ABSTRACT

TAGGART, CHRISTOPHER S. Average Packet Delay Analysis for Ultra Wideband Wireless Networks of Simple Nodes. (Under the direction of Professor Yannis Viniotis and Professor Mihail L. Sichitiu).

Networks using ultra-wideband wireless (UWB) as the physical layer show promise for application as personal area networks. UWB can provide high capacity (on the order of 1.3 Gbps over short distances) wireless links and allow variable capacity links to be formed between nodes. However, as with all communication technologies, there is an upper bound on the aggregate capacity of all links that transmit within specified interference boundaries.

Certain characteristics of UWB wireless links give rise to properties not seen with other wireless technologies. Two such properties are long synchronization times for establishing links between nodes and the ability to change individual link capacities by choosing pseudo-random (PN) codes of different lengths for channel coding.

The number of incoming and outgoing links for a single UWB node is driven by cost and technology and affects the topologies available for a network formed from UWB nodes. Whether composed of nodes with a single incoming and a single outgoing UWB link or M incoming and outgoing links UWB networks can be designed to minimize average network delay. UWB networks of different scales and node types also have unique management problems requiring specific management protocols in order to function efficiently. This thesis addresses the following problems related to networks composed of simple UWB nodes:

- Given a traffic matrix specifying the average packet delivery rate between nodes and an aggregate capacity bound for all links, determine a topology for the network and capacities for each link to minimize average packet delay.

- Given a traffic matrix specifying the average packet delivery rate between nodes, determine the effect of the acquisition time to establish a new network link on average packet delay and when packet switching operation performs better than packet forwarding operation.
• Given a ring network of UWB nodes, determine whether adding additional single direction links can improve delay performance for the network. Consider both fixed links and links that switch between nodes as a function of time.

• Given a ring network of UWB nodes, determine the network management issues for the network and propose protocols to efficiently address these issues.

Our contribution is three-fold: first, we derive analytical expressions for average delay in UWB networks with ring topologies. Then, we consider ways to add additional fixed or switching links to UWB ring networks in order to reduce average packet delay. We validate these analytical results via extensive simulations. Finally, we discuss the performance of protocols for managing such networks.
Average Packet Delay Analysis for Ultra Wideband Wireless Networks of Simple Nodes

by

Christopher S. Taggart

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2008

APPROVED BY:

Dr. Arne A. Nilsson
Dr. Michael Devetsikiotis

Dr. George N. Rouskas
Dr. Mihail L. Sichitiu
Co-Chair of Advisory Committee

Dr. Yannis Viniotis
Co-Chair of Advisory Committee
DEDICATION

To my wife Elizabeth for her constant love and encouragement. To my children Ross and Amy, who pulled me away from this work at all the right moments. To my parents Donald and Marlene Taggart, who encourage me in my academic endeavors.
BIOGRAPHY

Chris Taggart was born in a Navy family and grew up all over the country and in several parts of the world. He earned a Bachelor of Science degree in Optics in 1982 from the University of Rochester before entering the U.S. Navy to serve for six years as a submarine officer. He married fellow University of Rochester classmate and Naval Officer Elizabeth Pedro in 1983. In 1992, Chris earned a Masters of Science degree in Electrical Engineering at the University of Washington.

Chris and Elizabeth have two children. Their son Ross, entered Duke University in the Fall to study Engineering. Their daughter, Amy, is a Junior in high school. Chris and his family live in Greensboro, North Carolina where he is an engineering manager for General Dynamics Advanced Information Systems. He continues to serve the Navy and his country as a Captain in the Reserve Engineering Duty Officer community.
ACKNOWLEDGMENTS

I thank Yannis Viniotis and Mihail Sichitiu for their constant optimistic support and for always being available by e-mail, telephone, or in person for the next review or discussion along the way! I thank my committee members, Arne Nilsson, Mike Devetsikiotis, and George Rouskas for their advice and encouragement. I thank Matt Stallmann for sharing his expertise in proving intractability of mathematical problems. I thank Keith Townsend for introducing me to the ultra-wideband wireless research field. I thank Larry Hooker for encouraging me to pursue a doctorate. Finally, I thank General Dynamics for supporting this work.
# TABLE OF CONTENTS

**LIST OF TABLES** ......................................................................................... vii

**LIST OF FIGURES** ....................................................................................... viii

1 **Introduction** ............................................................................................. 1
   1.1 A Brief History of Ultra-wideband Wireless .................................. 2
   1.2 Status of UWB Standards ................................................................. 3
   1.3 Thesis Organization ........................................................................... 3

2 **Ultra-wideband Networking** ................................................................. 5
   2.1 UWB Characteristics and Usefulness for Wireless Networking ........ 5
   2.2 The Problems Studied in this Thesis ................................................ 6

3 **Minimizing Delay in UWB Ring Networks** ........................................... 8
   3.1 Topology Effects on Delay ............................................................... 9
   3.2 Choosing Link Capacities to Minimize Delay ................................ 11
   3.3 Effect of Ring Order on Delay ........................................................ 15
   3.4 Effect of Node Acquisition Times on Delay .................................... 19
      3.4.1 Three Node Network Conceptual Model ................................. 22

4 **Minimizing Delay in UWB Ring Networks with Added Links** .......... 37
   4.1 Effect of Adding Links on Network Topology .............................. 37
   4.2 Choosing Link Capacities to Minimize Delay ............................... 40
   4.3 Comparison of Delay for Network Topologies with Equal Loading ... 43
      4.3.1 Loading and Delay in the Ring Network ................................. 44
   4.4 Effect of Node Switching for Various Network Topologies ......... 52
      4.4.1 Conceptual Models ................................................................. 55
      4.4.2 Mathematical Models ............................................................ 57
      4.4.3 Simulation Models ................................................................. 61
      4.4.4 Results .................................................................................... 64
      4.4.5 Summary ............................................................................... 78

5 **UWB Network Management Challenges** ......................................... 79
   5.1 Joining a Ring .................................................................................. 80
      5.1.1 WTRP Join Method ................................................................. 80
      5.1.2 Bluetooth Join Method ............................................................ 81
      5.1.3 SDD Join Method ..................................................................... 82
      5.1.4 Analysis of Joining Protocol Suitability for UWB Networks .... 82
      5.1.5 Proposed UWB Ring Network Join Protocol ..................... 85
      5.1.6 Join Protocol Performance Comparisons ............................... 89
   5.2 Leaving a Ring ................................................................................... 93
LIST OF TABLES

Table 3.1 Queue service status and time intervals ........................................ 26
Table 3.2 Switching model queue size approximations ................................. 27
Table 3.3 Switching model time duration approximations ............................ 28
Table 3.4 Average packet delay comparison for packet forwarding models ........ 34

Table 4.1 Number of links needed for topologies of interest ....................... 39
Table 4.2 Average delay equations for various network topologies ............... 43
Table 4.3 Links and offered load in a ring network of R nodes with one bisecting cross-link ................................................................. 46
Table 4.4 Delay equations for various network topologies assuming equal loading ...... 47

Table 5.1 SDD parameters used to compare discovery overhead .................... 92
# LIST OF FIGURES

Figure 2.1 Two options to transfer information from a source S to two different destinations $D_1$ and $D_2$. ................................................................. 5

Figure 3.1 Block diagram of a network node .................................................. 9

Figure 3.2 If the topology is not a ring, some of the nodes cannot receive any traffic. . 11

Figure 3.3 Traffic on each link is the sum of offered loads that must traverse that link 12

Figure 3.4 The sum of the offered loads in the ring is the weighted sum of the diagonals of the traffic matrix ............................................................... 16

Figure 3.5 The several tours that must be optimized for a five node ring ................. 17

Figure 3.6 Heuristic methods compared well with exhaustive search .................... 20

Figure 3.7 Performance of heuristic methods difficult to evaluate for large networks . 21

Figure 3.8 Three node packet forwarding diagram ......................................... 22

Figure 3.9 Three node network switching diagram ........................................... 23

Figure 3.10 Queue operation as nodes switch and reacquire wireless links ............ 24

Figure 3.11 Lines of equal delay for packet forwarding and link switching operation based on the mathematical models ......................................................... 32

Figure 3.12 Comparison of average packet delay for packet forwarding versus packet switching operation based on the mathematical models ...................... 33

Figure 3.13 Good agreement is seen in model comparisons for $t_{synch} = 50msec$ and two values of average packet service rate ........................................... 34

Figure 3.14 The mathematical model underestimates delay in model comparisons for $t_{synch} = 50\mu sec$ and two values of average packet service rate ................. 35

Figure 3.15 Subplots of individual queue delays in model comparisons for $t_{synch} = 50msec$ and $t_{synch} = 50\mu sec$. ......................................................... 36

Figure 4.1 Plain eight node ring and two ways to add an extra single direction link... 38
Figure 4.2 Two ways to add two extra single direction links to an eight node ring .... 38
Figure 4.3 Three ways to add even more single direction links to an eight node ring . 39
Figure 4.4 Node designs for single link crossing ring ........................................... 40
Figure 4.5 Node designs for topologies with bidirectional links ............................. 41
Figure 4.6 Total offered load for eight node ring ..................................................... 44
Figure 4.7 Total offered load for eight node ring with one cross link ...................... 45
Figure 4.8 Comparison chart of average delay for various topologies ..................... 48
Figure 4.9 Comparison chart of average delay for more topologies ....................... 49
Figure 4.10 Comparison chart of average delay for more topologies and higher capacity bound ........................................................................................................ 50
Figure 4.11 Comparison of average delay by topology with aggregate capacity to load ratio=1.05 ........................................................................................................ 51
Figure 4.12 Comparison of average delay by topology with aggregate capacity to load ratio=1.15 ................................................................................................. 52
Figure 4.13 Comparison of average delay by topology with aggregate capacity to load ratio=1.25 ................................................................................................. 53
Figure 4.14 Promising ten node topologies to consider for link switching in a UWB ring 54
Figure 4.15 Two special dual queue nodes are needed to implement the dual switching ring ............................................................................................................. 54
Figure 4.16 Block diagram of single input, single output UWB nodes assumed for delay modeling of the ten node ring network ...................................................... 55
Figure 4.17 Block diagram of Node 3 ........................................................................ 56
Figure 4.18 The states of network operation used for calculating average delay ....... 57
Figure 4.19 Running simulations for 8 vs. 16 Node 3 queue switches ....................... 62
Figure 4.20 Delay plots can be hard to interpret if optimal capacities are allowed to vary for each acquisition time ................................................................. 64
Figure 4.21 Comparison of average delay versus excess capacity factor for mathematical and simulation models of the UWB packet forwarding ring network ............... 66
Figure 4.22 Ten node network delay versus threshold value-short acquisition times .... 67
Figure 4.23 Ten node network delay versus capacity excess at longer acquisition times. 68
Figure 4.24 Network delay versus acquisition time. ................................. 69
Figure 4.25 Effect of switching threshold on delay for 0 to 5 μs acquisition delay ... 70
Figure 4.26 Effect of capacity excess on delay for 0 to 5 μs acquisition delay ....... 71
Figure 4.27 Network delay versus acquisition time. ................................. 72
Figure 4.28 Effect of switching threshold on delay for 0 to 50 μs acquisition delay ... 73
Figure 4.29 Effect of capacity excess on delay for 0 to 50 μs acquisition delay ...... 74
Figure 4.30 Network delay versus acquisition times between 0 and 500 μs .......... 75
Figure 4.31 Effect of switching threshold capacity excess on delay for 0 to 500 μs acquisition delay .......................................................... 76
Figure 4.32 Effect of switching threshold on delay for 0 to 500 μs acquisition delay ... 77

Figure 5.1 WTRP requires several distinct data transmissions to allow a node to join the ring................................................................. 81
Figure 5.2 Bluetooth also requires several distinct data transmissions to allow a node to join a piconet ......................................................... 82
Figure 5.3 SDD considers long acquisition times in attempting to provide a joining protocol specifically for UWB networks ................................. 83
Figure 5.4 UWB node with joining manager to listen for joining nodes .............. 86
Figure 5.5 Proposed UWB joining protocol packet exchange diagram ............... 87
Figure 5.6 Network view of proposed UWB joining protocol .......................... 87
Figure 5.7 Proposed UWB joining protocol for first two nodes ..................... 88
Figure 5.8 Network view of proposed UWB joining protocol .......................... 88
Figure 5.9 WTRP joining sequence for discovery overhead calculation ............... 90
Figure 5.10 SDD joining sequence for discovery overhead calculation ............... 91
Figure 5.11 UWBJP joining sequence for discovery overhead calculation .......... 92
Figure 5.12 UWBJP outperforms WTRP and SDD using the long or short discovery intervals ................................................................. 93
Figure 5.13 Node 2 leaves the network with notification for WTRP ........................................ 94
Figure 5.14 Masters or slaves in Bluetooth piconets can request detachment .................. 95
Figure 5.15 SDD detects failed or out of area nodes by multiple failures to respond to RTS packets .................................................................................................................. 96
Figure 5.16 A simple protocol supporting nodes leaving a UWB network ..................... 98
Figure 5.17 A simple prepare and acknowledge protocol supports nodes leaving a UWB network ........................................................................................................................................ 98
Figure 5.18 When Node 2 disappears without notice, the proposed UWB protocol can easily re-form the ring .................................................................................................................. 98
Figure 5.19 When Node 2 disappears without notice, Node 3 discovers the flaw and tells Node 1 to set Node 3 as successor .............................................................. 99
Figure 5.20 We need a successor timer to continue when two nodes leave the network without notification ................................................................. 99
Figure 5.21 When two consecutive nodes leave without notification, supervisory timers can support ring recovery ................................................................. 100
Figure 5.22 Even when half of the nodes leave without notification, supervisory timers can support ring recovery ................................................................. 101
Chapter 1

Introduction

Networks using ultra-wideband wireless (UWB) as the physical layer show promise for application as personal area networks. Personal area networks may be established at work or at home to connect devices such as personal digital assistants, computer peripherals, video sources, and video displays. UWB can provide high capacity wireless links over short distances and allow variable capacity links to be formed between nodes. However, there is an upper bound on the aggregate capacity of all links that transmit within specified interference boundaries.

Certain characteristics of UWB wireless links give rise to two properties not seen with other wireless technologies:

- long synchronization times for establishing links between nodes and
- the ability to change individual link capacities by choosing pseudo-random (PN) codes of different lengths for channel coding.

The number of incoming and outgoing links for a single UWB node is driven by cost and technology and affects the topologies available for a network formed from UWB nodes. Whether composed of nodes with a single incoming and a single outgoing UWB link or M incoming and outgoing links, UWB networks can be designed to minimize average network delay. UWB networks of different scales and node types also have unique management problems requiring specific management protocols in order to function efficiently.
1.1 A Brief History of Ultra-wideband Wireless

Since the Federal Communication Commission (FCC) issued a Report and Order \cite{1} allocating 7,500 MHz of spectrum between 3.1 and 10.6 GHz for unlicensed use by ultra-wideband wireless devices, the landscape of UWB communications research has greatly changed. The primary drivers of the changes were the stringent spectral masks imposed for indoor and outdoor hand-held UWB communications devices.

Much previous research had focused on UWB as an enabling technology for covert, high capacity, multi-user communication systems \cite{2}. The FCC power and spectral density limitations pushed UWB squarely indoors, cutting its potential range to not more than about 10 meters at high data rates \cite{3}. Previous UWB research was focused on multi-user systems with higher transmit power and reasonably long ranges. These proposed systems used very short (\sim one ns) duration radio frequency pulses to transmit data bits. Coding techniques to channelize data included use of pseudo-random (PN) \cite{4}, orthogonal \cite{5}, or chaotic \cite{6} codes. These systems achieved the wide bandwidth characteristic of UWB by the use of the extremely short pulses.

More recent UWB schemes include more traditional ways to use the 7,500 MHz of unlicensed spectrum defined by the FCC. Multi-band schemes defining several channels of \geq 500 MHz have been proposed. In addition, techniques such as frequency hopping, spectral keying, and orthogonal frequency division multiplexing have been discussed \cite{7}.

As with most wireless data communications systems, however, lower data rates are possible at longer ranges. UWB is now viewed as a strong contender for providing high data rate wireless connectivity for personal area networks in the home or office environment, for example, replacing cables carrying high definition television (HDTV) signals or allowing high speed wireless access to printers \cite{3}. The IEEE 802.15 Wireless Personal Area Network (WPAN) working group has two committees looking at UWB as the physical layer for both high data rate and low power, low data rate applications \cite{8}.

UWB systems are not confined to the minds and simulators of academics - Pulse Link recently developed a chip set capable of 1.35 Gbps raw data rate over short distances using a pulsed UWB approach \cite{9}. Intel, along with a number of other high powered communications technology companies belong to Multiband OFDM Alliance to promote orthogonal frequency division multiplexing (OFDM) as the best alternative for the 802.15.3a UWB PHY and MAC specification \cite{3}.
Prior to the new FCC limits, the communication systems using UWB were developed for the U.S. Government. Because of their extremely low power spectral density, UWB transmissions are very difficult to detect - making UWB ideal for covert communications. A UWB ad hoc network of eight nodes was built and tested at Fort Campbell, KY. Called DRACO, this system was capable of sending unencrypted data at 1.544 Mbps at a range in excess of 1 km [10].

1.2 Status of UWB Standards

UWB technologies were considered as standards in the IEEE 802.15 working group for Wireless Personal Area Networks. Standard IEEE 802.15.3-2003 is a MAC and PHY standard for high-rate (11 to 55 Mb/s) Wireless Personal Area Networks.

Task group 3a., formed to develop standards for a UWB PHY for WPANs winnowed 23 UWB PHY proposals down to two competing proposals backed by two separate industry alliances before reaching a deadlock. The final two UWB PHY proposals were Multi-Band Orthogonal Frequency Division Multiplexing (MB-OFDM) supported by the WiMedia Alliance and Direct Sequence UWB (DS-UWB), supported by the UWB Forum.

In January 2006, unable to reach a consensus to move forward with a joint proposal, the IEEE 802.15.3a task group members voted to withdraw the December 2002 project authorization request that initiated the development of high data rate UWB standards, allowing the competing industry groups to develop products and let the market decide which was better.

Task group 4 (Low Rate WPAN) also considers UWB PHY as a standard. IEEE 802.15.4-2003 (Low Rate WPAN) describes standards for low complexity wireless PAN with low data rate but with battery life, extending to months or years.

Two PHY optional baselines, one UWB and one Chirp Spread Spectrum, are covered in IEEE 802.15.4a , which was published in March 2005. The UWB PHY is UWB Pulse Radio based on continuous pulsed UWB technology and can provide high precision ranging, as well as communications.

