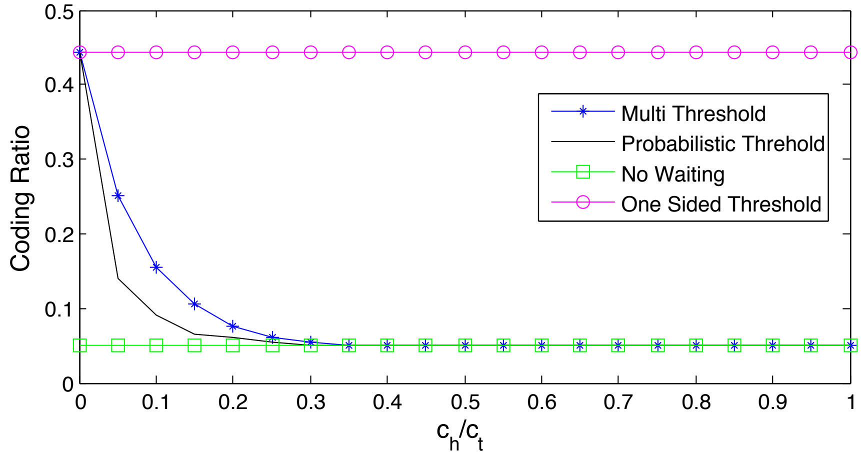


(a) Difference between the mean of each arrival: 5%

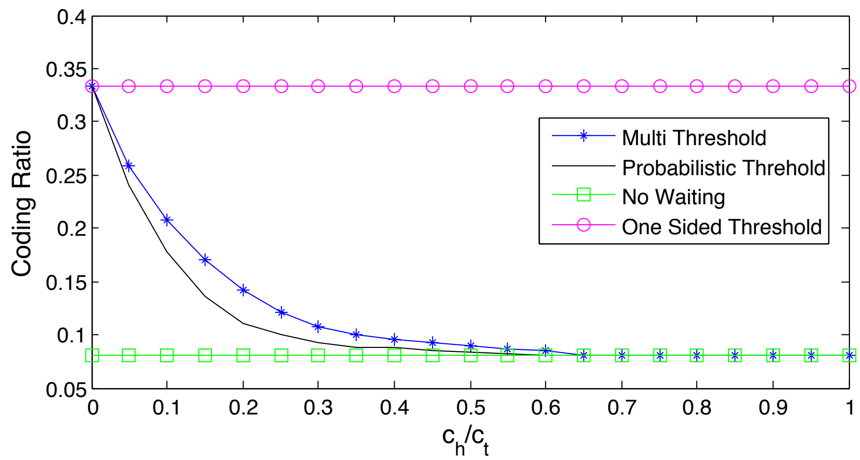
(b) Difference between the mean of each arrival: 20%

(c) Difference between the mean of each arrival: 40%

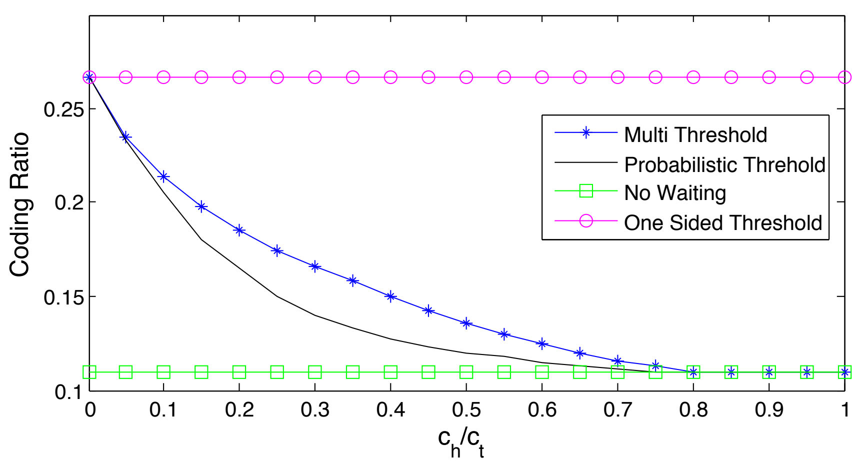
Figure 5.5: Comparision of Average Cost: (a) Arrival 1 and Arrival 2, (b) Arrival 1 and Arrival 2, (c) Arrival 1 and Arrival 4