1.3 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 discusses UWB characteristics that are useful for networks and presents the problem areas studied in this thesis.
Chapter 3 presents our analysis of methods to minimize delay in networks of simple (single input and output link) UWB nodes. Chapter 4 discusses delay improvements that can be achieved by adding single direction links to a UWB ring network. Chapter 5 discusses challenges to managing networks based on a UWB physical layer. Chapter 6 concludes and describes future work to characterize and evaluate the application of UWB to wireless networks.
Chapter 2

Ultra-wideband Networking

2.1 UWB Characteristics and Usefulness for Wireless Networking

UWB signals have several characteristics that suit it for use as the physical layer of wireless networks. UWB easily penetrates walls and has inherent immunity to multi-path interference due to the wide bandwidth of the transmitted signals. These characteristics make UWB ideal for indoor use. Combined with UWB’s ability to handle high capacity data rates, these properties make UWB ideal for self-organizing personal area networks in homes or offices. The primary UWB characteristics that affect network operation are long acquisition times between nodes and the ability to vary link data rates based on link loading.

Due to the very short pulse lengths used in pulsed radio UWB communications, the synchronization time between the receiver and transmitter can be long - on the order of milliseconds [11], [12]. Finding ways to shorten this synchronization time is an ongoing research area [13], [14], [15].

Figure 2.1: Two options to transfer information from a source S to two different destinations $D_1$ and $D_2$
Due to the relatively long acquisition times, it might be impractical for a source to frequently change between two destinations. Figure 2.1 depicts two different strategies for networking a source $S$ and two destinations $D_1$ and $D_2$. In Fig. 2.1(a), the traditional approach is depicted with the source sending packets to both destinations, alternating transmissions between them. In Fig. 2.1(b), the source directs all traffic to the first destination which, in turn, forwards the traffic to the second destination. The strategy in Fig. 2.1(a) is commonly employed in wireless networks. However, in UWB, the source has to synchronize with each destination before it can start the data transmission, a potentially very expensive (in terms of efficiency) proposition. Therefore, it is very likely that in UWB networks the scheme depicted in Fig. 2.1(b) will be far more efficient than the scheme in Fig. 2.1(a).

If $M$ transceivers are used for each node, $M$ distinct links can be maintained and used without the overhead of re-synchronization; however, it is likely that such a node will be significantly more complex and potentially expensive. In this thesis, we consider both the simple case where only one transceiver is available for each node and the more complex situation where network nodes have $M$ incoming and outgoing links.

Proposed UWB systems transmit tens to hundreds of very short pulses per information bit [4]. By changing the number of pulses per bit, the transmission rate can be varied. For example, for one ns pulses, using 100 pulses per bit results in a data rate of 10 Mbps, while using 10,000 pulses per bit results in 100 kbps. This property of UWB allows the allocation of different data rates to different nodes in a UWB network.

2.2 The Problems Studied in this Thesis

A self-organizing personal area network consists of nodes that must not only generate and receive node-specific traffic, but may be required to forward traffic from other nodes. Some of the nodes in a personal area network may be portable, implying possible mobility induced changes to the established network topology. Channelization of the available UWB capacity and the limited range of the UWB nodes create a challenging routing environment.

This thesis addresses the following problems related to networks composed of simple UWB nodes:

- Given a traffic matrix specifying the average packet delivery rate between nodes and
an aggregate capacity bound for all links, determine a topology for the network and capacities for each link to minimize average packet delay.

• Given a traffic matrix specifying the average packet delivery rate between nodes, determine the effect of the acquisition time to establish a new network link on average packet delay and when packet switching operation performs better than packet forwarding operation.

• Given a ring network of UWB nodes, determine whether adding additional single direction links can improve delay performance for the network.

• Given a ring network of UWB nodes, determine the network management issues for the network and propose protocols to efficiently address these issues.
Chapter 3

Minimizing Delay in UWB Ring Networks

Whether a UWB system uses frequency division multiple access (FDMA) or code division multiple access (CDMA) to provide separate data channels, the aggregate capacity of nodes within the wireless range of one another has an upper bound, typically much lower than the Shannon limit [16]. The 7,500 MHz of bandwidth allocated by the FCC limits the number of multi-band UWB channels. Likewise, if pseudo-random number (PN) codes are used to create separate channels, longer codes create more channels; but, each channel then has a lower bit rate. This feature is not unique to UWB wireless networks.

One key performance parameter for networks is end-to-end packet delay. We will analyze the performance of the compact network of simple UWB nodes with the aim of minimizing average packet delay through the network.

For our discussions, we assume that all nodes of the network are effectively co-located (a realistic assumption given the relatively reduced range allowed by FCC); and, hence, all nodes interfere with each other. For these networks, we have the flexibility to allocate different data rates to different nodes, but the total capacity of all of the links in the network is limited. We will denote the capacity limit by $B$. We assume perfect multiple user access, so that each node may transmit and receive simultaneously with no interference.

We consider two methods for minimizing average network delay. In the first, we identify a method for choosing link capacities assuming a given load matrix, $T$, and the aggregate capacity constraint, $B$. In the second, we consider minimizing average packet delay by choosing the ring order for the complex ring based on the load matrix, $T$. 


3.1 Topology Effects on Delay

We begin by defining the network of simple nodes that will be considered. We define simple UWB nodes as nodes having only one input and one output link. We define a network of \( R \) stationary wireless nodes, each of which generates its own traffic and routes traffic received from other nodes. Each node \( i \) has one incoming wireless link and one outgoing wireless link. Multiple input, multiple output nodes are certainly possible; however, for simple personal area networks where link redundancy is not a major concern, a single input single output assumption is reasonable and cost effective.

Each node generates packets exponentially distributed in time with an average packet generation rate of \( G_i \) packets per second. Each node has a single packet queue of infinite size and serves packets such that the inter-service intervals are exponentially distributed with an average service rate of \( c_i \) packets per second, which is synonymous with the capacity of the outgoing link from node \( i \). This assumption allows us to easily apply queuing theory for \( M/M/1 \) queues to the problem. Note that this assumption implies that packet sizes are exponentially distributed - constant bit rate transmitters send packets with exponentially distributed packet lengths resulting in exponentially distributed departure times per packet.

![Figure 3.1: Block diagram of a network node](image)

Arriving packets with destination \( i \) are received at an average rate of \( RX_i \). Finally, the average traffic on the incoming link to node \( i \) is given by \( P_i \). Arriving packets destined for other nodes are immediately queued for forwarding. A block diagram of a node from this network is depicted in Figure 3.1.
We define the traffic matrix

$$T = \begin{bmatrix} 0 & \lambda_{1,2} & \lambda_{1,3} & \ldots & \lambda_{1,R} \\ \lambda_{2,1} & 0 & \lambda_{2,3} & \ldots & \lambda_{2,R} \\ \lambda_{3,1} & \lambda_{3,2} & 0 & \ldots & \lambda_{3,R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{R,1} & \lambda_{R,2} & \lambda_{R,3} & \ldots & 0 \end{bmatrix}$$

(3.1)

where each $\lambda_{i,j}$ represents average offered load in packets or bits per second between node $i$ and node $j$ in the network.

Let $c_i$ represent the capacity of the outgoing link for node $i$. Let $B$ represent an upper bound on the aggregate capacity of the network, such that for all connections of capacity $c_i$ between nodes,

$$\sum_{i=1}^{R} c_i \leq B.$$ 

(3.2)

Recall that this constraint only arises in this form when all UWB nodes are in range of each other as is the case for this compact network and when all outgoing links are allowed to transmit simultaneously, i.e., there is no time division multiplexing of node signaling.

In general, this upper bound $B$ depends on channel characteristics and multi-user interference, which are not considered in this thesis. Our goal is to find a capacity matrix $c$ that will minimize the average network transit delay for the traffic specified in $T$ where $c_i$ is the capacity of the outgoing link from node $i$.

$$c = \begin{bmatrix} c_1 & c_2 & c_3 & \ldots & c_R \end{bmatrix}.$$ 

(3.3)

We will show that the nodes in the network must form a ring. Because the traffic matrix requires traffic to be delivered to every node $i$ in the network, no two nodes may connect with their outgoing links to the same node. As shown in Figure 3.2, if they did so, one of the nodes would be unable to receive its traffic.

No matter how the links are arranged, if two nodes connect their outgoing links to the same node, at least one node cannot receive traffic. The only topology that can be constructed so that all traffic in the traffic matrix $T$ can be delivered is a ring. The choice of exactly how the nodes should form the ring for optimal performance will be considered in Section 3.3.
Choosing Link Capacities to Minimize Delay

Examining Figure 3.1, it is clear that the capacity of the outgoing link must be greater than the arrival rate into the service queue, (or the queue will be unstable):

\[ c_i \geq G_i + P_i - RX_i. \]  \hspace{1cm} (3.4)

We can easily write equations for \( G_i \) and \( RX_i \) in terms of the traffic matrix \( T \):

\[ G_i = \sum_{k=1}^{R} \lambda_{i,k}, \]  \hspace{1cm} (3.5)

and

\[ RX_i = \sum_{m=1}^{R} \lambda_{m,i}. \]  \hspace{1cm} (3.6)

Figure 3.3 shows \( R \) nodes arranged in a ring. We will determine a value for \( P_i \) (the average number of packets per second on the incoming link to node \( i \) based on the traffic matrix \( T \). In a ring, the average flow \( P_i \) is the offered load from node \( i - 1 \) to node \( i \). To elaborate, \( P_i \) is the sum of all traffic from node \( i - 1 \) to any other node plus the sum of all traffic into node \( i \), making sure not to count the flow from node \( i - 1 \) to node \( i \) twice. Added to this is the sum of all traffic from nodes \( 2, 1, \ldots, i + 2 \), to nodes \( i + 1, \ldots, 2 \) that must traverse across the link between nodes \( i - 1 \) and \( i \), to reach the destination node. We are only concerned with the traffic to and from these nodes because we have already accounted for all traffic to and from nodes \( i - 1 \) and \( i \) in the previous step. Mathematically, this translates to the following equation:

\[ P_i = \sum_{h=1}^{R} \lambda_{i-1,h} + \sum_{k=1}^{R} \lambda_{k,i} + \sum_{r=i-2}^{i+2} \sum_{s=r-1}^{i+1} \lambda_{r,s}. \]  \hspace{1cm} (3.7)

Figure 3.2: If the topology is not a ring, some of the nodes cannot receive any traffic.
Figure 3.3: Traffic on each link is the sum of offered loads that must traverse that link.

We can, thus, calculate $P_i$ simply by knowing the $\lambda_{i,j}$ values in the traffic matrix $T$. We will now use the equations from this section to derive an expression for the capacity matrix $c$ that minimizes average network transit delay.

We define the average network transit delay as follows:

$$T_D(c) = \sum_{i=1}^{R} w_i T_i(c).$$  \hfill (3.8)

In this expression, $R$ is the total number of nodes in the network, $w_i$ is a weighting factor (it can be selected to be different for each node to give nodes preferential treatment), $T_i(c)$ is the average delay in seconds on link $i, j$ caused by transmission, processing, queuing, media access, and propagation delays, and $c$ is the capacity matrix defined in (3.3).

The delay on link $i, j$ is the sum of the transmission, queuing, processing, media access, and propagation delays from node $i$ to node $j$. That is,

$$T_i(c) = T_{\text{transmission } i,j} + T_{\text{queuing } i,j} + T_{\text{processing } i,j} + T_{\text{propagation } i,j} + T_{\text{media access } i,j}.$$  \hfill (3.9)

We assume that processing and propagation delays are small compared to the others and can be neglected. Note that the long synchronization times typically associated with UWB transmissions do not have an impact on the considered problem as the ring topology is fixed after network initialization. We also assume that media access time of the media access control (MAC) protocol is negligible (a realistic assumption given the CDMA character of UWB transmissions: the nodes do not have to wait for other nodes to complete their
transmissions before starting their own). Thus, the delay at node \( i \) is

\[
T_i(c) = T_{\text{transmission } i,j} + T_{\text{queuing } i,j}.
\] (3.10)

Finally, the queuing and transmission delays can be determined assuming, for simplicity, \( M/M/1 \) queues. We recognize that assumptions to justify the use of the \( M/M/1 \) queuing formulas may not apply here, but use them to achieve a closed-form solution to a minimization problem, rather than to calculate some explicit value for delay. In addition, packet framing issues are not addressed here in order to achieve a closed-form solution for delay. As usual in delay studies, frame error rates are assumed to be zero (and, hence, frame error recovery techniques are not taken into account). No packet fragmentation is assumed to occur.

Under these conditions, we can write the delay equation using the \( M/M/1 \) system delay formula [17]:

\[
T_i(c) = \frac{\frac{1}{c_i}}{1 - \frac{AR_i}{c_i}}
\] (3.11)

where \( AR_i \) is the arrival rate of packets into the service queue at node \( i \) and \( c_i \) is the service rate at node \( i \) (equal to the capacity \( c_i \) of the outgoing link). The arrival rate of packets into the queue is the sum of packets arriving via the incoming links and those generated within node \( i \), itself, less those packets that are received by node \( i \). Here ”received” means taken out of the service queue and placed in a processing queue within the node. If we let \( RX_i \) be the rate at which packets destined for node \( i \) are received, and \( P_i \) be the average flow rate of packets on the incoming link to node \( i \), we can write that the arrival rate to the node \( i \) queue \( AR_i \) is given by:

\[
AR_i = P_i + G_i - RX_i.
\] (3.12)

Using these identities, we can write the following equation for the delay at node \( i \):

\[
T_i(c) = \frac{\frac{1}{c_i}}{1 - \frac{P_i + G_i - RX_i}{c_i}}
\] (3.13)

where \( c_i \) is the outgoing link capacity for node \( i \) and \( RX_i \) is the rate at which packets whose destination is node \( i \) are received by the node. After simplifications, we obtain:

\[
T_i(c) = \frac{1}{c_i + RX_i - P_i - G_i}.
\] (3.14)
So, we can now write the equation for average network transit delay as:

\[ T_D(c_i) = \sum_{i=1}^{R} w_i T_i = \sum_{i=1}^{R} \frac{w_i}{c_i + RX_i - P_i - G_i}. \quad (3.15) \]

Now, we state the optimization problem we are trying to solve, namely, to minimize the average network transit delay \( T_D \) subject to the constraint (3.2).

We introduce the Lagrange multiplier \( \beta \) and the Lagrangian function \( L \) that we will use to determine the minimum values for \( c_i \):

\[ L = \sum_{i=1}^{R} (w_i T_i(c) + \beta c_i) = \sum_{i=1}^{R} \left( \frac{w_i}{c_i + RX_i - P_i - G_i} + \beta c_i \right). \quad (3.16) \]

Taking the partial derivatives with respect to \( c_i \) and setting the result equal to zero yields:

\[ \frac{\partial L}{\partial c_i} = \beta - \frac{w_i}{(c_i + RX_i - P_i - G_i)^2} = 0. \quad (3.17) \]

Solving for \( c_i \), we obtain the following:

\[ c_i = \sqrt{\frac{w_i}{\beta} - RX_i + P_i + G_i}. \quad (3.18) \]

The total capacity constraint (3.2) becomes:

\[ \sum_{i=1}^{R} c_i = \sum_{i=1}^{R} \left( \sqrt{\frac{w_i}{\beta} - RX_i + P_i + G_i} \right) = B. \quad (3.19) \]

We solve this equation for \( \sqrt{\frac{1}{\beta}} \) and substitute in (3.18), while considering the queue arrival rate (3.12). We obtain:

\[ c_i = \left( \frac{\sqrt{w_i}}{\sum_{i=1}^{R} \sqrt{w_i}} \right) (B - \sum_{i=1}^{R} P_i) + AR_i. \quad (3.20) \]

This result in (3.20) states that to minimize average network transit delays, any excess capacity must be distributed among the network links in proportion to the square
roots of the weights assigned to each link:

\[ c_i = \left( \frac{\sqrt{w_i}}{\sum_{i=1}^{R} \sqrt{w_i}} \right) \left( B - \sum_{i=1}^{R} \bar{P}_i \right) + \frac{AR_i}{\text{arrival rate}}. \]  

(3.21)

This section developed a method to calculate optimal network link capacities for UWB nodes arranged in a ring. We obtained the result by constructing a network model, developing an equation specifying average network transit delay, and applying queuing theory delay equations for M/M/1 queues. We used the Lagrangian function to find link capacities to minimize the delay.

While not specifically applied to mobile nodes, this method can be used to specify link capacities required in an ad hoc UWB network with a ring topology to minimize average network transit delay.

We next turn our attention to examining the optimum order of nodes in the ring topology to minimize delay based on the offered load matrix, \( T \).

### 3.3 Effect of Ring Order on Delay

Previously, we defined the average network transit delay as follows:

\[ T_D(c) = \sum_{i=1}^{R} w_i T_i(c). \]  

(3.22)

In this expression, \( R \) is the total number of nodes in the network, \( w_i \) is a weighting factor (it can be selected to be different for each node to give nodes preferential treatment), \( T_i(c) \) is the average delay in seconds on link \( i \), caused by transmission, processing, queuing, media access, and propagation delays, and \( c \) is the capacity matrix defined above.

We neglected propagation, processing, and media access delays and used the \( M/M/1 \) queuing formula for system delay to obtain an expression for average network transit delay, \( T_D \).

We wrote the following equation for the delay at node \( i \):

\[ T_i(c_i) = \frac{w_i}{c_i - AR_i}. \]  

(3.23)

By plugging the optimum value for \( c_i \), called \( c_{opt} \) here for convenience, into this equation, we obtain
\[
T_i(c_{opt}) = \frac{\sqrt{w_i} \sum_{k=1}^{R} \sqrt{w_k}}{(B - \sum_{j=1}^{R} P_j)},
\]
(3.24)
and
\[
T_D(c_{opt}) = \frac{\sum_{j=1}^{R} \sqrt{w_j} \sum_{i=1}^{R} \sqrt{w_i}}{(B - \sum_{k=1}^{R} P_k)}.
\]
(3.25)

Because all terms in this equation are constants, except one, the only way to minimize \(T_D c_{opt}\) and is to choose the order of the ring such that we minimize the sum of the offered loads.