(a) Difference between the mean of each arrival: 5%



(b) Difference between the mean of each arrival: 20%



(c) Difference between the mean of each arrival: 40%

Figure 5.6: Comparision of Coding Ratio: (a) Arrival 1 and Arrival 2, (b) Arrival 1 and Arrival 3, (c) Arrival 1 and Arrival 4

Chapter 6

Conclusion and Future Work

We have analyzed the Waiting vs. Transmission trade-off in Opportunistic Network-Coding in the context of a two-way relay wireless access point with phase-type packet arrivals. For our analysis we developed an MDP model to find an optimal waiting policy that minimizes the total long-term average cost. We proved the existence of an average-cost optimal policy within our MDP model, and that our problem was convex under sufficient conditions. Then, using the Steps of Value Iterations Algorithm for the MDP, we proved that the nature of the global optimal solution is a threshold-based policy, which defines a separate threshold for each phase vector of phase-type arrival functions within each queue. For the case of unobservable arrival phases, we introduced a technique to probabilistically predict the arrival phase vector in order to determine a threshold policy.

Due to the difficulty of using this MDP to numerically solve for the optimal solution, we developed a novel FIFO PH/PH/1 two-ended DTMC to model this system and efficiently calculate the optimal solution using a gradient descent algorithm. Using a technique that records the age of the HoL packet and the time that it leaves the queue, we presented a novel technique to keep track of the age and the number of waiting packet in the two ended push-out queue. We then showed how this DTMC can be reducible to a level dependant QBD structure to compute the stationary vector, and then calculate the total transmission and waiting costs.

Using the numerical results generated for several test arrivals, we showed that our proposed multi-threshold policy is most effective at minimizing the total cost for all the ranges of the normalized packet holding cost, c_h/c_t. Furthermore, we compared the performance of our threshold policy with the One-Sided Waiting Policy (the policy with the highest Coding Ratio within a stable system) and also with No-Waiting Policy, (the policy with the lowest Coding Ratio within a stable system) which will always transmit unpaired packets without waiting.

From our numerical results we conclude when the arrival is observable that the Multi-Threshold Policy performs better than the other waiting policies in all situations. However, the No-Waiting and One-Sided Waiting policies provide the nearly same performance as the Multi-Threshold Policy in two different operating regions. Whereas the One-Sided waiting policy performs near-optimally when the normalized holding cost c_h/c_t is very small, the No-Waiting Policy performs near-optimally when the normalized holding cost c_h/c_t is very large. These policies are especially attractive within their respective regions due to their simplicity and ease of implementation. Therefore, at the extremes of a very large normalized holding cost c_h/c_t, and very small normalized holding cost one should select the No-Waiting and One-Sided policies, respectively. Using this approach would eliminate the need for additional resources for the computation of the Multi-Threshold Policy. However, for all other values of c_h/c_t our Multi-Threshold Policy should be used to minimize the average total cost. In the case that the arrival phase is not observable our Approximated Threshold Policy should be used for all other values of c_h/c_t.

6.1 Possible Future Extensions

A possible future extension of our work is to apply our Multi-Threshold Policy to a star network topology as opposed to our two-way relay Access Point topology.

To extend the work to more than 2 nodes one could develop a star network protocol based on Time Division Multiple Access (TDMA). For example, consider a traditional wireless star network with a single antenna communicating with nodes using TDMA. It takes N TDMA time slots to collect data from N nodes and another N time slots to switch the collected data to the intended destinations.