The sum of all the \(P_i\) offered load terms turns out to be easy to conceptualize. Because traffic originating from a node, say, three hops away from its destination must travel through those three hops, this traffic forms part of the average input traffic for three nodes. Thus, when summing the input traffic for the entire ring, the three hop traffic is included three times. This is true for all hop counts. Single hop traffic appears only once in the sum, four hop traffic appears four times, and so on. Knowing this we can write
\[
\sum_{k=1}^{R} P_k = \sum_{\text{ring}} \text{one hop traffic} + 2 \sum_{\text{ring}} \text{two hop traffic} + \ldots + (R-1) \sum_{\text{ring}} (R-1) \text{hop traffic}.
\]
(3.26)

This has an interesting interpretation when viewed in the context of the offered load matrix, \(T\). As shown in the figure for five nodes, the diagonals of \(T\) identify one hop, two hop, three hop, and four hop traffic. Finding the best ring order equates to swapping rows and columns of \(T\) to minimize the weighted sums of the diagonals of \(T\).

![Figure 3.4: The sum of the offered loads in the ring is the weighted sum of the diagonals of the traffic matrix](image)

The first heuristic approach we tried for this was to try to minimize delay by maximizing the one hop traffic, reasoning that this would prevent the large offered loads.
from having to traverse multiple hops. However, a quick implementation of a four node ring showed counter-examples where maximizing the one hop traffic did not minimize $\sum_{k=1}^{R} P_k$.

Further study of this problem showed that it is actually a variant of the well-known traveling salesman problem (TSP) [18], [19].

For example, for $R$ given cities, the $T$ matrix could represent the cost to travel between each city. The classic TSP asks us to find the shortest tour through the cities passing through each one only once and returning to the starting city. Suppose that once the minimal tour is found, you rearrange the row and columns of $T$ to form a new matrix, $T'$ such that the nodes are in numerical order for the optimal tour, $1,\ldots,R$.

In this case, the sum of the tour cost is given by the sum of the one hop diagonal of $T'$ - $(\lambda_{1,2}, \lambda_{2,3}, \ldots, \lambda_{R,1})$. Note that this sum must be the least possible sum for any rearrangement of the rows and columns of $T$, otherwise the new arrangement would be a shorter tour than the optimal tour.

Our case is a more complicated version of the TSP. Our problem requires the minimization of the weighted sum of all the diagonals of $T$. This can be illustrated in a five node example. As shown in the figure below, minimizing $\sum_{k=1}^{5} P_k$ requires us to find the optimum ordering of nodes so that the path once around the one hop paths, twice around the two hop paths, three times around the three hop paths, and four times around the four hop paths is of minimum length.

![Figure 3.5: The several tours that must be optimized for a five node ring.](image-url)
This problem of choosing the ring order to minimize delay is similar to routing and wavelength assignment problem in all optical networks, which is known to be in the set of \textit{NP-Complete problems} \cite{20}. It is similar to multi-commodity network flow problems, many of which are provably \textit{NP complete} \cite{21} and to provisioning of calls in circuit switched various types of circuit switched networks. Off line permutation scheduling on ring networks and allocating bandwidth for call connections in asynchronous transfer mode (ATM) ring networks were also shown to be \textit{NP complete} by transformation from the vertex coloring problem for circular arc graphs, which is known to be NP complete \cite{22}, \cite{23}. The exact problem of choosing the ring order to minimize delay is conjectured to be NP complete in \cite{24} and a proof by restriction from the traveling salesman problem is sketched in \cite{25} for minimizing the power costs of UWB nodes ring networks. However, try as we might, we were unable to provide a rigorous proof that finding the minimum delay ring permutation as described herein is \textit{NP complete}.

Because this problem is analytically intractable, we investigated several heuristic approaches for methods for calculating the minimum delay ring order. Four different algorithms were developed to find the optimal ring order. Implemented in MATLAB, these algorithms were named: \textit{buildring}, \textit{searchlarge}, \textit{searchsmall}, and \textit{findstrings}. For a specific traffic matrix, \(T\), the \textit{buildring} algorithm builds the best ring order permutation one node at a time using a variation of the cheapest insertion heuristic for TSP. \cite{26}. The algorithm starts with two nodes and inserts the remaining nodes randomly one at a time by checking for the "best" place to insert the new node, where the "best" place is where the weighted sum of the offered loads, \(P_i\), is minimized.

For the traffic matrix, \(T\), the \textit{searchlarge} algorithm uses a greedy heuristic search starting with the largest N elements in the \(T\) matrix and then trying to complete different R hop paths (where R is the network size) that maximize the traffic on the single hop paths. The algorithm completes a specified number of trials and then compares all to find the permutation that minimizes the offered load function. The \textit{searchsmall} algorithm works in a similar fashion, but starts with the smallest N elements in the traffic matrix and creates trial paths to minimize traffic on the paths that must go completely around the ring.

Finally, the \textit{findstrings} algorithm uses a heuristic search to find "good" ring order permutations. This heuristic is based on building a ring order that puts large elements on paths containing few hops between source and destination and places small elements on paths requiring more hops. The output of the program is several permuted orders of the
nodes as chosen using the algorithm. The resulting ring permutations are compared using
the offered load function to find the best one.

As seen in Figure 3.6, the four ring ordering heuristics were used to find the best
permutations for networks between five and nine nodes. The results were compared to
an algorithm that simply found all possible ring permutations and selected the best one
based on minimizing the offered load function. For small networks, the buildring algorithm
performed best, finding ring permutations with offered loads on average less than one percent
different than the optimal permutation. The other three algorithms performed about the
same with steadily increasing 'error' as the number of nodes increased.

As seen in Figure 3.7, the four ring ordering heuristics were also used to find the
best permutations for networks between ten and 200 nodes. The results were compared
only to the performance of the buildring algorithm because for large numbers of nodes,
the time required for finding all possible ring permutations and selecting the best grows
exponentially. For these larger networks, the buildring algorithm still performed best, but
there is no way of knowing how far the results of any of these algorithms were from the
optimal solution.

3.4 Effect of Node Acquisition Times on Delay

As was previously discussed, in a network of simple UWB nodes, we require a small
acquisition time for a node to stop communicating with one node and begin communicating
with another. This section describes conceptual, mathematical, and simulation models for
a simple data network. The nodes in the model use ultra-wideband (UWB) wireless as
the physical layer, requiring relatively long node acquisition times when links are dropped
and reestablished. Only three nodes are considered and each has only one input and one
output link. We use the models to study average packet delay for two schemes of network
operation. In the first model, the three nodes are connected in tandem and packets destined
for the last node are simply forwarded through the second node. In the second model, the
first node drops synchronization with the second node and establishes a link with the third
node to send any packet bound for the third node. While this is a simple model in terms
of the number of nodes, we expect this model may be readily extensible to larger networks
of simple UWB nodes because we can partition the larger network into smaller networks of
nodes when examining the behavior of the larger network.
<table>
<thead>
<tr>
<th>N=R</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>numRuns=30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum(P_i) values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bestPerm</td>
<td>2303</td>
<td>3866.2</td>
<td>6304.1</td>
<td>9928.6</td>
<td>14117</td>
</tr>
<tr>
<td>buildRing</td>
<td>2310.3</td>
<td>3872.2</td>
<td>6324.1</td>
<td>10015</td>
<td>14216</td>
</tr>
<tr>
<td>searchLarge</td>
<td>2354.7</td>
<td>3966.9</td>
<td>6562.1</td>
<td>10452</td>
<td>14940</td>
</tr>
<tr>
<td>searchSmall</td>
<td>2358.3</td>
<td>4063.3</td>
<td>6594.3</td>
<td>10474</td>
<td>14898</td>
</tr>
<tr>
<td>findstrings</td>
<td>2335.9</td>
<td>3953.3</td>
<td>6566.9</td>
<td>10374</td>
<td>14862</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ratio of Sum(P_i) value to that of best permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bestPerm</td>
</tr>
<tr>
<td>buildRing</td>
</tr>
<tr>
<td>searchLarge</td>
</tr>
<tr>
<td>searchSmall</td>
</tr>
<tr>
<td>findstrings</td>
</tr>
<tr>
<td>5 6 7 8 9</td>
</tr>
<tr>
<td>1 1 1 1 1</td>
</tr>
<tr>
<td>1.00317 1.001552 1.003173 1.008702 1.007013</td>
</tr>
<tr>
<td>1.022449 1.026046 1.040926 1.052716 1.058299</td>
</tr>
<tr>
<td>1.024012 1.05098 1.046034 1.054932 1.055323</td>
</tr>
<tr>
<td>1.014286 1.022529 1.041687 1.04486 1.052773</td>
</tr>
</tbody>
</table>

Figure 3.6: Heuristic methods compared well with exhaustive search
### Comparison of heuristics for choosing ring permutation

![Comparison of heuristics for choosing ring permutation](image)

**Figure 3.7:** Performance of heuristic methods difficult to evaluate for large networks
The short pulse lengths used in UWB communications require relatively long synchronization times between the receiver and transmitter - on the order of microseconds [27] to milliseconds [16]. Finding ways to shorten this synchronization time is an ongoing research area [28] and [29].

Because of the relatively long acquisition times, it is not practical for a high data rate source to frequently switch between two destinations, as was discussed in section 2.1. If \( M \) transceivers are used for each node, \( M \) distinct links can be maintained and used without the overhead of re-synchronization. However, such a node is significantly more complex and potentially expensive. In this chapter, we focus nodes with a single transceiver. Hence, each network node can send to a single destination and receive from a single source. In previous work [30], we showed that minimum delay is achieved by forming nodes into rings and apportioning the aggregate capacity available to each link in proportion to offered load.

### 3.4.1 Three Node Network Conceptual Model

The conceptual models of the operation of the three node network with packet forwarding and with link switching can be best explained through diagrams with explanatory remarks.

Understanding the operation of the packet forwarding model is important because packet forwarding is common and useful in ad hoc networks, providing a convenient alternative to requiring nodes to drop and reacquire each other to send data packets. Figure 3.8 shows the conceptual model for a three node network operating solely in a packet forwarding mode. Node 1 sends all traffic to Node 2. Node 2 keeps its own traffic while forwarding traffic bound for Node 3, which includes packets from Nodes 1 and 2. In this network set up, the existing wireless links are used - dropping of links and reacquisition is not performed because link switching is not required.

![Three node packet forwarding diagram](image)

**Figure 3.8: Three node packet forwarding diagram**

We define a network of three wireless nodes, each of which is always within wireless...
range of the others. Each node generates its own traffic and routes traffic received from other nodes. Each node $i$ has one incoming wireless link and one outgoing wireless link.

Nodes 1 and 2 generate packets exponentially distributed in time with an average packet generation rate of $G_i$ packets per second, which is defined as

$$G_1 = \lambda_{12} + \lambda_{13}$$

$$G_2 = \lambda_{23}. \quad (3.27)$$

$$G_3 = \lambda_{23}. \quad (3.28)$$

Nodes 1 and 2 have packet queues of infinite size and serve packets such that the inter-service intervals are exponentially distributed with an average service rate of $c_i$ packets per second.

Arriving packets with destination $i$ are received at an average rate of $RX_i$. Finally, the average traffic on the incoming link to node $i$ is given by $P_i$. Arriving packets destined for other nodes are immediately queued for forwarding. Node 3 neither generates nor queues packets, but simply receives packets at an average rate of $RX_3$.

As discussed, the alternative to simply forwarding packets from node 1 to node 3 via node 2 is to allow node 1 to drop its link with node 2, acquire and establish a new link with node 3, and forward packets from Node 1 directly to Node 3. The network set-up is provided in Figure 3.9 below.

![Figure 3.9: Three node network switching diagram](image)

Node 1 contains two queues served by a single server with exponentially distributed service times. The average rate of packet service is $c_1$, with the server being shared between the two queues. Each queue is fed by packet generators with exponentially distributed generation times with averages $\lambda_{12}$ and $\lambda_{13}$, where $\lambda_{ij}$ signifies the average rate, in packets per second, of traffic originating at node $i$ and destined for node $j$. 

23
Node 2 contains only a single queue and server, with packet generation and service times exponentially distributed at average rates $\lambda_{23}$ and $c_2$, respectively. Node 3 simply receives packets for which it is the destination. In this model, all packets from Node 2 have Node 3 as their destination. Packets from Node 1 may have Node 2 or Node 3 as their destination.

![Figure 3.10: Queue operation as nodes switch and reacquire wireless links](image)

Figure 3.10: Queue operation as nodes switch and reacquire wireless links
Figure 3.10 shows how the queue behavior changes during system operation. From $t_0$ to $t_1$, the server in Node 1 serves only the queue with packets destined for Node 2. We will call this queue $Q_{12}$. The queue with packets destined for Node 3 ($Q_{13}$) simply fills up during this time because the node 1 server is busy serving $Q_{12}$.

Meanwhile, $Q_{23}$ in Node 2 is served by the Node 2 server and packets are forwarded to Node 3. At Node 3, packets are simply received with no additional delay. The nodes continue in this state until the size of $Q_{12}$ drops below a lower threshold, $NT_{low}$.

From $t_1$ to $t_2$, the nodes are synchronizing as part of the link acquisition process and no packets are served in any queue. (Later, we will define and use the synchronization time, $t_{synch}$, related to this synchronization event.) Again, all queues just fill up because packets are being generated but not served in each queue.

From $t_2$ to $t_3$, Node 1 has acquired Node 3 and is sending packets from $Q_{13}$. $Q_{12}$ and $Q_{23}$ are not being served, so they just fill up. Service continues in this manner until the queue size of $Q_{13}$ drops below a lower threshold, $NT_{low}$.

Finally, from $t_3$ to $t_4$, the nodes synchronize wireless links again during acquisition and all queues fill during the synchronization period. The stages of this process repeat for future time increments. During the time represented in Figure 3.10, synchronization between network nodes occurs twice.

We draw heavily from queuing theory to develop mathematical models for the average packet delay for the packet forwarding and the switching network models.

Approximating the queues as $M/M/1$ systems, we can write for the average system delays:

$$T_1 = \frac{1}{c_1 - \lambda_{12} - \lambda_{13}}$$  \hspace{1cm} (3.29)

$$T_2 = \frac{1}{c_2 - \lambda_{13} - \lambda_{23}}$$  \hspace{1cm} (3.30)

Keeping in mind that packets generated in Node 1 that are bound for Node 3 must wait in both the Node 1 and Node 2 queues, we obtain the expression for average packet
Table 3.1: Queue service status and time intervals

<table>
<thead>
<tr>
<th>Period</th>
<th>In Service</th>
<th>Duration Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0 - t_1$</td>
<td>$Q_{12}, Q_{23}$</td>
<td>$t_1 - t_0 = \frac{N_{12}(t_0) - NT_{low}}{c_1 - \lambda_{12}}$</td>
</tr>
<tr>
<td>$t_1 - t_2$</td>
<td>None</td>
<td>$t_2 - t_1 = t_{synch}$</td>
</tr>
<tr>
<td>$t_2 - t_3$</td>
<td>$Q_{13}$</td>
<td>$t_3 - t_2 = \frac{N_{13}(t_2) - NT_{low}}{c_1 - \lambda_{13}}$</td>
</tr>
<tr>
<td>$t_3 - t_4$</td>
<td>None</td>
<td>$t_3 - t_4 = t_{synch}$</td>
</tr>
</tbody>
</table>

For comparison with the behavior of the switching model, we develop delay equations for the special case of a homogeneous network where the packet generation and service rates for all nodes are the same. That is, $G_i = G$ and $c_i = c$. Substituting $\lambda$ for $\lambda_{ij}$ and $c$ for $c_i$ as described above, we obtain:

$$T_{forwarding} = \frac{\lambda_{12}}{\sum_{12,13,23} \lambda_{ij}} \left( \frac{1}{c_1 - \lambda_{12} - \lambda_{13}} \right) + \frac{\lambda_{13}}{\sum_{12,13,23} \lambda_{ij}} \left( \frac{1}{c_1 - \lambda_{12} - \lambda_{13}} \right) + \frac{1}{c_2 - \lambda_{13} - \lambda_{23}} + \frac{\lambda_{23}}{\sum_{12,13,23} \lambda_{ij}} \left( \frac{1}{c_2 - \lambda_{13} - \lambda_{23}} \right),$$

(3.31)

which reduces to:

$$T_{forwarding} = \frac{4}{3c(1 - 2\rho)},$$

(3.32)

where $\rho = \frac{\lambda}{c}$. Later in the chapter, we will use this homogeneous network model to compare delays between the packet forwarding model and the link switching model.

For the node switching model, the time interval for each period can be approximated by the equations shown in Table 3.1. Here, $N_{ij}(t_k)$ represents the number of packets in queue $ij$ at time $t_k$. Synchronization time, $t_{synch}$, is the time it takes for a node to drop wireless synchronization with one node and establish a new wireless link with the other node and is assumed to be a constant for this analysis. We also make the very important steady state assumption that queue sizes for each queue are the same at the beginning and end of this switching and synchronization cycle.

Using Little’s law and Table 3.1, we define a series of iterative equations to approximate the number of packets in each queue at each time, $t_0$ to $t_4$. Then, in order to
obtain a solution for the steady state situation, we set $N_{12}(t_4)$ equal to $N_{12}(t_0)$ and $N_{13}(t_4)$ equal to $N_{13}(t_0)$. This condition states that at the time we observe the system of queues, the number of packets in $Q_{12}$ and $Q_{13}$ is stable over a complete cycle of switching service and the required synchronization between nodes for wireless link establishment.

At time $t_1$, $N_{12}$ has reached the low threshold at which switching must begin so

$$N_{12}(t_1) = N_{T_{low}}.$$  \hspace{1cm} (3.34)

Synchronization occurs during $t_1$ to $t_2$ with no queues being served, so we can approximate

$$N_{12}(t_2) = N_{T_{low}} + \lambda_{12}t_{\text{synch}}.$$ \hspace{1cm} (3.35)

Applying the iterative equations and the steady state condition to the queue sizes for $Q_{12}$ and $Q_{13}$, yields the approximations in Table 3.2.

**Table 3.2: Switching model queue size approximations**

<table>
<thead>
<tr>
<th>$Q_{12}$ size</th>
<th>$Q_{13}$ size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{12}(t_0) = N_{12}(t_3) + \lambda_{12}t_{\text{synch}}$</td>
<td>$N_{13}(t_0) = N_{13}(t_3) + \lambda_{13}t_{\text{synch}}$</td>
</tr>
<tr>
<td>$N_{12}(t_1) = N_{T_{low}}$</td>
<td>$N_{13}(t_1) = N_{13}(t_0) + \lambda_{13}(t_1 - t_0)$</td>
</tr>
<tr>
<td>$N_{12}(t_2) = N_{12}(t_1) + \lambda_{12}t_{\text{synch}}$</td>
<td>$N_{13}(t_2) = N_{13}(t_1) + \lambda_{13}t_{\text{synch}}$</td>
</tr>
<tr>
<td>$N_{12}(t_3) = N_{12}(t_2) + \lambda_{12}(t_3 - t_2)$</td>
<td>$N_{13}(t_3) = N_{T_{low}}$</td>
</tr>
<tr>
<td>$N_{12}(t_4) = N_{12}(t_3) + \lambda_{12}t_{\text{synch}}$</td>
<td>$N_{13}(t_4) = N_{13}(t_3) + \lambda_{13}t_{\text{synch}}$</td>
</tr>
</tbody>
</table>

These simultaneous linear equations can be solved in the standard way, noting that there are only two independent variables, $N_{12}(t_0)$ and $N_{13}(t_2)$.

Starting with the equations in Table 3.2, we write

$$N_{12}(t_0) = N_{12}(t_3) + \lambda_{12}t_{\text{synch}}$$ \hspace{1cm} (3.36)

$$= N_{12}(t_2) + \lambda_{12}(t_3 - t_2) + \lambda_{12}t_{\text{synch}}$$

$$= N_{T_{low}} + 2\lambda_{12}t_{\text{synch}} + \lambda_{12}\left(\frac{N_{13}(t_2) - N_{T_{low}}}{c_1 - \lambda_{13}}\right)$$

Simplifying and regrouping, we obtain:

$$N_{12}(t_0) = \left(\frac{\lambda_{12}}{c_1 - \lambda_{13}}\right)N_{13}(t_2) + 2\lambda_{12}t_{\text{synch}}$$ \hspace{1cm} (3.37)

$$+ N_{T_{low}}\left(1 - \frac{\lambda_{12}}{c_1 - \lambda_{13}}\right).$$

27
In a similar fashion, we obtain for $N_{13}(t_2)$:

$$N_{13}(t_2) = \left( \frac{\lambda_{13}}{c_1 - \lambda_{12}} \right) N_{12}(t_0) + 2\lambda_{13}t_{\text{synch}}$$

$$+ \ NT_{\text{low}} \left( 1 - \frac{\lambda_{13}}{c_1 - \lambda_{12}} \right).$$

Solving these equations simultaneously yields:

$$N_{12}(t_0) = NT_{\text{low}} + 2\lambda_{12}t_{\text{synch}} \left[ \frac{(c_1 - \lambda_{12})}{c_1 - \lambda_{12} - \lambda_{13}} \right],$$

and

$$N_{13}(t_2) = NT_{\text{low}} + 2\lambda_{13}t_{\text{synch}} \left[ \frac{c_1 - \lambda_{13}}{c_1 - \lambda_{12} - \lambda_{13}} \right].$$

Using these equations, we solve for the time intervals $t_1 - t_0$ and $t_3 - t_2$ and because $t_2 - t_1$ and $t_4 - t_3$ are both synchronization periods, we obtain Table 3.3.

Table 3.3: Switching model time duration approximations

<table>
<thead>
<tr>
<th>Duration</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1 - t_0$</td>
<td>$2\lambda_{12}t_{\text{synch}} \frac{(c_1 - \lambda_{12})}{c_1 - \lambda_{12} - \lambda_{13}}$</td>
</tr>
<tr>
<td>$t_2 - t_1$</td>
<td>$t_{\text{synch}}$</td>
</tr>
<tr>
<td>$t_3 - t_2$</td>
<td>$2\lambda_{13}t_{\text{synch}} \frac{c_1 - \lambda_{13}}{c_1 - \lambda_{12} - \lambda_{13}}$</td>
</tr>
<tr>
<td>$t_4 - t_3$</td>
<td>$t_{\text{synch}}$</td>
</tr>
</tbody>
</table>

Here, again, we use results from $M/M/1$ queues with service vacations to approximate the average system delays.

Let $V_1, V_2, \ldots$ be durations of successive vacations for an $M/G/1$ queue, where vacation durations are assumed to be independent and identically distributed.

From [17], [31], the expected system delay is:

$$T = \bar{X} + \frac{\lambda \bar{X}^2}{2(1 - \rho)} + \frac{\bar{V}^2}{2\bar{V}},$$

where $\lambda$ is the average arrival rate in the queue, $\bar{X}$ is the average service time for the server, $\bar{X}^2$ is the second moment of the average service time, $\bar{V}$ is the average vacation duration, and $\bar{V}^2$ is the second moment of the average vacation duration.

For our model, the duration of the server vacations is a constant. For queue $Q_{12}$, the duration of server vacations is the sum of the time needed for the nodes to synchronize,
the time for queue $Q_{13}$ to empty to the low threshold value $NT_{low}$, and the time for the nodes to synchronize again.

$$V_{12} = 2t_{synch} + \frac{N_{13}(t_2) - NT_{low}}{c_1 - \lambda_{13}},$$

which reduces to

$$V_{12} = 2t_{synch} \left( \frac{c_1 - \lambda_{12}}{c_1 - \lambda_{12} - \lambda_{13}} \right).$$

Similarly,

$$V_{13} = 2t_{synch} \left( \frac{c_1 - \lambda_{13}}{c_1 - \lambda_{12} - \lambda_{13}} \right),$$

and, because queues $Q_{12}$ and $Q_{23}$ are "on vacation" for the same duration,

$$V_{23} = 2t_{synch} \left( \frac{c_1 - \lambda_{12}}{c_1 - \lambda_{12} - \lambda_{13}} \right).$$

For a constant, $c$,

$$\bar{c} = c \quad (3.46)$$

$$\bar{c}^2 = c^2 \quad (3.47)$$

so we can express the waiting time in queue $Q_{12}$ as follows:

$$T_{12}(t_0 \rightarrow t_4) = \frac{1}{c_1} + \frac{\lambda_{12}}{c_1(c_1 - \lambda_{12})} + t_{synch} \left( \frac{c_1 - \lambda_{12}}{c_1 - \lambda_{12} - \lambda_{13}} \right).$$

(3.48)

The delays for queues with vacations $Q_{13}$ and $Q_{23}$ can be calculated in a similar fashion yielding

$$T_{13}(t_0 \rightarrow t_4) = \frac{1}{c_1} + \frac{\lambda_{13}}{c_1(c_1 - \lambda_{13})} + t_{synch} \left( \frac{c_1 - \lambda_{13}}{c_1 - \lambda_{12} - \lambda_{13}} \right),$$

(3.49)

and

$$T_{23}(t_0 \rightarrow t_4) = \frac{1}{c_2} + \frac{\lambda_{23}}{c_2(c_2 - \lambda_{23})} + t_{synch} \left( \frac{c_1 - \lambda_{12}}{c_1 - \lambda_{12} - \lambda_{13}} \right).$$

(3.50)
Using the average delay for each queue, we compute the overall average delay for the network by taking the weighted average of the average queue delays with respect to the number of packets generated in each queue:

\[
T_{\text{overall}}(t_0 \rightarrow t_4) = \frac{N_{12}^{\text{total}}}{N_{\text{total}}} T_{12}(t_0 \rightarrow t_4) + \frac{N_{13}^{\text{total}}}{N_{\text{total}}} T_{13}(t_0 \rightarrow t_4) + \frac{N_{23}^{\text{total}}}{N_{\text{total}}} T_{23}(t_0 \rightarrow t_4),
\]

(3.51)

where

\[
N_{ij}^{\text{total}} = \lambda_{ij}(t_4 - t_0),
\]

(3.52)

and

\[
N_{\text{total}} = N_{12}^{\text{total}} + N_{13}^{\text{total}} + N_{23}^{\text{total}}.
\]

(3.53)

Using these equations, we can calculate average network delay as long as we know generation and service rates, and the threshold value, \(NT_{\text{low}}\), for switching the in-service queue.

For use later in comparing the average delays for the case when all nodes are homogeneous, we derive the equivalent homogenous node delay equation to (3.33) for the link switching case by substituting \(c\) for \(c_i\) and \(\lambda\) for \(\lambda_{ij}\):

\[
\overline{T}_{\text{switching}} = \frac{1}{c} + \frac{\lambda}{c(c - \lambda)} + t_s\left(\frac{c - \lambda}{c - 2\lambda}\right).
\]

(3.54)

Using \(\rho = \lambda/c\), this equation becomes

\[
\overline{T}_{\text{switching}} = \frac{1}{c(1 - \rho)} + t_s\left(\frac{1 - \rho}{1 - 2\rho}\right).
\]

(3.55)

We used MATLAB to develop event-based simulation models for both the three node network in both packet forwarding and in link switching operation.

Inputs to the model included average packet generation rates for packets, \(\lambda_{12}, \lambda_{13}\), and \(\lambda_{23}\) and the average service rates at Nodes 1 and 2, \(c_1\) and \(c_2\). Each simulation ran until 30,000 packets were received at nodes 2 and 3. We ran the simulation 50 times to get the average packet delay. We plotted error bars indicating the 95% confidence intervals of the data. The simulation model for three node switching operation was also written in MATLAB.
Inputs to this model also included average packet generation rates for packets, \(\lambda_{12}, \lambda_{13},\) and \(\lambda_{23}\) and the average service rates at Nodes 1 and 2, \(c_1\) and \(c_2\). The model run duration depended on the number of queue switches desired. In general, we ran the model for three to ten queue switching periods, but ignored the delay data from the first one to four switching periods to try to prevent system start up behaviors from affecting the delay data collected. We used the total number of received packets at nodes 1 and 2 as a secondary limit to the number of queue switches for the cases when short synchronization times allowed rapid switching between the queues. Again, when presenting the data, we plotted error bars indicating the 95 percent confidence intervals.

The contemplated and demonstrated data rates for UWB operation are about up to about 500 megabits per second [16]. (This is not too far from the maximum 480 mbps rate specified for WiMedia UWB, either [27].) Assuming an average packet length of 1500 bytes, this yields a transmission rate of 42,000 packets per second, so we choose our basic service rate to be 40,000 packets per second. To check that models were applicable over a range of parameters, we also looked at situations with average service rates that were 50 percent higher, or 60,000 packets per second.

For continuous stable operation in the packet forwarding mode, we realized that the combined Node 1 packet generation rate, the sum of \(\lambda_{12}\) and \(\lambda_{13}\), could not exceed the average service rate at Node 1. This led to our choice of 15,000 packets per second as the basic average packet generation rate in each node.

Our choice of values for synchronization time were initially based on previous studies of UWB systems indicating that synchronization times were on the order of \(\mu s\) [27] to tens of ms [16]. However, we quickly realized that for longer synchronization times (ms), packet forwarding would always achieve significantly less average packet delay than switching operation for our three node network. So, we needed to define a "sweet spot" in the operation of this system where the combination of certain generation rates, service rates, and shorter synchronization times might make switching between nodes more desirable than simply forwarding packets through the established wireless links.

One very helpful way to identify this sweet spot is to use the mathematical models for packet forwarding and for link switching operation in the case of homogeneous nodes. By homogeneous, we mean that the nodes all have the same packet generation and service rates. By setting (3.33) for the forwarding model equal to (3.55) and simplifying, we obtain...
The equation for synchronization time is given by:

$$t_{synch} = \frac{1 + 2\rho}{3c(1 - \rho)^2}. \quad (3.56)$$

When plotted in Figure 3.11 for various service rates, $c$, and utilization factors $\rho$, this equation shows lines where the total average delay for forwarding packets is equal to the delay obtained when switching the links. For a given service rate and utilization, if the synchronization time used in the system is a bit less than the value on the equal delay line, less delay will be incurred by switching links. If the synchronization time used in the system is a bit more than the equal delay line value, less delay will occur when packets are forwarded without disturbing existing links.

![Figure 3.11: Lines of equal delay for packet forwarding and link switching operation based on the mathematical models](image)

In other words, this chart can be used to determine for our network whether for a given achievable synchronization time, the system should be set up to switch links or simply forward traffic without disturbing the links.

It is clear from the figure that for service rates characteristic of UWB networks, 40,000 packets per second, and for the utilization factor considered, $\rho \approx 0.375$, that acquisition times must be better than about 50 $\mu$sec to tip the advantage towards switching links.
rather than simply forwarding the packets through the intermediate node. This observation helps us focus our modeling on interesting values of acquisition time.

Figure 3.12 shows us the same information in a more intuitive way by plotting average delay time as a function of the utilization factor, $\rho$, for different synchronization times. The average packet service rate is fixed at our standard 40,000 packets per second. The plot clearly shows that under these conditions, switching operation incurs less delay than forwarding operation only in the region

$$10\mu\text{sec} \lesssim t_{\text{synch}} \lesssim 50\mu\text{sec}. \quad (3.57)$$

The remaining problem, though, is that we don’t know a priori how accurate the mathematical models are. If the models are accurate, this technique provides a useful way to determine the best mode of operation for the network. If the models are not accurate, utility decreases. We discuss the results of the models, including accuracy when compared to the simulation models later in this section.

The mathematical and analytical models for the three node network operating only by packet forwarding agreed very well, as can be seen in Table 3.4. The average delays
Table 3.4: Average packet delay comparison for packet forwarding models

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Packet Delay</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematical</td>
<td>133.3 µs</td>
<td>NA</td>
</tr>
<tr>
<td>Simulation</td>
<td>133.6 µs</td>
<td>9.7 µsec</td>
</tr>
</tbody>
</table>

![Average packet delay as a function of node acquisition time](image)

Figure 3.13: Good agreement is seen in model comparisons for $t_{synch} = 50msec$ and two values of average packet service rate agree very well and the 95% confidence interval is small in comparison with the mean delay.

For longer synchronization times, the mathematical and simulation models for the three node network with link switching operation also agree well. Figure 3.13 shows good agreement between the mathematical model and the simulation model for synchronization times from 0 to 50 ms for two different average packet service rates.

The results for much shorter synchronization times are not quite so agreeable. For synchronization times from 0 to 50 µsec, Figure 3.14 shows the mathematical model underestimates the actual delay for the same values of average packet service rate.

The probable reasons for this failure to agree is a bit clearer when we examine average packet delays in each individual node. Figure 3.15 compares the overall and individual queue delays for two cases. The left four plots are for a longer synchronization time - $t_{synch} = 50msec$. The right four plots are for a short synchronization time - $t_{synch} = 50µsec$. Each group of four subplots shows the overall delay, the delay in $Q_{12}$, the delay in $Q_{13}$, and the delay in $Q_{23}$ clockwise around the figure starting in the upper left quadrant.

In Figure 3.15(a), we see that for $t_{synch} = 50msec$, the mathematical and simulation models match well for $Q_{12}$ and $Q - 13$, but not as well for $Q_{23}$. In Figure 3.15(b),
the models don’t match that well for $Q_{12}$ and $Q_{13}$, but the correspondence is even worse in $Q_{23}$.

We believe the discrepancies arise from the assumptions used in the mathematical model regarding exhaustive service in the queues and achievement of steady state operation. In using the server with vacations model for queue operation, we inherit the implicit assumption that the queues are all exhaustively served. This is true for $Q_{12}$ and $Q_{13}$ in our model, but not exact for queue $Q_{23}$. Because $Q_{23}$ only switches service when $Q_{12}$ switches service, the service policy at $Q_{23}$ is a limited service policy, not an exhaustive one. During a switching cycle, the number of packets in $Q_{12}$ is always forced to zero at the end of the cycle by requiring exhaustive service of the queue. However, $Q_{23}$ may still have packets in the queue, even when $Q_{12}$ is empty. Even if generation and service rates are the same for the two queues, over many, many queue switches, $Q_{23}$ may end up with more packets and therefore with a longer average delay for each packet.

Also implicit in the derivation of our mathematical delay model is the idea that each queue effectively is in steady state and that the maximum queue size of each queue is approximately constant from cycle to cycle. This allowed us to set up and solve the simultaneous equations for delay, but assumed that queues were in equilibrium. For very short synchronization times, the queues may not reach this equilibrium condition before service is switched. Thus, our mathematical delay formulation underestimates the actual delays seen in the simulation model results.

Figure 3.14: The mathematical model underestimates delay in model comparisons for $t_{\text{synch}} = 50 \mu \text{sec}$ and two values of average packet service rate.
Figure 3.15: Subplots of individual queue delays in model comparisons for $t_{synch} = 50\text{msec}$ and $t_{synch} = 50\mu\text{sec}$. 
Chapter 4

Minimizing Delay in UWB Ring Networks with Added Links

In the last chapter, we determined that for many cases switching links between UWB nodes increases average packet delay because of the time needed for nodes to resynchronize during acquisition. Thus, in some cases a static ring of single input, single output UWB nodes may provide the best delay performance. When examining this ring, we might naturally ask, ”What happens if we add more single direction links to the ring?” In this chapter, we will examine this question and a few others as well, namely:

- What does the addition of more single direction links do to average delay?
- Does our previous method of mathematical analysis work to apportion link capacities and to calculate average delay?
- How should these links be added to minimize average delay?
- How do delays predicted by the mathematical model compare to measured delays using a simulation model?

4.1 Effect of Adding Links on Network Topology

Starting simply, by changing our node design for two nodes in the ring, we can add an extra single direction link between any two nodes. We will discuss the design changes needed for UWB nodes to support addition of more single direction links in a later section.
The concept of adding one single direction link is illustrated in Figure 4.1, which shows an eight node ring (a) with one link connecting a single node back to its upstream neighbor, and (b) a ring with a single added link that bisects the ring.

![Figure 4.1: Plain eight node ring and two ways to add an extra single direction link](image)

Two interesting ways of adding two single direction links to a plain ring of eight nodes is illustrated in Figure 4.2. In (a), the two links bisect the ring in orthogonal directions. In (b), the links bisect the ring between two nodes allowing bidirectional communication between these nodes.

![Figure 4.2: Two ways to add two extra single direction links to an eight node ring](image)

Given a sufficient number of additional single direction links, the topologies shown in Figure 4.3 are possible. In (a), the four additional links bisect the ring. In (b), 8 additional links allow bidirectional communication around the ring. Finally, in (c), 48 additional links connect the eight nodes in a full mesh network. For a ring of \( R \) UWB nodes, the number of links needed to set up the bidirectional ring and mesh topologies is shown in Table 4.1.

Thus, we have identified several interesting topologies to examine for adding single direction links to the UWB ring network. Next, we will look at how the UWB node design
Table 4.1: Number of links needed for topologies of interest

<table>
<thead>
<tr>
<th>Topology</th>
<th># of nodes</th>
<th>Total links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bidirectional ring</td>
<td>R</td>
<td>2R</td>
</tr>
<tr>
<td>Mesh</td>
<td>R</td>
<td>R(R-1)</td>
</tr>
</tbody>
</table>

in Figure 3.1 must change in order to support addition of single direction links.