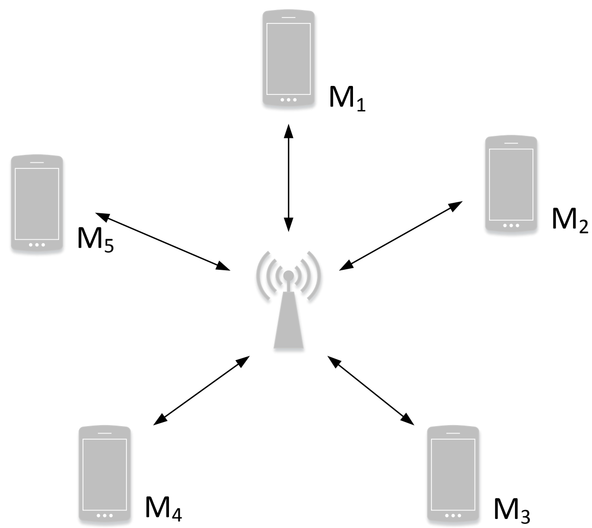


Figure 6.1: TDMA Wireless Start Network

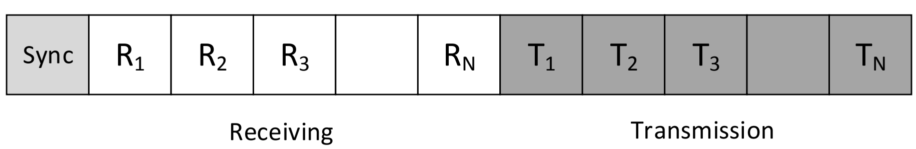


Figure 6.2: Time Slot Arrangement of One TDMA Cycle

We can introduce network coding to this and reduce the number of time slots required for transmission. For example if M_1 communicates with M_2, and M_3 communicates with

M_4, we need to utilize four transmission timeslots to switch packets from M_1 to M_2, M_1 to M_2, M_3 to M_4 and M_4 to M_3 but we can do the same communication by transmitting two coded packets $M_1 \oplus M_2$ and $M_3 \oplus M_4$ using only two time slots. We can see that network coding can reduce the number of TDMA time slots $M(M < N)$ (Figure 6.3) in a wireless star network..

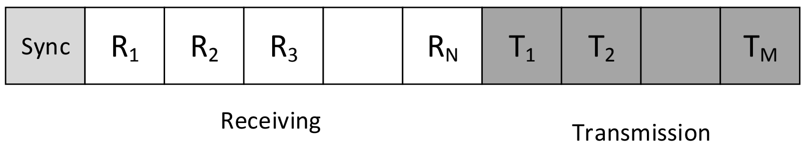


Figure 6.3: Reduced transmittion time slots with network coding

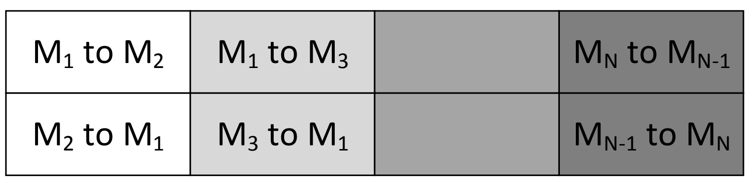


Figure 6.4: Proposed New Data Queues Pairs in the AP

The AP creates pairs of packets based on the source and the destinations of packets, then codes them and transmits. It can maintain queues according to flow information. There are $N(N-1)$ queues for each source destination, pairs M_1 to M_2, M_1 to M_3, M_1 to M_4, etc., as shown in Figure 6.4. However, finding an optimal waiting policy for the access point to optimize the total cost will be a challenging problem.

During each TDMA cycle, each N nodes in the star network can generate data packets according phase type distributions and transmit to one or more of the $N-1$ destination nodes via the AP. Thus these needs to be buffered at the AP upon their arrival based on their arrival and destination nodes so that they can be paired for network coding. Since some of these packets can be coded together, we can save the number of TDMA time slots required for transmissions and reduce the length of the overall TDMA frame the all network throughput. The graph in Figure. 6.5 shows a simple test case simulation where every node always transmits a data packet to a randomly selected destination node during every TDMA frame. Based on their source and destination nodes the data packets together at the AP and transmitted as coded packets. The simulation was run for 30 nodes and shows how changing M (the number of transmission time slots) in the TDMA frame can increase the overall network throughput.