Figure 4.4 (a) shows an eight node UWB ring with a single link added between Nodes 3 and 7. It is readily apparent for this network that six nodes maintain a single input, single output link design and two nodes must have a different design. By inspection, Node 3 must have one input link and two output links. Node 7 must have two input links and one output link. In (b), Node 3 requires a single incoming link from Node 2, a generator of packets, a receiver for Node 3 packets, and two output queues. As before, $G_3$ is the packet generation rate, $c_3$ and $c_9$ are packet service rates, $RX_3$ is the receiving rate for packets bound for Node 3 and $P_3$ is the offered load on the incoming link to Node 3. Note that we adopt the notation $c_9$ for the service rate of packets going from Node 3 to Node 7. Adding more links that bisect the ring, as in Figures 4.2 (a) and 4.3 (a) is easily accomplished by adding more nodes with two-in-one-out and one-in-two-out link configurations.

Two of the topologies we are interested in can be constructed using nodes with two input and two output links. As shown in Figure 4.5, two different node designs are needed for the two different topologies. The topology with the bidirectional link between Nodes 3 and 7 shown in Figure 4.5(a) requires two nodes with the fully symmetric node design in (b), while the full bidirectional ring of Figure 4.5(c) requires that each node be designed as in (d). The reason for this is that in the case of the bidirectional ring, packets received from Node 2 will always be sent to Node 4. Conversely, packets received from Node 4 will be sent to Node 2. There is no reason for incoming packets from Node 2 to be queued up
to go back to Node 2, which makes for a slightly simpler node design for supporting the bidirectional ring topology. On the other hand, for the topology with bidirectional links between Nodes 3 and 7, incoming packets may be queued to go to either of outgoing links.

4.2 Choosing Link Capacities to Minimize Delay

Our first task is to determine whether the analysis methods of Section 3.2 can be used to find the best way to apportion link capacities to minimize delay in the network. Beginning with the ring topology with a single cross-link as in Figure 4.1 (b), we assume that each queue operates as an $M/M/1$ queue. Recognizing that (3.8) still holds for this topology, except that there is now one extra queue adding delay, we can write for the eight node network with cross-link:

$$T_D(c) = \sum_{i=1}^{9} w_i T_i(c),$$

(4.1)

where the total number of queues in the network is nine, $w_i$ is a weighting factor, $T_i(c)$ is the average delay in seconds on link $i$, $j$ and $c$ is the capacity matrix defined in (3.3). Applying the $M/M/1$ delay equations allows us to write:

$$T_D(c) = \sum_{i=1}^{9} \frac{w_i}{c_i - AR_i},$$

(4.2)

recalling that $AR_i$ is the arrival rate of packets into each queue and defining $AR_9$ as the arrival rate into the queue in Node 3 that will send packets to Node 7 as shown in Figure 4.1. Using the same Lagrangian optimization approach as Section 3.2, we can find the link capacities $c_i$ that minimize the average packet delay. For this topology, the eight node case easily generalizes for $R$ nodes and we obtain the optimal values for the link capacities:
Figure 4.5: Node designs for topologies with bidirectional links
\[ c_i = \frac{\sqrt{w_i}}{\sum_{i=1}^{R+1} \sqrt{w_i}} \left[ B - \sum_{k=1}^{R+1} AR_k \right] + AR_i, \]  
(4.3)

where \( B \) is the aggregate capacity bound for the network, \( w_i \) is a weighting factor for each link, and \( AR_i \) is packet arrival rate in the queues in each node. Calling this the optimal capacity, \( c_{i \text{ opt}} \), and substituting back into (4.2), but generalizing the result for \( R \) nodes plus one cross-link yields:

\[ T_{D}(c_{i \text{ opt}}) = \sum_{i=1}^{R+1} \sqrt{w_i} \sum_{j=1}^{R+1} \sqrt{w_j} \left[ B - \sum_{k=1}^{R+1} AR_k \right], \]  
(4.4)

Recall that the weights, \( w_i \) are known, as is the capacity bound, \( B \). So, the only term that affects the delay is the sum of the arrival rates for every queue. Now, we carefully examine the arrival rate term for the new topology.

For this eight node topology with one cross-link, we can express the sum of the arrival rates as

\[ \sum_{k=1}^{9} AR_k = \sum_{1,2,4,5,6,8} (G_k + \bar{P}_k - RX_i + AR_3 + AR_7 + AR_9). \]  
(4.5)

Let \( G_3(4,5,6) \) be the packets generated in Node 3 that have Nodes 4, 5, or 6 as their destinations. Let \( \bar{P}_3(4,5,6) \) represent that portion of the offered load to Node 3 that must be forwarded to Nodes 4, 5, or 6. Finally, let \( G_3(7,8,1,2) \) and \( \bar{P}_3(7,8,1,2) \) be defined similarly. We now write:

\[ \sum_{k=1}^{9} AR_k = \sum_{1,2,4,5,6,8} (G_k + \bar{P}_k - RX_i) + G_3(4,5,6) + \bar{P}_3(4,5,6) + \]

\[ + G_3(7,8,1,2) + \bar{P}_3(7,8,1,2). \]  
(4.6)

Noting now that

\[ G_3(4,5,6) + G_3(7,8,1,2) = G_3 \]  
(4.8)

and

\[ \bar{P}_3(4,5,6) + \bar{P}_3(7,8,1,2) = \bar{P}_3 - RX_3, \]  
(4.9)

we find that

\[ \sum_{k=1}^{9} AR_k = \sum_{i=1}^{8} G_i + \sum_{i=1}^{8} P_i - \sum_{i=1}^{8} RX_i + \bar{P}_9. \]  
(4.10)
Table 4.2: Average delay equations for various network topologies

<table>
<thead>
<tr>
<th>Topology</th>
<th>Average delay formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain ring of R nodes</td>
<td>$R^2 \sum_{k=1}^{R} P_k$</td>
</tr>
<tr>
<td>Ring of R nodes with one cross link</td>
<td>$B - \sum_{k=1}^{R+1} P_k (R+1)^2$</td>
</tr>
<tr>
<td>Ring of R nodes with two cross links</td>
<td>$B - \sum_{k=1}^{R+2} P_k (R+2)^2$</td>
</tr>
<tr>
<td>Ring of R nodes with bidirectional links</td>
<td>$B - \sum_{k=1}^{2R} P_k (2R)^2$</td>
</tr>
</tbody>
</table>

Noting also that the sum of all packets generated is equal to the sum of all packet received, we observe that the sum of all arrival rates is equal to the sum of the offered loads to all nodes in the network:

$$\sum_{k=1}^{9} AR_k = \sum_{i=1}^{9} P_i.$$  \hspace{1cm} (4.11)

Realizing that the eight node calculations apply in general and substituting (4.11) into (4.3) we obtain a very useful expression for the average packet delay in a ring of UWB nodes with one single direction cross-link:

$$T_D(c_{opt}) = \frac{\sum_{i=1}^{R+1} \sqrt{w_i} \sum_{j=1}^{R+1} \sqrt{w_j}}{B - \sum_{k=1}^{R+1} P_k},$$  \hspace{1cm} (4.12)

There are two very important points to note here:

1. For given weights and aggregate capacity bound, average delay depends only on the sum of the offered loads on each link.

2. This result, which requires summing loads over all ring links and cross links, generalizes for adding any number of extra links.

This will allow us to compare delay performance for different topologies given a constant offered load matrix, $T$. Table 4.2 provides the equations for delay for a number of different topologies.

4.3 Comparison of Delay for Network Topologies with Equal Loading

As discussed in our earlier work [32], one way to compare the performance of the different network topologies we considered is to look at the case when offered loads are equal
for all nodes. Recalling our previous definition of the traffic matrix, $T$:

$$
T = \begin{bmatrix}
0 & \lambda_{1,2} & \lambda_{1,3} & \ldots & \lambda_{1,R} \\
\lambda_{2,1} & 0 & \lambda_{2,3} & \ldots & \lambda_{2,R} \\
\lambda_{3,1} & \lambda_{3,2} & 0 & \ldots & \lambda_{3,R} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{R,1} & \lambda_{R,2} & \lambda_{R,3} & \ldots & 0
\end{bmatrix}
$$

where each $\lambda_{i,j}$ represents average offered load in packets or bits per second between node $i$ and node $j$ in the network, we set $\lambda_{i,j} = \lambda$ to obtain equal loading for all nodes in the network.

### 4.3.1 Loading and Delay in the Ring Network

For an eight node network arranged in a ring with single direction links, offered loads for the network are as shown in Figure 4.6. For simplicity, only the loads for the two of the eight links are shown in the figure.

![Figure 4.6: Total offered load for eight node ring](image)

For the eight node ring it is straightforward, if somewhat laborious, to write down the offered load for each queue. Figure 4.6 shows offered loads for $P_1$, $P_2$, and $P_3$ as triangles of $\lambda_{ij}$ load values. Load values are listed in order based on the number of hops between the source and destination for that load value. As can be seen, there is always...
one load value that requires one hop for delivery, two that require two hops, etc. The total offered load to any queue is $28\lambda$, obtained by substituting $\lambda$ for $\lambda_{ij}$ and summing. For $R$ nodes in a ring, the offered load into the queue in each node is

$$\frac{R(R - 1)\lambda}{2};$$

thus the sum of all offered loads for an $R$ node ring network is

$$\frac{R^2(R - 1)\lambda}{2}$$

and for equal weights, $w_i = 1$, delay for a "plain" ring of $R$ nodes is given by:

$$T_{D-ring} = \frac{R^2}{B - \frac{R^2(R - 1)\lambda}{2}}.$$ (4.16)

For the eight node ring with one cross-link between Nodes 3 and 7, we can, again, laboriously write down the offered loads for the network, as shown in Figure 4.7. We can see by inspection that this topology tends to reduce the number of high hop count paths,
Table 4.3: Links and offered load in a ring network of R nodes with one bisecting cross-link

<table>
<thead>
<tr>
<th>No. of links</th>
<th>Offered load</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{R}{2} )</td>
<td>( \frac{R(R-1)\lambda}{2} )</td>
</tr>
<tr>
<td>( \frac{R}{2} )</td>
<td>( \frac{R(R-1)\lambda}{2} - \frac{\frac{R}{2}(\frac{R}{2} + 1)\lambda}{2} )</td>
</tr>
<tr>
<td>1</td>
<td>( \frac{\frac{R}{2}(\frac{R}{2} + 1)\lambda}{2} )</td>
</tr>
</tbody>
</table>

which reduces average delay. For the equal loading scenario, there are four links that pass 28\( \lambda \) of traffic load, four that pass 18\( \lambda \) of traffic load, and one that carries 10\( \lambda \) of traffic load. For this eight node network, assuming weights \( w_i \) of one, the average delay is:

\[
T_{D\text{--cross--linked ring}} = \frac{81}{B - 194\lambda}.
\] (4.17)

With a little work, we can generalize the delay equation for any network of R nodes, with R even. Table 4.3 lays out the number of links and the offered load for each for a network of R nodes with a single cross link.

Multiplying the offered load times the number of links and summing over all links provides the total offered load for the network:

\[
\sum_{i=1}^{R+1} P_i = \frac{R}{2} \frac{R(R-1)\lambda}{2} + \frac{R}{2} \left[ \frac{R(R-1)\lambda}{2} - \frac{\frac{R}{2}(\frac{R}{2} + 1)\lambda}{2} \right] + \frac{\frac{R}{2}(\frac{R}{2} + 1)\lambda}{2} \quad (4.18)
\]

yielding

\[
\sum_{i=1}^{R+1} P_i = \frac{R(7R^2 - 8R + 4)\lambda}{16}. \quad (4.19)
\]

Finally, we can write the equation for delay in a network of R nodes with a single cross link bisecting the ring and equal loading of all links as

\[
T_{D\text{--cross--linked ring}} = \frac{(R + 1)^2}{B - \frac{R(R^2 - 8R + 4)\lambda}{16}}. \quad (4.20)
\]

We limit this equation to R being an even integer because an odd number of nodes in a ring network cannot be split in half by a single added link. Thus the load values for the links in a ring with an odd number of nodes will be slightly different that those derived here for ring networks with an even number of nodes.

We can evaluate all the topologies discussed in a similar fashion to produce equations for average delay. Some of the equations can be generalized for a network of R nodes, while others cannot. Delays generalized for R nodes for various topologies are provided in Table 4.4.
Table 4.4: Delay equations for various network topologies assuming equal loading

<table>
<thead>
<tr>
<th>Topology</th>
<th>Delay Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>$\frac{R^2}{B - \frac{B^2(R-1)\lambda}{(R+1)^2}}$</td>
<td>R an integer</td>
</tr>
<tr>
<td>Ring with link to one upstream node</td>
<td>$\frac{R^3 - R^2 - 2R + 4\lambda}{(R+1)^2}$</td>
<td>R any integer</td>
</tr>
<tr>
<td>Ring with bisecting cross-link</td>
<td>$\frac{(R+1)^2}{B - \frac{(7R^3 - 8R^2 + 4R)\lambda}{16}}$</td>
<td>R even</td>
</tr>
<tr>
<td>Ring with two bisecting cross-links</td>
<td>$\frac{(R+1)^2}{B - \frac{(R+2)^2}{(R+1)\lambda}}$</td>
<td>$R = 8$; no general equation</td>
</tr>
<tr>
<td>Ring with bisecting bidirectional link</td>
<td>$\frac{4R^2}{B - \frac{R^2}{4}}$</td>
<td>R even</td>
</tr>
<tr>
<td>Ring with bidirectional links</td>
<td></td>
<td>R an integer</td>
</tr>
</tbody>
</table>

Using the formulas for delay for the various network topologies provided in Table 4.4, we can calculate average delay for networks with different topologies and number of nodes, $R$. For an offered load matrix, $T$, with $\lambda_{ij} = \lambda$, we normalize the delays by assuming a value of $\lambda = 1$ packet/s. There is no need to be too picky about what we choose for $\lambda$, because any value will allow us to compare the relative merits of the different topologies with respect to average packet delay.

Figure 4.8 shows offered load and delay for three of our topologies of interest - the ring, the ring with bisecting cross link, and the ring with a one hop cross-link back to an upstream node. For increasing numbers of nodes, $R$, the value of offered load with $\lambda = 1$ packet/s is shown for each of the three topologies. In order to calculate average delay, we must have a value for the aggregate capacity bound, $B$. However, before discussing what this bound should be for the delay calculations, we can learn some interesting things by looking at the offered load values for these three topologies.

First, note that even though the two topologies with the extra cross-links each require an additional transmitting link to function, the total offered load for these networks is actually less than that of the plain ring. So, for example, if we had 88 packets/s available as our aggregate capacity bound to operate a six node ring, both of the topologies with cross-links would operate stably because offered load would be less than the total capacity bound. However, the plain ring would be unstable.

Second, the total offered load for a ring with a one hop cross-link back to an upstream node is only slightly better than that of the plain ring, no matter the number of nodes, $R$. But, the ring with a single extra link bisecting the ring has significantly less offered load.
In order to compare the topologies fairly, we must choose an aggregate capacity bound that allows each network to operate stably. So, we choose a value for $B$ that is some fraction larger than the largest of the offered loads for the three topologies. In Figure 4.8, we have chosen $B$ to be 1.05 times larger than the largest offered load value (always that of the plain ring) of the three topologies for a given number of nodes, $R$.

Using this method, we calculate the average packet delays for each topology shown in the last three columns of the table. For increasing numbers of nodes, $R$, the average delays for the plain ring and for the ring with a cross-link back to an adjacent upstream node rapidly converge to the same value. However, the ring network with a cross-link bisecting the ring has less delay than the other two topologies even as $R$ gets large.

<table>
<thead>
<tr>
<th>Lambda 1</th>
<th>Factor to increase $B$ above total offered load</th>
<th>1.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Total offered load-plain ring</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Total offered load - ring with bisecting cross-link</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Maximum capacity bound $B$ for comparison</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Delay - plain ring (s)</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with bisecting cross-link (s)</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with one hop cross-link (s)</td>
<td>3.0</td>
</tr>
<tr>
<td>8</td>
<td>Total offered load-plain ring</td>
<td>224</td>
</tr>
<tr>
<td></td>
<td>Total offered load - ring with bisecting cross-link</td>
<td>194</td>
</tr>
<tr>
<td></td>
<td>Maximum capacity bound $B$ for comparison</td>
<td>218</td>
</tr>
<tr>
<td></td>
<td>Delay - plain ring (s)</td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with bisecting cross-link (s)</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with one hop cross-link (s)</td>
<td>2.0</td>
</tr>
<tr>
<td>15</td>
<td>Total offered load-plain ring</td>
<td>792</td>
</tr>
<tr>
<td></td>
<td>Total offered load - ring with bisecting cross-link</td>
<td>687</td>
</tr>
<tr>
<td></td>
<td>Maximum capacity bound $B$ for comparison</td>
<td>782</td>
</tr>
<tr>
<td></td>
<td>Delay - plain ring (s)</td>
<td>832</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with bisecting cross-link (s)</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with one hop cross-link (s)</td>
<td>1.2</td>
</tr>
<tr>
<td>20</td>
<td>Total offered load-plain ring</td>
<td>1575</td>
</tr>
<tr>
<td></td>
<td>Total offered load - ring with bisecting cross-link</td>
<td>1368</td>
</tr>
<tr>
<td></td>
<td>Maximum capacity bound $B$ for comparison</td>
<td>1562</td>
</tr>
<tr>
<td></td>
<td>Delay - plain ring (s)</td>
<td>1554</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with bisecting cross-link (s)</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with one hop cross-link (s)</td>
<td>0.9</td>
</tr>
<tr>
<td>25</td>
<td>Total offered load-plain ring</td>
<td>3800</td>
</tr>
<tr>
<td></td>
<td>Total offered load - ring with bisecting cross-link</td>
<td>3305</td>
</tr>
<tr>
<td></td>
<td>Maximum capacity bound $B$ for comparison</td>
<td>3782</td>
</tr>
<tr>
<td></td>
<td>Delay - plain ring (s)</td>
<td>3990</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with bisecting cross-link (s)</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Delay - ring with one hop cross-link (s)</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 4.8: Comparison chart of average delay for various topologies

Figure 4.9 provides a table comparing four different network topologies in a similar fashion. The topologies compared are:

- a plain ring,
- a ring with a single bisecting cross-link,
- a ring with two bisecting bidirectional cross-links, and
- a ring with all bidirectional links.

In this table, the topologies with cross-links clearly show less delay than the plain ring. The ring with two bisecting bidirectional cross-links has the least delay, though delay for all of the cross-link topologies converge to the same value for large numbers of nodes, $R$. 

48
There are interesting implications for the network design engineer here. For example, for most networks, better delay performance can be achieved by setting up the bisecting bidirectional cross-link topology using $R - 2$ presumably less expensive single input, single output nodes and two presumably more expensive two input, two output nodes. In fact this topology would certainly seem to provide a less expensive, higher performance alternative than using, say, a network of $R$ two input, two output nodes formed in a ring with bidirectional links.

But, the ring with all bidirectional links has a performance advantage that the other topologies lack - the ability to operate stably at the lowest aggregate capacity bound, $B$. To illustrate, consider a network of 12 nodes in which only 500 packets/s of aggregate bandwidth can be supplied. In this case, only the ring with all bidirectional links, which has an total offered load of 432 packets/s can operate stably. All other topologies would fail miserably.

<table>
<thead>
<tr>
<th>No. of nodes R</th>
<th>Total offered load-plain ring</th>
<th>Total offered load - ring with bisecting cross-link</th>
<th>Total offered load - ring with bidirectional cross-link</th>
<th>Total offered load - ring with all bidirectional links</th>
<th>Maximum capacity bound $B$ for comparison</th>
<th>Plain ring delay (s)</th>
<th>Ring with bisecting cross-link delay (s)</th>
<th>Ring with bidirectional cross-link delay (s)</th>
<th>Ring with all bidirectional links delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>90</td>
<td>78</td>
<td>66</td>
<td>54</td>
<td>95</td>
<td>8.0</td>
<td>3.0</td>
<td>2.2</td>
<td>3.6</td>
</tr>
<tr>
<td>8</td>
<td>224</td>
<td>194</td>
<td>164</td>
<td>128</td>
<td>235</td>
<td>5.7</td>
<td>2.0</td>
<td>1.4</td>
<td>2.4</td>
</tr>
<tr>
<td>12</td>
<td>792</td>
<td>687</td>
<td>582</td>
<td>432</td>
<td>832</td>
<td>3.6</td>
<td>1.2</td>
<td>0.8</td>
<td>1.4</td>
</tr>
<tr>
<td>15</td>
<td>1575</td>
<td>1368</td>
<td>1161</td>
<td>844</td>
<td>1654</td>
<td>2.9</td>
<td>0.9</td>
<td>0.6</td>
<td>1.1</td>
</tr>
<tr>
<td>20</td>
<td>3800</td>
<td>3305</td>
<td>2810</td>
<td>2000</td>
<td>3990</td>
<td>2.1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.8</td>
</tr>
<tr>
<td>25</td>
<td>7500</td>
<td>6530</td>
<td>5559</td>
<td>3906</td>
<td>7875</td>
<td>1.7</td>
<td>0.5</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 4.9: Comparison chart of average delay for more topologies

We can go one step further with this comparison by adjusting our aggregate capacity bound, $B$ in the model. In Table 4.10, the aggregate capacity bound, $B$, has been increased to 1.15 times the maximum total offered load of the alternatives compared. This means that for all network topologies being compared, we assume that the capacity available is 1.15 times that of the plain ring. In this case, an interesting reversal occurs. Both of the cross-linked topologies still outperform the plain ring with respect to average delay, but now for smaller numbers of nodes, $R$, the ring with all bidirectional links now has higher average packet delay than the plain ring! As $R$ increases above eight nodes, the delay performance of the ring with all bidirectional links improves and surpasses that of the plain ring, but
only by an insignificant amount! This type of analysis may spare the engineer from building a bidirectional ring with expensive two input, two output nodes that achieves only slight delay improvement over a much less expensive ring with single input, single output nodes.

<table>
<thead>
<tr>
<th>No. of nodes R</th>
<th>Total offered load-plain ring</th>
<th>Total offered load - ring with bisecting cross-link</th>
<th>Total offered load - ring with bidirectional bisecting cross-links</th>
<th>Total offered load - ring with all bidirectional links</th>
<th>Maximum capacity bound B for comparison</th>
<th>Plain ring delay (s)</th>
<th>Ring with bisecting cross-link delay (s)</th>
<th>Ring with bidirectional bisecting cross-link delay (s)</th>
<th>Ring with all bidirectional links delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>90</td>
<td>78</td>
<td>66</td>
<td>54</td>
<td>104</td>
<td>2.7</td>
<td>1.9</td>
<td>1.7</td>
<td>2.9</td>
</tr>
<tr>
<td>8</td>
<td>224</td>
<td>194</td>
<td>164</td>
<td>128</td>
<td>258</td>
<td>1.9</td>
<td>1.3</td>
<td>1.1</td>
<td>2.0</td>
</tr>
<tr>
<td>12</td>
<td>792</td>
<td>687</td>
<td>582</td>
<td>432</td>
<td>911</td>
<td>1.2</td>
<td>0.8</td>
<td>0.6</td>
<td>1.2</td>
</tr>
<tr>
<td>15</td>
<td>1575</td>
<td>1368</td>
<td>1161</td>
<td>844</td>
<td>1811</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>0.9</td>
</tr>
<tr>
<td>20</td>
<td>3800</td>
<td>3305</td>
<td>2810</td>
<td>2000</td>
<td>4370</td>
<td>0.7</td>
<td>0.3</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>25</td>
<td>7500</td>
<td>6530</td>
<td>5559</td>
<td>3906</td>
<td>8625</td>
<td>0.6</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Figure 4.10: Comparison chart of average delay for more topologies and higher capacity bound

Plotting the average delay performance for the various topologies is instructive as well. Figure 4.11 shows a log log plot of average delay versus the number of nodes for all topologies discussed for aggregate capacity at 1.05 times the offered load for the plain ring. The ring with the bidirectional bisecting cross-links outperforms all others. The ring with a bisecting single direction cross-link outperforms the ring with all bidirectional links. The ring with a single cross-link to an adjacent upstream node outperforms the plain ring only for networks with few nodes.

Figure 4.12 shows a log log plot of average delay versus the number of nodes for all topologies discussed for aggregate capacity at 1.15 times the offered load for the plain ring. The percent improvement among all topology choices is smaller, as indicated by the closer grouping of the performance plots for each alternative topology. Again, the ring with the bidirectional bisecting cross-links outperforms all others. The ring with a bisecting single direction cross-link outperforms the ring with all bidirectional links. The ring with a single cross-link to an adjacent upstream node and the ring with all bidirectional links perform nearly the same as the plain ring.

Finally, Figure 4.13 shows a log log plot of average delay versus the number of nodes for all topologies discussed for aggregate capacity at 1.25 times the offered load for the plain ring. The percent improvement achievable by various topology choices is lessened even further. Once again, the ring with the bidirectional bisecting cross-links outperforms
Figure 4.11: Comparison of average delay by topology with aggregate capacity to load ratio = 1.05
all others. The ring with a bisecting single direction cross-link outperforms the ring with all bidirectional links. Now, however, the ring with a single cross-link to an adjacent upstream node and the plain ring both outperform the ring with all bidirectional links with respect to delay.

### 4.4 Effect of Node Switching for Various Network Topologies

In the previous section, we saw delay performance improvements for various ring topologies with added links. This section explores the possibility of using switching links to provide virtual cross-links in the UWB ring as a way to improve delay performance. Of the many topologies with added links in Figures 4.1, 4.2, and 4.3, the three options shown in Figure 4.14 seem the most promising in terms of ease of implementation and potential delay reductions.

The most obvious is the plain ring Figure 4.14(a), which should be modeled for comparison with the any link switching cases. The single cross link case shown in Figure 4.14(b) is promising for it’s simplicity and relatively good delay performance for the equal
Figure 4.13: Comparison of average delay by topology with aggregate capacity to load ratio=1.25
loading situation explored in the last section. Finally, Figure 4.14(c) shows a dual switching ring in which the ring splits into two smaller rings during the switching phase.

![Figure 4.14: Promising ten node topologies to consider for link switching in a UWB ring](image)

The dual switching ring requires two nodes of the configuration shown in Figure 4.15. Each node would require two queues for keeping packets destined for the upper half of the ring separate from those bound for nodes in the lower half.

![Figure 4.15: Two special dual queue nodes are needed to implement the dual switching ring](image)

The dual ring will also need a rather complicated management scheme to reform the entire ring once the switching operation is performed. Once the ring is split into two rings, it will be difficult for one ring to "know" that the other ring is ready to switch back. This would involve a rejoining protocol based on accurate timing or signaling between the two sub-rings. Because of the complexity of needing two special nodes and a complex management protocol for switching two sub-rings back into the larger ring, we chose to analyze only the plain ring and the simpler single cross connect ring shown in Figures 4.14(a) and (b).

We analyze use of link switching to provide a virtual cross-link that bisects the ring
when switching occurs. This section expands upon research originally published in [33,34] that describe conceptual, mathematical, and simulation models for a simple data network. The expansion includes consideration of a ring network of ten nodes in which one of the nodes may switch links between its normal downstream neighbor and the node opposite on the ring. The nodes in the model use ultra-wideband (UWB) wireless as the physical layer, requiring relatively long node acquisition times when links are dropped and reestablished. We present a mathematical model of average delay as a function of network parameters and node acquisition time and compare this model to simulation model results. For comparison, we also model a ten node ring network that simply forwards packets around the ring to their required destinations without using any type of link switching.

4.4.1 Conceptual Models

**Packet Forwarding Model**  The ten node packet forwarding model is simply a ring of ten nodes as previously seen in Figure 4.14(a). Each node generates, receives, and forwards traffic as determined by the origins and destinations of packets generated in accordance with the traffic matrix, $T$. Because there is no link switching in this model, acquisition times for establishing new links do not affect the average packet delay in the network. The nodes in the are as described in Chapter 2 and as shown again for convenience in Figure 4.16.

![Figure 4.16: Block diagram of single input, single output UWB nodes assumed for delay modeling of the ten node ring network](image)

**Packet Switching Model**  However, for the ten node packet switching network shown in Figure 4.14(b), Node 3 occasionally switches between sending packets on to Node 7 and sending packets to Node 8. This behavior requires Node 3 to have a different physical
structure than other nodes. As shown in Figure 4.17, Node 3 must have a queue for each
half of the network. One queue holds packets destined for Nodes 1, 2, 8, 9, and 10 and is
termed the top half queue. The other queue holds packets destined for Nodes 4, 5, 6, and
7 and is called the bottom half queue. The top half queue has a low threshold \( N^L_3 \) and a
high threshold, \( N^H_3 \). These values are used to control Node 3 switching in the network.

![Block diagram of Node 3](image)

Figure 4.17: Block diagram of Node 3

The sequence of operations is shown in Figure 4.18. Here is how this network
operates:

- \((t_0 \text{ to } t_1)\) All nodes forward packets normally. Node 3 holds packets for top half
  (Nodes, 1, 2, 8, 9, 10) until the top half queue reaches the high threshold value, \( N^H_3 \).

- \((t_1 \text{ to } t_2)\) Node 3 signals Node 7 to stop sending to Node 8. Node 7 acknowledges the
  order from Node 3. Node 3 stops sending packets to Node 4. Node 7 stops sending
  packets to Node 8.

- \((t_2 \text{ to } t_3)\) Node 3 stops sending to Node 4. Node 3 acquires Node 8.

- \((t_3 \text{ to } t_4)\) Node 3 sends packets to Node 8 until the top half queue reaches the low
  threshold, \( N^L_3 \).

- \((t_4 \text{ to } t_5)\) Node 3 stops sending to Node 8. Node 3 acquires Node 4.

- \((t_5 \text{ to } t_6)\) Node 3 signals Node 7 to acquire Node 8. Node 7 acquires Node 8.

- \((t_6 \text{ to } t_7)\) Node 7 signals Node 3 that it has acquired Node 8. Node 3 waits for Node
  7 acknowledgement before sending data. Behavior continues with the same events as
  in \( t_0 \text{ to } t_1 \).
Figure 4.18: The states of network operation used for calculating average delay.

From Figure 4.18, we define four states: *Node 3 sending to Node 4*, *Acquisition*, *Node 3 sending to Node 8*, and *Waiting*. The durations of each state are also shown in Figure 4.18 and will be used later in delay calculations for the ten node switching model.

### 4.4.2 Mathematical Models

**Ten Node Packet Forwarding Operation**

Our model approach for packet forwarding in the ten node ring will be to assume that each queue operates as an $M/M/1$ queue and use the delay equations for each link derived in [30]. We will sum the delays over all links, assuming that each link is weighted equally. Thus, total delay is given by

\[
T_D(c_i) = \sum_{i=1}^{10} \frac{1}{c_i - AR_i},
\]

(4.21)
where $c_i$ is the capacity and $AR_i$ the arrival rates at each Node $i$. Arrival rate is given by

$$AR_i = G_i + \overline{P}_i - RX_i,$$  \hspace{1cm} (4.22)

where $G_i$ is the generation rate of packets, $RX_i$ is the receive rate of packets, and $\overline{P}_i$ is the offered load at Node $i$.

**Ten Node Switching Operation**

The approach for modeling the effect of switching behavior on delay in the ten node UWB ring is a bit more complex. We sum the delays in each node to arrive at total delay. Nodes 3 and 7 behave as servers with vacations, but when the ring is switched to send packets from Node 3 to Node 8, the offered loads to all nodes change as well. For example, Nodes 4 and 8, while served constantly go through periods when the offered load to each node goes to zero. This occurs when Nodes 3 and 7 stop sending traffic, for example when they are synchronizing with Nodes 4 and 8. Our approach is to calculate delays for each node in each network state defined in Figure 4.18 and then weight and sum for each delay based on the duration spent in each state of operation. In this discussion acquisition time, $t_{acq}$, is the same as synchronization time, $t_{sync}$, for the three node model.

For all nodes but 3 and 7, the delay formula in (4.21) works well, but because of the way the network operates with Node 3 having two queues, the bottom half nodes (4, 5, 6, 7) never forward any of the traffic bound for the top half nodes (1, 2, 8, 9, 10) from Node 3. Node 3 holds on to all of this traffic to send directly to Node 8 after the link switching operation. The arrival rate in the queue, $AR_i$, for each of these nodes changes as the network changes state. For the four states shown in Figure 4.18, here are the applicable delay equations for Node 1. Delay equations for all nodes but 3 and 7 follow similarly.

**Node 3 sending to 4 state:**

$$T_{3to4}^1 = \frac{1}{c_1 + RX_{3to4}^1 - \overline{P}_{3to4}^1 - G_1}$$  \hspace{1cm} (4.23)

**Node 3 sending to 8 state:**

$$T_{3to8}^1 = \frac{1}{c_1 + RX_{3to8}^1 - \overline{P}_{3to8}^1 - G_1}$$  \hspace{1cm} (4.24)

**Acquisition state:**

$$T_{acq}^1 = \frac{1}{c_1 + RX_{acq}^1 - \overline{P}_{acq}^1 - G_1}$$  \hspace{1cm} (4.25)
Wait state:

$$T_{\text{wait}}^{1} = \frac{1}{c_{1} + RX_{\text{wait}}^{1} - P_{\text{wait}}^{1} - G_{1}}.$$  \hspace{1cm} (4.26)

To evaluate these equations, we define offered load matrix, $T$:

$$T = \begin{bmatrix}
0 & \lambda_{1,2} & \lambda_{1,3} & \ldots & \lambda_{1,10} \\
\lambda_{2,1} & 0 & \lambda_{2,3} & \ldots & \lambda_{2,10} \\
\lambda_{3,1} & \lambda_{3,2} & 0 & \ldots & \lambda_{3,10} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{10,1} & \lambda_{10,2} & \lambda_{10,3} & \ldots & 0
\end{bmatrix}$$  \hspace{1cm} (4.27)

where each $\lambda_{i,j}$ represents the average offered load in packets or bits per second between node $i$ and node $j$ in the network and 10 is the number of nodes in the network. We calculate the values for $P_{i}^{\text{state}}$ and $RX_{i}^{\text{state}}$ by inspection from the offered load matrix, $T$.

For example,

$$P_{1}^{\text{3 to 4}} = \sum_{i=4,5,6,7,8,9} \lambda_{i,1} + \sum_{i=4,5,6,7,8,9} \lambda_{i,2} + \sum_{i=4,5,6,7,8,9} \lambda_{i,3} + \sum_{i=4,5,6,7,8,9} \lambda_{i,4} + (4.28)$$

$$+ \sum_{i=6,7,8,9,10} \lambda_{i,5} + \sum_{i=7,8,9,10} \lambda_{i,6} + \sum_{i=8,9,10} \lambda_{i,7} + \sum_{i=9,10} \lambda_{i,8} + \lambda_{10,9}$$

and

$$RX_{1}^{\text{3 to 4}} = \sum_{i=4,5,6,7,8,9,10} \lambda_{i,1}.$$  \hspace{1cm} (4.29)

The task of calculating these values is laborious, but not difficult. Nodes 3 and 7 behave differently, so we use another approach to calculate delays in each state.

First, recall that Node 3 actually has two queues, so we must calculate delays for both queues. In addition, during the Acquisition and Wait states no packets are served in Nodes 3 and 7, so average delay for packets during this time is just 1/2 of the duration of the state. Finally, when Node 3 sends to Node 4, packets in the upper half queue in Node 3 do not get served, so the delay is 1/2 of the duration of the state. The converse is true when the upper half queue of Node 3 is operating and the lower half queue is without service. For convenience, we number the upper half queue in Node 3 as Queue 11. We write
\[ T_{3 \rightarrow 4}^{3to4} = \frac{1}{c_3 + RX_{3 \rightarrow 4}^{3to4} - P_{3 \rightarrow 4}^{3to4} - G_{3}^{LH}} \]  
(4.30)

\[ T_{3 \rightarrow 8}^{3to8} = \frac{1}{2} \left( \frac{N_{3}^{H} - N_{4}^{L}}{(c_3 - AR_{3}^{U})} \right) \]  
(4.31)

\[ T_{3 \text{acq}}^{3to8} = \frac{1}{2} (2t_{\text{acq}}) \]  
(4.32)

\[ T_{3 \text{wait}}^{3to8} = \frac{1}{2} \left( t_{\text{acq}} + \sum_{i=1}^{10} \frac{1}{c_i} \right). \]  
(4.33)

For Queue 11, the upper half queue in Node 3, we write

\[ T_{11 \rightarrow 4}^{3to4} = \frac{1}{2} \left( N_{3}^{H} - N_{4}^{L} \right) + \sum_{i=1}^{10} \frac{1}{c_i} \]  
(4.34)

\[ T_{11 \rightarrow 8}^{3to8} = \frac{1}{c_3 + RX_{3 \rightarrow 8}^{3to8} - P_{7 \rightarrow 8}^{3to8} - G_{3}^{UH}} \]  
(4.35)

\[ T_{11 \text{acq}}^{3to8} = \frac{1}{2} (2t_{\text{acq}}) \]  
(4.36)

\[ T_{11 \text{wait}}^{3to8} = \frac{1}{2} \left( t_{\text{acq}} + \sum_{i=1}^{10} \frac{1}{c_i} \right). \]  
(4.37)

Finally, for Node 7 we write

\[ T_{7 \rightarrow 4}^{3to4} = \frac{1}{c_7 + RX_{7 \rightarrow 4}^{3to4} - P_{7 \rightarrow 4}^{3to4} - G_{7}} \]  
(4.38)

\[ T_{7 \rightarrow 8}^{3to8} = \frac{1}{2} \left( \frac{N_{3}^{H} - N_{4}^{L}}{(c_3 - AR_{3}^{U})} \right) \]  
(4.39)

\[ T_{7 \text{acq}}^{3to8} = \frac{1}{2} (2t_{\text{acq}}) \]  
(4.40)

\[ T_{7 \text{wait}}^{3to8} = \frac{1}{2} \left( t_{\text{acq}} + \sum_{i=1}^{10} \frac{1}{c_i} \right). \]  
(4.41)

For each state, we sum the delays in all nodes. Then we take the sum of the delays by state, weighting for the duration of each state.

\[ T_{\text{overall}} = W_{3to4} \sum_{j=1}^{11} T_{j \rightarrow 4}^{3to4} + W_{3to8} \sum_{j=1}^{11} T_{j \rightarrow 8}^{3to8} + W_{\text{acq}} \sum_{j=1}^{11} T_{j \text{acq}}^{3to8} + W_{\text{wait}} \sum_{j=1}^{11} T_{j \text{wait}}^{3to8}, \]  
(4.42)

where

\[ W_{\text{state } i} = \frac{\text{Duration}_{\text{state } i}}{\sum_{\text{all states } i} \text{Duration}_{\text{state } i}} \]  
(4.43)
4.4.3 Simulation Models

We used MATLAB to develop event-based simulation models for the ten node ring network in packet forwarding and in link switching operation. Both models assign loads randomly in the offered load matrix, $T$, defined previously in Figure 4.27. The offered loads, $\lambda_{i,j}$, are assigned randomly from a uniform distribution between 0 and 15,000 packets/s.