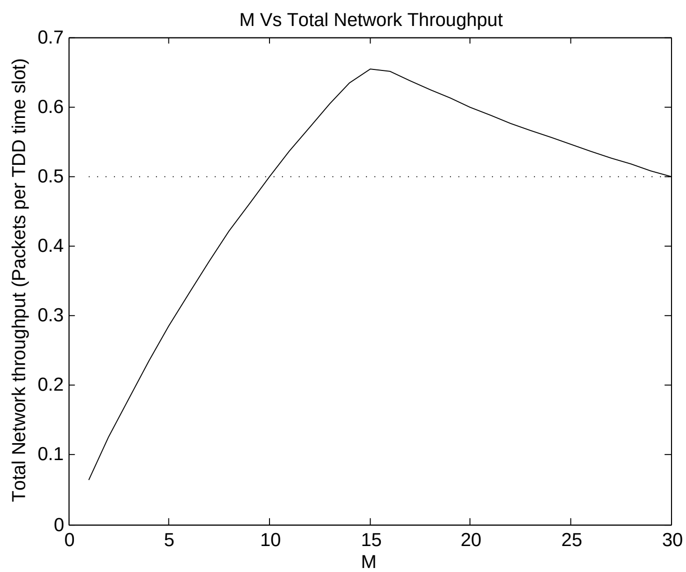


Figure 6.5: Network Throughput vs M (N=30).

In a more practical setting these nodes generate data packets according to a general arrival function which can be modelled using discrete phase type distribution and it might be advantageous for some packets what cannot be paired during the same TDMA to be delayed for a coding opportunity. The optimal waiting policy for these packets can be dependent on a number of parameters such as, the number of packets waiting in each queue, the age of the waiting packets, the arrival functions of each nodes, and the QoS requirements etc. And also once packets have been paired for coding, there should be a transmission scheduling algorithm to determine the priority of the packets to be allocated to the available TDMA time slots. Therefore, one can use extend the work de presented for optimal waiting policy and develop an optimal waiting and scheduling policy to minimise the total cost of communication in a star network topology.

Appendix A

Derivation of the rest of the DTMC in (4.3)

$$
A_1^{-i} = \begin{array}{cc}
 & \begin{matrix} -(i-1) & -i & -(i+1) & -(i+2) & \cdots \end{matrix} \\
\begin{matrix} -i \\ -(i+1) \\ -(i+2) \\ -(i+3) \\ \vdots \end{matrix} &
\begin{pmatrix}
\bar{J}_1 & 0 & 0 & & \cdots \\
\bar{J}_2 & \bar{H}_1 & 0 & 0 & \cdots \\
\bar{J}_3 & H_2 & \bar{H}_1 & 0 & \cdots \\
\bar{J}_4 & H_3 & \bar{H}_2 & \bar{H}_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix}
\end{array}
\tag{A.1}
$$

$$
A_0^{-i} = \begin{array}{cc}
 & \begin{matrix} -i & -(i+1) & -(i+2) & -(i+3) & \cdots \end{matrix} \\
\begin{matrix} -i \\ -(i+1) \\ -(i+2) \\ \vdots \end{matrix} &
\begin{pmatrix}
\bar{B}_1 & \bar{C}_0 & 0 & 0 & 0 \\
\bar{B}_2 & \bar{C}_1 & \bar{C}_0 & 0 & 0 \\
\bar{B}_3 & \bar{C}_2 & \bar{C}_1 & \bar{C}_0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix}
\end{array}
\tag{A.2}
$$

$$
A_{-1}^{-i} = \begin{array}{c c} & \begin{array}{cccc} -(i+1) & -(i+2) & -(i+3) & \cdots \end{array} \\ \begin{array}{c} -i \\ -(i+1) \\ \vdots \\ \\ \end{array} & \begin{pmatrix} w\gamma \otimes I \otimes T & 0 & 0 & 0 \\ 0 & w\gamma \otimes I \otimes T & 0 & 0 \\ 0 & 0 & \gamma \otimes I \otimes T & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \end{array} \tag{A.3}
$$

Where,

$$\bar{J}_l = t\alpha \otimes W^{l-1} \otimes W \tag{A.4}$$

$$\bar{H}_l = t\alpha \otimes W^{l-1} w\gamma \otimes W \tag{A.5}$$

$$\bar{V}_l = (t\alpha + T) \otimes W^{l-1} w\gamma \otimes W \tag{A.6}$$

$$\bar{X}_l = (t\alpha + T) \otimes W^{l-1} \otimes W \tag{A.7}$$

$$\bar{B}_l = t\alpha \otimes W^{l-1} \otimes w\gamma \tag{A.8}$$

$$\bar{C}_l = t\alpha \otimes W^{l-1} w\gamma \otimes w\gamma \tag{A.9}$$

$$\bar{C}_0 = T \otimes I \otimes W \tag{A.10}$$

The matrices, $A_0^{L_{1,e_1}} \cdots A_{-(n_1+1)}^{L_{1,e_1}} \cdots A_0^{L_{1,e_{n1}}} \cdots A_{-(n_1+1)}^{L_{1,e_{n1}}}$ and $A_0^{L_{2,e_1}} \cdots A_{(n_1+1)}^{L_{2,e_1}} \cdots A_0^{L_{2,e_{n2}}} \cdots A_{(n_2+1)}^{L_{2,e_{n2}}}$ represents the transmissions when the waiting thresholds hit while waiting. If the number of waiting packets exceeds the threshold we transmit all the excess packets encoded, so these elements can be defined using a similar approach we took to define the previous elements the only difference is when a threshold is met the next transition level is determined by the threshold value corresponding to current phase vector.

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, Jul 2000.

[2] T. Ho and D. Lun, *Network Coding: An Introduction.* Cambridge University Press, 2008.

[3] W. Chen, K. Letaief, and Z. Cao, “Opportunistic network coding for wireless networks,” in *IEEE International Conference on Communications*, pp. 4634–4639, June 2007.

[4] H. Xiang and A. Yener, “On the energy-delay trade-off of a two-way relay network,” *42nd Annual Conference on Information Sciences and Systems (CISS)*, pp. 865–870, Mar. 2008.

[5] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, “Minimum-cost multicast over coded packet networks,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2608–2623, June 2006.

[6] L. Chen, S. Low, M. Chiang, and J. Doyle, “Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[7] L. Chen, T. Ho, S. Low, M. Chiang, and J. Doyle, "Optimization based rate control for multicast with network coding," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1163–1171, May 2007.

[8] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *IEEE Transactions on Information Theory*, vol. 55, pp. 797–815, Feb 2009.

[9] S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *ACM Transactions on Networking*, vol. 16, pp. 497–510, Jun. 2008.

[10] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *IEEE International Symposium on Information Theory*, pp. 1758–1762, July 2006.

[11] M. A. Johnson and M. R. Taaffe, "Matching moments to phase distributions: Mixtures of erlang distributions of common order," *Communications in Statistics. Stochastic Models*, vol. 5, no. 4, pp. 711–743, 1989.

[12] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory*, pp. 442–, June 2003.

[13] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, Oct 2006.

[14] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2235–2245 vol. 4, March 2005.

[15] “http://www.bittorrents.com.”

[16] M. Wang and B. Li, “R2: Random push with random network coding in live peer-to-peer streaming,” *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 1655–1666, December 2007.

[17] D. Lucani, M. Stojanovic, and M. Medard, “Random linear network coding for time division duplexing: When to stop talking and start listening,” in *IEEE INFOCOM 2009*, pp. 1800–1808, April 2009.

[18] D. Lucani, M. Medard, and M. Stojanovic, “Random linear network coding for time-division duplexing: Queueing analysis,” in *IEEE International Symposium on Information Theory, ISIT 2009*, pp. 1423–1427, June 2009.

[19] Y. Wu, “Existence and construction of capacity-achieving network codes for distributed storage,” *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 277–288, February 2010.

[20] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sept 2010.

[21] B. Akdeniz, B. Tepekule, A. Pusane, and T. Tugcu, “Network coding applications in molecular communication,” in *23th Signal Processing and Communications Applications Conference (SIU)*, pp. 1212–1215, May 2015.

[22] V. Kebkal, O. Kebkal, and K. Kebkal, “Network coding for underwater acoustic networks of butterfly topology,” in *OCEANS 2014 - TAIPEI*, pp. 1–5, April 2014.

[23] B. Shrader and A. Ephremides, “A queueing model for random linear coding,” in *IEEE Military Communications Conference, 2007. MILCOM 2007.*, pp. 1–7, Oct 2007.

[24] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, “Minimum-cost multicast over coded packet networks,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2608–2623, June 2006.

[25] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, “Capacity of wireless erasure networks,” *IEEE Transactions on Information Theory*, vol. 52, pp. 789–804, March 2006.

[26] D. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, “Achieving minimum-cost multicast: a decentralized approach based on network coding,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 1607–1617 vol. 3, March 2005.

[27] J. Zhang, K. Cai, K. Letaief, and P. Fan, “A network coding unicast strategy for wireless multi-hop networks,” in *IEEE Wireless Communications and Networking Conference, WCNC 2007.*, pp. 4221–4226, March 2007.