Packet forwarding model

The packet forwarding simulation model takes the traffic matrix, $T$ and the excess capacity factor, $B \text{ factor}$, as inputs and provides a text file of average delay and standard deviation of delay for each of 30 simulation runs. We set excess capacity factor, $B \text{ factor}$, to 1.005, 1.01, or 1.05. Simulations are stopped based on the total number of packets received in the network. Experimentation showed that for higher excess capacity factors, terminating the simulation when 8000 packets were received gave results indicating the network was operating at equilibrium.

The capacities for each node were set by solving (4.3) for capacities $c_i$ with link weights, $w_i$, set to one. The aggregate capacity, $B$, is the product of the excess capacity factor and the maximum offered load:

$$B = B \text{ factor} \times \sum_{i=1}^{10} P_i.$$  \hspace{1cm} (4.44)

Packet switching model

The packet switching simulation model also takes the traffic matrix, $T$, and the excess capacity factor, $B \text{ factor}$, as inputs. The value for the high switching threshold, $N^H_3$, for the top half queue in Node 3 is variable; the low threshold is always set to zero. We set the high threshold between 5 and 50 packets. We set the capacities for the nodes to values that provide aggregate capacity that exceeds the total offered load by a selectable amount. We looked at cases for the aggregate capacity at 1.05, 1.01. and 1.005 times total offered load. Generally, we examined average delay for acquisition times from 0 to 5 $\mu$s, 0 to 50 $\mu$s, and 0 to 500 $\mu$s.

We terminated simulation runs based on the number of times the Node 3 queues switched operation. Experimentation with the model indicated that 8 or 16 queue switches was usually enough to attain steady state operation. Figure 4.19 shows the difference in
Figure 4.19: Running simulations for 8 vs. 16 Node 3 queue switches

terminating simulations at eight switches vs. 16 switches. The delays in Figure 4.19(b) are
greater than those in Figure 4.19(a), because after eight switches the system has not yet
reached equilibrium.

Setting the capacities for each node also turned out to be somewhat complex. In
order for the system to be stable, we needed to guarantee that there was always enough
capacity to empty each queue so that the queues did not simply grow continuously during
operation. This was complicated by the fact that Node 3 switches operation periodically,
affecting the inputs to all other queues. We reasoned that we needed to select capacities for
each node that were large enough to handle all packets added to the nodes during all phases
of operation shown in Figure 4.18: Node 3 sending to 4, Node 3 sending to 8, Acquisition,
and Wait.

From Figure 4.18, we approximated the durations of Node 3 sending to 4 and
Node 3 sending to 8 to be the same and called this value for operational duration $D_{ops}$. We
also approximated the Wait duration as $t_{acq}$. These approximations allowed us to write the
following equations for node capacities and a function of other system parameters:

$$c_i \geq \frac{D_{ops}(AR_i^{3to4} + AR_i^{3to8}) + 3t_{acq}AR_i^{acq}}{2D_{ops} + 3t_{acq}} \quad \text{for } c_i = 1, 2, 4, 5, 6, 8, 9, 10; \quad (4.45)$$

$$c_3 \geq \frac{D_{ops}(AR_3^{3to4} + AR_3^{3to8}) + 3t_{acq}AR_3^{acq}}{2D_{ops}}; \quad (4.46)$$
and

\[ c_7 \geq \frac{D_{ops}(AR_7^{tot4} + AR_7^{tot8}) + 3t_{acq}AR_7^{acq}}{D_{ops}}. \] (4.47)

The operational duration, \( D_{ops} \), is given by:

\[ D_{ops} = \frac{N^H_3}{c_3 - AR_3^{Upper\ half\ queue}}, \] (4.48)

and \( AR_i^{state} \) is the arrival rate in queue \( i \) in system condition \( state \). All these equations say is that the service rate times the operating duration for the node must exceed the arrival rate in the queue times the sum of the operational and acquisition durations. Simply put, the node capacities must be able to empty out the queues even though the queues fill continuously and the nodes only send packets during their operational periods. The operational period is defined in a straightforward manner in (4.48). It is simply the time it takes to empty or fill the upper half queue of Node 3.

We solve (4.45) and (4.47) outright to obtain capacities for nodes 1, 2, and 4 through 10. We solve (4.46) and (4.48) simultaneously to obtain \( c_3 \). By summing the capacities for Nodes 1 through 10, we obtain the aggregate capacity for the system. Multiplying by the excess capacity factor (\( B \) factor) yields the total capacity used in the mathematical or simulation model.

In order to fairly compare the packet forwarding ring with the packet switching ring, we must choose the optimal node capacities for each model. So, while we use the capacities, \( c_i \), we derived above for the switching ring, we must take the total capacity obtained above and use (3.20) to back out the optimal capacities for each node when the ring is operating simply in a packet forwarding mode.

Finally, we note that the optimal node capacities and therefore the aggregate network capacity for the switching network are functions of the acquisition time. As acquisition time increases, node capacities must increase to keep the system stable because more packets get added during the longer acquisition times that must then be served at each node. To simplify plots of delay versus acquisition times that we will use for performance evaluation, we will use the capacities for the largest acquisition time in a plot set to calculate the delay for all points in that plot. If we do not adopt some such convention, we end up with plots like Figure 4.20 that are very hard to interpret due to abrupt changes in modeled packet delay as the node capacities change based on acquisition times to ensure stable operation.
### 4.4.4 Results

In this section we provide the results of mathematical and simulation models for comparing average packet delay as a function of acquisition time for a ten node UWB ring network with switching operation and for a ten node UWB ring network that forwards packets around the ring without any type of switching of links.

**Packet forwarding model**

For the packet forwarding UWB ring network, acquisition time is not a factor because nodes do not have to re-synchronize during normal operation of the ring network. The average packet delay is a function of the capacities chosen for each of the ten nodes and of the aggregate capacity for the network. Figure 4.21 shows results of the mathematical and simulation models for various excess capacity factors. Recall that the excess capacity factor (\(B \text{ factor}\)) is multiplied by the total network capacity to get the total aggregate capacity at which the network will operate. A higher \(B \text{ factor}\) means that there is more network capacity above and beyond what is required for the network to handle the total offered load.

In Figure 4.21, the blue circles show average delay predicted by the mathematical model for \(B \text{ factors}\) of 1.01, 1.05, 1.10 and 1.25. The asterisks with whisker plots showing...
95% confidence intervals represent the delays obtained running the simulation model for the packet forwarding UWB ring network. There are two different simulation results shown for the \( B \) factor of 1.01 that we will discuss later in this section.

In general, for \( B \) factor at or above 1.05, agreement is very good between the models for average packet delay. In fact, at the very high \( B \) factor value of 1.25, the agreement is excellent. However, the agreement between the simulation and mathematical models for \( B \) factor of 1.01 is not good at all. The delay obtained using the simulation model is much smaller than that predicted by the mathematical model.

What we are seeing is a different version of a classic queuing system effect. That is, the closer the ratio of service rate to arrival rate approaches one, the more random effects dominate the performance of the system. The simulation results shown in black were obtained with 60 runs of the ten node packet forwarding simulation each terminating when 8,000 packets were received by the nodes. When excess capacity greatly outstrips the total offered load, agreement between the models is excellent. But, the closer the capacity approaches the total offered load, the more uncertainty creeps into the process. At a \( B \) factor of 1.01, collecting delay data for 8,000 received packets just is not a large enough sample to attain the certainty of agreement we would like to see. This brings us to the red simulation data point shown in Figure 4.21. We obtained this data point by running the simulation for 60 runs that terminated after the nodes had received a total of 16,000 packets. As can be seen, the new data point has moved closer to the delay predicted by the mathematical model for a \( B \) factor of 1.01.

For the rest of our analysis, we will use only the mathematical model of delay for the packet forwarding model when comparing delay performance with the ten node switching model.

**Packet switching model**

**Mathematical model** One very interesting feature of the mathematical delay model for the ten node switching ring network is the non-linear shape of the delay plot, which can be seen in Figure 4.22. This plot shows average delay for acquisition times between zero and 100 \( \mu s \) with excess capacity factor of 1.05 and a switching threshold of five packets. Intuitively, we expect delay to increase monotonically with acquisition time, but the plot shows a slight decrease in average delay for increasing acquisition time for a small region of acquisition times. The parabolic shape of the delay curve is due to the effect of weighting
Figure 4.21: Comparison of average delay versus excess capacity factor for mathematical and simulation models of the UWB packet forwarding ring network
and summing the individual delays to obtain total average delay. As the acquisition time increases from zero, the delays in the Node 3 sending to 4 and Node 3 sending to 8 states of Figure 4.18 increase, but are weighted less because the duration of these states become increasingly short as compared to the Acquisition and Wait states. The weighting effect dominates at first causing the total delay curve to slope slightly downwards. Before long, however, the decrease in weighting due to increasing acquisition time is overcome by the fact that actual delays in all states are increasing or remaining constant as the acquisition time increases. This causes the total delay to ramp up again. As we will see, this effect does not seem to occur in the simulation model.

Figure 4.23 shows the effect of excess capacity and queue switching threshold on the ten node network with acquisition times ranging from 0 to 5 ms. For longer acquisition times, neither excess capacity nor the value chosen for the switching threshold significantly affects the average delay. For these cases, the packet forwarding ring network outperforms the ring with switching nodes except for very short acquisition times. The effect of excess capacity and the switching threshold are greatly outweighed by the large acquisition time. In fact, comparing Figure 4.23 with Figure 3.13 for the simple three node switching model shows that even the effect on delay of the number of nodes in the network (ten vs. three) is completely overshadowed by large acquisition delays.
The three factors that influence the delay performance of the ten node switching network model are the high switching threshold for the top half queue in Node 3, $N^H_3$, the excess capacity ratio, $B$, and the acquisition time required for nodes to re-synchronize with a new link. We look at these factors by analyzing model results in three acquisition time regimes: 0 to 5 $\mu$s, 0 to 50 $\mu$s, and 0 to 500 $\mu$s.

**Acquisition times from 0 to 5 $\mu$s**

Figure 4.24 plots average delay as a function of acquisition time for the case when excess capacity is 1.01 times the total offered load. The capacity bound is the excess capacity factor times the total network capacity. The average node capacity is the capacity bound, $B$, divided by the total number of network nodes. The average offered load is the average $\lambda_{i,j}$ value from the traffic matrix, $T$. The top dotted line is the mathematical model prediction of average delay as a function of acquisition time. The solid straight line is the mathematical model prediction of average packet delay for the packet forwarding ring with no switching. The hashed line in the middle with 95 percent confidence bars is the simulation model delay versus acquisition time. The average delay from the simulation model is consistent with the mathematically modeled delay for a switched network with these parameters. The delay predicted by the mathematical model for the packet forwarding ring network is nearly the same as that modeled for the switching network.

Figure 4.25 shows the effect of switching threshold $N^H_3$ on performance of the ten node switching network when acquisition time ranges from 0 to 5 $\mu$s. For these short
Figure 4.24: Network delay versus acquisition time

acquisition times, as the switching threshold increases, more average delay accumulates because of packets waiting in the top half queue in Node 3 until the queue size reaches the switching threshold. The simulation model shows the effect clearly. The changes in average delay caused by changes in switching thresholds predicted by the mathematical model are not as great. In 4.25(b) and (c), the delay predicted for the packet forwarding mathematical model is greater than that shown by either switching model.

Figure 4.26 shows the effect of changes in the excess capacity factor for the switching network with acquisition delays ranging from 0 to 5 $\mu$s. In this case, the mathematical model seems much more sensitive to changes in excess capacity. As we see, the mathematical model predicts increasing average delay as excess capacity factor is reduced from 1.05 to 1.005. The delay predicted by the mathematical model for the simple ring of nodes forwarding all traffic is a bit less than the average delay shown by either model of the switching network.

**Acquisition times from 0 to 50 $\mu$s**

Figure 4.27 shows plots of model outputs for acquisition times from 0 to 50 $\mu$s with a switching threshold of five packets and an excess capacity figure of 1.01. The mathematical model for average delay for the switched ten node network is not far off from
Figure 4.25: Effect of switching threshold on delay for 0 to 5 $\mu$s acquisition delay
Figure 4.26: Effect of capacity excess on delay for 0 to 5 µs acquisition delay
the delays predicted for the forwarding ten node ring. However, the simulation model for packet switching yields a higher average delay. Overall, the models predict that forwarding operation in the ten node ring will have the smallest average delay.

For acquisition times between 0 and 50 $\mu$s, Figure 4.28 shows the effect of a larger switching threshold for the top half queue in Node 3. The simulation models shows progressively increasing average delay in (a) through (c) due to packets having to wait longer in the top half queue before switching allows them to be serviced. For this range of acquisition times, the mathematical models for the forwarding network and the switching network are now somewhat insensitive to the switching threshold, at least between 10 and 20 packets in (b) and (c). The packet forwarding ring network continues to show the smallest average delays. In Figure 4.29, there is nearly no effect to changing the excess capacity factor within the range of 1.005 to 1.05 on the delay plots. Once again, the mathematical model for the ten node forwarding network yields the smallest average delay over this interval of acquisition times.

**Acquisition times from 0 to 500 $\mu$s**

For longer acquisition times, on the order of 0 to 500 $\mu$s, the mathematical and simulation model delay predictions for the switched ring network begin to diverge as shown
Figure 4.28: Effect of switching threshold on delay for 0 to 50 $\mu$s acquisition delay
Math model − 10 node ring − switching
Math mode − 10 node ring − no switching
Simulation model − 10 node ring − switching
Queue switches=16

Figure 4.29: Effect of capacity excess on delay for 0 to 50 µs acquisition delay
in Figure 4.30. For reasons we have not been able to determine, the delay calculated by the mathematical model for the switching network shows a steeper increasing slope than that of delays obtained using the simulation model. The mathematical model for the forwarding ring network provides the smallest delay.

Figure 4.31 shows a slight increase in average delay for the switched network simulation as the excess capacity decreases from 1.05 to 1.005 times the offered load. This corresponds with the expected behavior for the system - less capacity excess should lead to longer average packet delays.

The effect of increasing switching threshold for the switched network is shown in Figure 4.32. As shown in (a), average delay is higher in the switched network simulation model because the network switches at a lower threshold incurring the penalty of the additional required acquisition times for each switch. When the switching threshold is increased to 10 or 20 packets as in Figure 4.32(b) and (c), the average delay is less. The plots also show best delay performance by the forwarding network mathematical model and a significant mismatch between the mathematical and simulation models for the switching network as acquisition times increase.
Acquisition time (s)
Average delay (s)
Average offered load = 7532 packets/s
Capacity bound, $B = 4364286$ packets/s
Average node capacity = 436429 packets/s
$B$ factor = 1.05
$N_3^H = 20$ packets
Math model − 10 node ring − switching
Math mode − 10 node ring − no switching
Simulation model − 10 node ring − switching
Queue switches=16

Acquisition time (s)
Average delay (s)
Average offered load = 8007 packets/s
Capacity bound, $B = 4942412$ packets/s
Average node capacity = 494241 packets/s
$B$ factor = 1.01
$N_3^H = 20$ packets
Math model − 10 node ring − switching
Math mode − 10 node ring − no switching
Simulation model − 10 node ring − switching
Queue switches=8

Acquisition time (s)
Average delay (s)
Average offered load = 7532 packets/s
Capacity bound, $B = 4177245$ packets/s
Average node capacity = 417725 packets/s
$B$ factor = 1.005
$N_3^H = 20$ packets
Math model − 10 node ring − switching
Math mode − 10 node ring − no switching
Simulation model − 10 node ring − switching
Queue switches=16

Figure 4.31: Effect of switching threshold capacity excess on delay for 0 to 500 $\mu$s acquisition delay

76
Figure 4.32: Effect of switching threshold on delay for 0 to 500 µs acquisition delay
4.4.5 Summary

This section describes mathematical and simulation models for the average packet delay for two types simple ten node UWB networks operating without a UWB MAC layer. In one type of operation, the ring operates with one node that can switch between delivering packets to its downstream neighbor and delivering packets to the opposite node on the ring. In the other operational mode, the ten node ring network operates by simply forwarding packets through the intermediate nodes from source to destination.

We saw that setting up the modeled parameters to ensure fair comparisons between the forwarding UWB and the switching UWB ring was challenging because the capacities needed by each node to guarantee stable operation varied by network type and by acquisition time. We settled on a method that uses the highest capacities needed by either network at the maximum acquisition time to be plotted in a single plot.

For the ten node packet forwarding ring, we showed the effect of excess capacity on the number of samples needed for statistical confidence. We also showed good agreement between the mathematical and simulation models for packet delay as a function of excess capacity. For a $B$ factor of 1.01, we showed that the simulation model delay was significantly less than that predicted by the mathematical model for delay in the forwarding ring network.