[28] S. Sengupta, S. Rayanchu, and S. Banerjee, “An analysis of wireless network coding for unicast sessions: The case for coding-aware routing,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications.*, pp. 1028–1036, May 2007.

[29] S. Chachulski, S. Chachulski, M. Jennings, M. Jennings, S. Katti, D. Katabi, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” 2007.

[30] S. Katti and D. Katabi, “Mixit: The network meets the wireless channel,” *Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory*, 2007.

[31] T. Cui, L. Chen, and T. Ho, “Energy efficient opportunistic network coding for wireless networks,” in *INFOCOM 2008. The 27th IEEE Conference on Computer Communications*, April 2008.

[32] P. Popovski and H. Yomo, "Physical network coding in two-way wireless relay channels," in *IEEE International Conference on Communications, ICC '07.*, pp. 707–712, June 2007.

[33] P. Larsson, N. Johansson, and K.-E. Sunell, "Coded bi-directional relaying," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2, pp. 851–855, May 2006.

[34] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information Theory*, vol. 48, pp. 1135–1149, May 2002.

[35] M. Neely, "Optimal energy and delay tradeoffs for multi-user wireless downlinks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[36] M. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 2915–2934, July 2006.

[37] E. Ciftcioglu, A. Yener, and R. Berry, "Stability of bi-directional cooperative relay networks," in *IEEE Information Theory Workshop, ITW '08.*, pp. 144–148, May 2008.

[38] Y. Sagduyu, D. Guo, and R. Berry, "Throughput optimal control for relay-assisted wireless broadcast with network coding," in *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008. SECON Workshops '08. 5th IEEE Annual Communications Society Conference on*, pp. 1–6, June 2008.

[39] J. Zhang and P. Fan, "Optimal scheduling for network coding: Delay v.s. efficiency," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, Dec 2010.

[40] P. Ostovari, J. Wu, and A. Khreishah, “Deadline-aware broadcasting in wireless networks with local network coding,” in *2012 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1123–1127, Jan 2012.

[41] Y. Sagduyu and A. Ephremides, “Some optimization trade-offs in wireless network coding,” *40th Annual Conference on Information Sciences and Systems*, pp. 6–11, Mar. 2006.

[42] R. B. J. Goseling and J. van Ommeren, “Queueing analysis of wireless network coding,” *Dept of Applied Mathematics, University of Twente*, 2011.

[43] D. Umehara, S. Denno, M. Morikura, and T. Sugiyama, “Coded packet immediate access for contention-based wireless relay networks,” in *4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–9, Dec 2010.

[44] O. Abdelrahman and E. Gelenbe, “Performance trade-offs in a network coding router,” *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, pp. 2–5, Aug. 2010.

[45] E.-C. Davri, K. Kontovasilis, and C. Skianis, “Optimized network coding efficiency under qos constraints in two-way relay networks with timeouts,” in *IEEE International Conference on Communications (ICC)*, pp. 6157–6162, June 2015.

[46] S. J. Kim, P. Mitran, C. John, R. Ghanadan, and V. Tarokh, “Coded bi-directional relaying in combat scenarios,” in *IEEE Military Communications Conference, MILCOM*, pp. 1–7, Oct 2007.

[47] Y. Sagduyu and A. Ephremides, “Cross-layer optimization of mac and network coding in wireless queueing tandem networks,” *IEEE Transactions on Information Theory*, vol. 54, pp. 554–571, Feb 2008.

[48] C.-H. Liu and F. Xue, "Network coding for two-way relaying: Rate region, sum rate and opportunistic scheduling," in *IEEE International Conference on Communications, ICC '08.*, pp. 1044–1049, May 2008.

[49] O. Abdelrahman and E. Gelenbe, "Queueing performance under network coding," in *IEEE Information Theory Workshop on Networking and Information Theory, ITW 2009*, pp. 135–139, June 2009.

[50] J. Price and T. Javidi, "Network coding games with unicast flows," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1302–1316, September 2008.

[51] J. Marden and M. Effros, "A game theoretic approach to network coding," in *IEEE Information Theory Workshop on Networking and Information Theory, ITW 2009.*, pp. 147–151, June 2009.

[52] E. Ciftcioglu, Y. Sagduyu, R. Berry, and A. Yener, "Cost-delay tradeoffs for two-way relay networks," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 4100–4109, December 2011.

[53] M. Amerimehr and F. Ashtiani, "Delay and throughput analysis of a two-way opportunistic network coding-based relay network," *IEEE Transactions on Wireless Communications*, vol. 13, pp. 2863–2873, May 2014.

[54] A. Khreishah, I. Khalil, P. Ostovari, and J. Wu, "Flow-based xor network coding for lossy wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 2321–2329, June 2012.

[55] H.-M. Wang, X.-G. Xia, and Q. Yin, "A linear analog network coding for asynchronous two-way relay networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3630–3637, December 2010.

[56] P. Upadhyay and S. Prakriya, "Performance bounds for analog network coding based two-way relaying with multiuser selection diversity," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 333–338, March 2011.

[57] A. Mohapatra, N. Gautam, S. Shakkottai, and A. Sprintson, "Network coding decisions for wireless transmissions with delay consideration," *IEEE Transactions on Communications*, vol. 62, pp. 2965–2976, Aug 2014.

[58] J. Sundararajan, D. Shah, and M. Medard, "On queueing in coded networks - queue size follows degrees of freedom," in *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, pp. 1–6, July 2007.

[59] Y. Yuan, K. Wu, and W. Jia, "On the queueing behavior of network coding," in *IEEE International Symposium on Network Coding (NetCod)*, pp. 1–6, June 2010.

[60] L. Ding, F. Yang, L. Qian, L. Liu, and P. Wu, "Cost-delay trade off of network coding in asymmetric two-way relay networks," in *International Symposium on Network Coding (NetCod)*, pp. 1–4, July 2011.

[61] S. Chieochan, E. Hossain, T. Issariyakul, and D. Niyato, "Opportunistic network coding and dynamic buffer allocation in a wireless butterfly network," in *IEEE Global Telecommunications Conference, GLOBECOM 2009.*, pp. 1–6, Nov 2009.

[62] P. Parag and J. Chamberland, "Queueing analysis of a butterfly network for comparing network coding to classical routing," *IEEE Transactions on Information Theory*, vol. 56, pp. 1890–1908, April 2010.

[63] R. Zhang and Q. Qian, "Probabilistic network coding with priority over wireless networks," in *2011 Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 2, pp. 918–921, Jan 2011.

[64] C. Gunasekara, A. Alfa, and P. Yahampath, "A queueing theoretic model for opportunistic network coding," in *International Conference on Computing, Networking and Communications (ICNC)*, pp. 999–1004, Jan 2013.

[65] B. van Houdt, R. Lenin, and C. Blondia, "Delay distribution of (im)patient customers in a discrete time d-map/ph/1 queue with age-dependent service times," *Queueing Systems*, vol. 45, no. 1, pp. 59–73, 2003.

[66] M. F. Neuts and A. S. Alfa, "Pair formation in a markovian arrival process with two event labels," *Journal of Applied Probability*, vol. 41, pp. 1124–1137, Dec. 2004.

[67] A. S. Alfa, *Queueing Theory for Telecommunications - Discrete Time Modelling of a Single Node System.* Springer, 2010.

[68] M. O. Sren Asmussen, Olle Nerman, "Fitting phase-type distributions via the em algorithm," *Scandinavian Journal of Statistics*, vol. 23, no. 4, pp. 419–441, 1996.

[69] A. S. Alfa, "Discrete time queues and matrix-analytic methods," *Top*, vol. 10, no. 2, pp. 147–185, 2002.

[70] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[71] L. I. Sennott, "Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs.," *Operations Research*, vol. 37, no. 4, p. 626, 1989.

[72] L. Sennott, *Handbook of Markov Decision Processes*, vol. 40. International Series in Operations Research and Management Science, 2002.

[73] M. L. Puterman, *Markov decision processes : discrete stochastic dynamic programming.* Wiley series in probability and mathematical statistics, New York: John Wiley & Sons, 1994. A Wiley-Interscience publication.

[74] L. I. Sennott, "Another set of conditions for average optimality in markov control processes," *Systems and Control Letters*, vol. 24, no. 2, pp. 147 – 151, 1995.

[75] K. Murota, *Recent developments in discrete convex analysis.* Springer-Verlag, 2009.