For the ten node ring switching network, we showed the effect of switching threshold, excess capacity, and acquisition time on average packet delay. These effects show that for small acquisition times, the switching model can deliver packets with about the same delay as simply forwarding packets around the ring. We also showed that large synchronization times overshadow effects of the switching threshold and excess capacity and recognized that the mathematical model results were very similar for the three node and ten node models for large acquisition times.

The most significant finding was that even for very short acquisition times, the ten node ring network that simply forwards packets around the ring always delivers packets with less average delay than the ring node with a single switching link.
Chapter 5

UWB Network Management Challenges

As with any network topology that may experience dynamic changes caused by link failure or node movement, the UWB ring networks described in this thesis must include network management protocols. Two of the crucial events that the network management protocols must handle are the nodes joining and nodes leaving the ring. We examined existing protocols for nodes joining and leaving networks to evaluate their applicability to the UWB ring networks presented here. We chose Wireless Token Ring Protocol (WTRP) [35, 36], Bluetooth [37–40], and Self-Organizing Device Discovery Protocol (SDD) [41–45]. We chose WTRP because it is a medium access control (MAC) protocol suitable for wireless ad hoc ring networks running over ethernet. We chose Bluetooth because of its widespread success in supporting implementation of personal area networks - a venue for which UWB is also well-suited. We chose SDD because it was specifically designed for pulse radio UWB networks. We concluded that the methods used in these protocols for nodes to join and leave the network were not well suited for UWB ring networks with relatively long acquisition times. We present our own protocol for nodes to join and leave the UWB ring network, explain its operation, and compare it with those used in WTRP, Bluetooth, and SDD.
5.1 Joining a Ring

5.1.1 WTRP Join Method

First applied to intelligent transportation systems [46], WTRP was later proposed for use in wireless ad hoc networks [47]. Inspired by the IEEE 802.4 Token Bus Protocol, WTRP added special tokens, new fields to existing tokens, and new event timers to improve performance in partially connected networks. The creators of the protocol have continuously improved the original concept and others have built on the concept suggesting additional modifications, such as the Wireless Dynamic Token Protocol (WDTP) [48]. Suggested modifications to WTRP include changing the token transfer algorithm to allow token passing to transcend the normal ring path. Instead, the token can be passed along a path adapted to the dynamic structure of the network to maximize the number of nodes in each subnet.

WTRP has been used in situations requiring rapid establishment of communications networks where quality of service (QoS) guarantees are important. It has been implemented in research projects involving automated control of highway traffic and control of autonomous helicopters. It is a well-researched and suitable protocol for controlling ring networks in ad hoc networks. Consequently, we chose to investigate the methods used in WTRP for nodes joining and leaving the ring as a possible model for control of ring networks of simple UWB nodes.

As shown in Figure 5.1, assume Node 2 wants to join the ring network containing Nodes 1 and 3. Node 1 is assumed to be the operating Admission Control Manager (ACM) for the network. As such, Node 1 periodically solicits other stations to join if ring resources are available [36]. First, Node 1 broadcasts a `solicit_successor` token that includes the address of Node 3, the current successor of Node 1. Node 2 (along with any other node responding to the solicitation) sends a `set_successor` token to Node 1. The ACM resident on Node 1 waits for the duration of a preset response window and then decides which of the responding nodes will be allowed to join the ring. Node 1 decides to admit Node 2 to the ring and sends a `set_predecessor` token to Node 2. Node 2 then sends a `set_predecessor` token to Node 3 and the ring is closed. As described, there are four separate transmissions needed to accomplish the ring join for Node 2 under WTRP.
Figure 5.1: WTRP requires several distinct data transmissions to allow a node to join the ring

### 5.1.2 Bluetooth Join Method

Bluetooth is an open standard for wireless communication primarily intended to replace cables between computers, cell phones, and various peripheral devices with short range radio links [39]. Bluetooth devices are organized into groups of two to eight active devices called piconets. Each piconet has one master node and up to seven active slave nodes. Piconets that share common nodes can be combined into scatternets. Nodes comprising a piconet all transmit on 79 one-Mhz channels using a frequency hopping spread spectrum (FHSS) technique [38]. The master node in the piconet controls all transmissions.

One important Bluetooth function is to operate in the office and home as part of a personal area network. Because networks based on UWB also have short ranges and are intended to support personal area network applications, we naturally chose Bluetooth to provide ideas for possible UWB network joining protocols.

Figure 5.2 shows the Bluetooth joining process. The master node in a piconet starts the joining process by inquiring what Bluetooth devices are in range. Bluetooth nodes in range not already connected to the master node’s piconet respond with an inquiry response containing the unique Bluetooth device address for the node. The master node follows up with a separate page to each node the master desires to join the piconet that contains the frequency hop synchronization (FHS) information for the piconet and a slave identification (ID) code. The nodes paged respond by sending their slave IDs back to the master node. When master node wants to create a connection involving layers above the link manager, the link manager sends a \textit{LMP\_host\_connection\_req} to the new slave node.
The slave node responds with a \textit{LMP\_accepted} protocol data unit.

Figure 5.2: Bluetooth also requires several distinct data transmissions to allow a node to join a piconet

5.1.3 SDD Join Method

SDD proposes using the unique time-hopping codes that channelize pulse radio UWB networks to aid in discovering new devices and adding them to the network. As shown in Figure 5.3, Node 1 begins the discovery process by sending \textit{inquiry} (IS) packets on a common time-hopping (TH) channel. The IS packet contains the TH code used for Node 1 transmission. Node 2 acquires the \textit{inquiry} and monitors the TH code for Node 1 transmissions. Node 2 waits for a random number of \textit{back off} slots before responding with an \textit{inquiry response} (IR) packet. Node 1 continuously scans for \textit{inquiry responses} while also periodically sending \textit{Low Rate Synchronization} (LRS) packets to any nodes that responded to the \textit{inquiry}. The IR packet contains the TH code for Node 2’s transmissions. Node 2 sends a \textit{request to send} (RTS) packet on the common TH code to Node 2. Node 2 responds with a \textit{clear to send} packet also on the common channel. Data transmission between Nodes 1 and 2 now proceeds on their respective transmit and receive TH code channels.

5.1.4 Analysis of Joining Protocol Suitability for UWB Networks

In order to analyze whether or not WTRP, Bluetooth, and SDD joining protocols are suitable for the UWB networks described in this thesis, we need to review some of the features of these networks. Because WTRP is a token ring protocol, it creates a controlled time division multiplexed network. The rules governing token passing, token hold times, token rotation times, and the other parameters needed for the token ring protocol enforce
fair and deterministic network behavior. Simply having a token at all prevents simultaneous transmission by multiple nodes.

WTRP starts network joining process by having a node in the ring broadcast messages soliciting nodes to join the ring. This process must be repeated every so often to give nodes a chance to join. In addition, the soliciting node must wait for a certain period of time after a solicitation to ensure that nodes wishing to join the ring have a chance to answer. Underlying carrier sense multiple access (CSMA) protocols used in conjunction with WTRP do not require significant time to synchronize between transmitting and receiving nodes. WTRP also relies on implicit acknowledgements to confirm successful token transmission [36]. Any packet "heard" by a node after token transmission that has the same ring address as the receiving node constitutes implicit acknowledgement. This is possible in WTRP because the underlying physical protocol allows nodes to "overhear" packets that are not necessarily sent to them.

Several features of the WTRP joining protocol are not well-suited to the UWB case. Having one node in the ring periodically solicit other nodes to join the ring requires one node to periodically drop out of the UWB ring in order to transmit the solicitation. This process incurs the overhead of acquisition times needed to reform the UWB ring when the soliciting node drops out and to readmit the soliciting node and any joining nodes. In addition, any solicitation wait times add to the delay of packets that need to get from the soliciting node to other nodes in the UWB network and to packets in the UWB node that need to get to the soliciting node. Using a token of any type to govern node transmissions in the UWB network works against the ability of the UWB nodes to transmit simultaneously using different TH or PN codes. The relatively large number of

Figure 5.3: SDD considers long acquisition times in attempting to provide a joining protocol specifically for UWB networks
data exchanges that occur in WTRP between the soliciting and joining nodes would incur delay due to acquisition times needed to establish these bidirectional links. Finally, implicit packet acknowledgement inherent in WTRP is not possible for pulsed type UWB networks because the unique TH or PN codes used to implement multiple channels prevent nodes from “overhearing” transmissions not intended for them.

Bluetooth uses frequency hopped spread spectrum with one packet sent per frequency hop. All communications in a piconet go through the master node. Slaves respond to master transmissions in a time division duplex manner, responding in the frequency hop directly after that used by the master node [49]. The master node initiates node joining by inquiring or paging. Once the FHSS sequence is known and master and slave nodes are connected, the protocol can be used to put slaves in park, sniff, or hold modes to preserve connections without sending much traffic. A sophisticated protocol stack allows establishment of asynchronous connectionless data transfers and synchronous connection-oriented data transfers above the link management layer. When the master node inquires and pages, it must send transmissions out on each of 32 possible frequency hopping wake-up channels to cover all potentially joining nodes [38]. Nodes that wish to join a piconet must also scan the specific frequency hopping channels to discover inquiries and pages.

Using the Bluetooth joining protocol is also not well-suited to the UWB network described in this thesis. If a master node in the ring must initiate joining, we increase average delay by allowing the node to periodically drop from the network to query for nearby nodes that want to join. Again, the relatively large number of data transfers between the nodes increases delay by requiring acquisition times to establish data links between the soliciting and joining nodes. For our UWB network, using a similar inquiry and scan process would be very slow because of the large number of UWB channels available via different TH or PN channel codes that would have to be transmitted or scanned.

The underlying networks discussed in papers [42], [43], [45] on the SDD protocol are somewhat different than the UWB ring networks described in this paper. For example, SDD networks are not ring networks, hence a desire in SDD for nodes to discover all other nodes in the immediate area and establish point to point links with them. Also, nodes in SDD networks set up essentially duplex communication on a point to point basis and then time division multiplex when talking to other nodes or when sending the LRS packets to keep links to other nodes synchronized. Like Bluetooth and WTRP networks, SDD uses active solicitation of new members, that is, one or more nodes already connected in a UWB
network must stop communicating on the network to solicit (inquire) for nodes that might want to join the network.

The biggest shortcoming with the underlying UWB networks using the SDD protocol is that the ability of all nodes to transmit simultaneously using different (possibly orthogonal) TH or PN code channels is not used. Instead, nodes communicate point to point and time division multiplex when sending traffic to other nodes. In addition, requiring one or more nodes that are active in a network to periodically leave the network to solicit new nodes increases average delay in the network. During the solicitation or inquiry time, the inquiring node cannot send, receive, or forward packets in the network.

Before describing the joining protocol we propose for this type of UWB network, it will be useful to consider again some of the underlying features of the UWB networks considered in this paper. Because the network is a ring, it is very easy to determine the ring topology by simply knowing predecessors and successors for every node. The predecessor and successor is known for every node because physical layer acquisition processes require use of TH or PN codes that are unique to the transmitters and receivers of each node. By choosing orthogonal TH or PN codes, we can enable simultaneous transmission of all nodes in the network, maximizing throughput. Any time the operating ring is perturbed by nodes joining and leaving, delay performance is adversely affected by the acquisition times needed to synchronize nodes prior to transmission of network traffic. We propose a joining protocol in the next section that is well-suited to the advantages and limitations of UWB networks.

### 5.1.5 Proposed UWB Ring Network Join Protocol

Because neither WTRP, Bluetooth, nor SDD provide the features needed in a UWB network joining protocol, we propose a new protocol called UWB Joining Protocol (UWBJP). In this section we will describe the operation of UWBJP and then describe its advantages and limitations. To prevent having to disrupt the ring to solicit joining nodes, we assign one of the UWB PN or TH codes for the joining channel. This is similar to the common channel specified in the SDD protocol. As shown in Figure 5.4, we add a joining manager to the simple UWB node previously described. The joining manager must be able to process the incoming radio frequency (RF) channel to determine whether any nodes are transmitting on the joining channel. This processing must be simultaneous with normal receive channel processing. Nodes that are soliciting to join a network will need to periodically send queries on the joining channel to see if a network is available.
These inquiring nodes apply a random time offset to these periodic requests to minimize the probability of collisions on joining channel. Nodes in the ring need a mechanism to elect one of the members to respond when a node is detected on the joining channel, so that all nodes in the ring don’t start executing the joining sequence. This could be implemented similarly to the Admission Control Manager (ACM) in WTRP, with a specific token to designate which node is the active Admission Control Manager. Each node can generate an ACM token to start with and then each node deletes tokens with a lower generation sequence than its own. The result would be one node as the ACM. If that node leaves, the ring, its predecessor can generate a new ACM token and take over the ACM functions. This is very similar to how ring owners are managed in WTRP [36].

With a workable method to elect and replace the Admission Control Manager, the UWBJP joining sequence shown in Figure 5.5 can be implemented. Node 2 enters the area and wants to join the ring network. Node 2 sends a join inquiry that is received by nodes in the ring listening to the joining channel, including Node 1, as shown in Figure 5.6(a). The join inquiry includes the PN code used by Node 2 for reception. Node 1, the designated node for answering joining inquiries, responds by sending a prepare for joining to Node 3. Once Node 1 receives ack prepare for joining from Node 3, it drops Node 3 and acquires Node 2 incurring an acquisition delay. Node 3 then sends an invitation to join to Node 2 that includes the receive PN code for Node 3 as shown in Figure 5.6(b). Node 2 then acquires Node 3, incurring an acquisition delay, and the ring is formed as shown in 5.6(c).

The UWB joining protocol must have a unique state for nodes when they receive a joining inquiry, but they are not already part of a network. This allows two unconnected nodes to form a network of two nodes using a variation of the joining protocol shown in Figure 5.5. If in the not-networked state, a node receiving a joining inquiry responds as shown in Figure 5.7. Node 2 sends its PN or TH code information and Node 1 acquires
Figure 5.5: Proposed UWB joining protocol packet exchange diagram

Figure 5.6: Network view of proposed UWB joining protocol
Node 2. Node 1 then sends its own PN or TH code to Node 2 and Node 2 acquires Node 1. The nodes then exchange data over their wireless links.

Figure 5.7: Proposed UWB joining protocol for first two nodes

Figure 5.8 shows the sequence of events from a network point of view. In Figure 5.8(a), both non-networked nodes send joining inquiries on the common joining channel searching for networks to join. The time interval between the sending joining inquiries in each node is selected from a uniform random distribution to reduce the probability of joining inquiry collisions. As shown in Figure 5.8(b) and (c), the node that first receives a valid joining inquiry, in this case Node 1, ceases its own joining inquiry transmissions and sends an invitation to join a network, which contains codes needed for synchronization. In Figure 5.8(c) and (d), Node 2 acquires Node 1 on its normal data transmission channel and forms a two-node network.

Figure 5.8: Network view of proposed UWB joining protocol

UWBJP is better suited to the UWB network than WTRP, Bluetooth, or SDD. The protocol does not require nodes already in the ring to incur acquisition delays by having to periodically leave the ring to check for nodes that want to join. Instead, nodes are able to
listen for a specific *join inquiry* from external nodes while continuing to receive and process data packets in the ring. UWBJP protocol minimizes two way data exchanges between the inquiring node and the inviting node in order to minimize acquisition delays. Finally, the protocol allows all nodes in the network to transmit simultaneously using different TH or PN coded channels because the nodes pass data in a ring, rather than forming point to point channels. The major drawback of UWBJP is that it requires a joining manager that can monitor the common joining channel, which adds complexity to the UWB nodes.

### 5.1.6 Join Protocol Performance Comparisons

We compared discovery overhead for WTRP, Bluetooth, SDD, and UWBJP protocols adapted to work with the UWB networks described in this thesis. We define discovery overhead as the fraction of time spent by the discovering node performing discovery and attachment of new nodes to the network, that is,

\[
\text{Discovery overhead} = \frac{T_{\text{discovery}}}{T_{\text{total}}},
\]

This parameter measures the fraction of operational time the different networks spend in acquiring and adding new nodes to the network.

First, we examined each protocol and constructed a sequence diagram showing how the protocol would work for discovering and adding a node to an existing network. We then developed equations for the discovery overhead of each protocol sequence. The joining sequence for WTRP is shown in Figure 5.9. In WTRP, Node 1 periodically solicits new network members. This solicitation occurs at times \( S_1 \) and \( S_2 \) in the diagram. Node 1 inquires for nodes during period \( I_1 \) and waits for a response for time \( W_1 \). Once a reply is made, Node 2 spends time \( \text{Acq}^{N_1} \) acquiring Node 1 to obtain network topology information and PN or TH codes needed to join the network. Note that the acquisition period \( \text{Acq}^{N_1} \) can run concurrently with the wait time \( W_1 \). Nodes 1 and 2 simultaneously acquire nodes in the network to reform the ring topology. At this point, data transfer begins. The joining protocol for Bluetooth was similar enough to that of WTRP (periodic solicitation of nodes wishing to join the network) that we consider the analysis for WTRP to apply equally to any "Bluetooth-like" joining protocol that we could apply to a ring network of UWB nodes.

We will assume for this calculation that times \( I_1 \) and \( W_1 \) are the same as the node
to node acquisition time, $t_{acq}$, in the network, that is,

\begin{align*}
I_1 &= t_{acq} \\
W_1 &= t_{acq} \\
Acq^{N1} &= t_{acq} \\
Acq^{N2-R} &= t_{acq} \\
Acq^{N1-R} &= t_{acq}.
\end{align*}

(5.2)

We also assume that even though the normal method of gaining members is through periodic solicitation, WTRP can respond to a node that simply tries to acquire the admission control manager node through a direct query. We assume that responding to and acquiring such nodes requires two acquisition time periods - one to acquire the controlling node and one to reform the ring with the new node. Defining the rate of join inquiries received as $R_{join}$, we note that the number of join inquiries during discovery interval $T_{int}$ is the product of the join rate and the discovery interval. So, during each discovery interval, two periods of time make up the overhead: the time needed for a full periodic solicitation ($3t_{acq}$) and the sum of join times resulting from join inquiries received during the discovery period, each of which uses $2t_{acq}$ of time. Thus, the equation for discovery overhead for the WTRP protocol is

\[
T_{OVHD}^{WTRP} = \frac{3t_{acq} + 2t_{acq}(R_{join}T_{int})}{T_{int}},
\]

(5.3)

where

\[
T_{int} = S_2 - S_1.
\]

(5.4)

We will use this equation later to compare discovery overhead of WTRP with the other protocols.

We lay out the joining sequence for the SDD protocol in a similar fashion as shown in Figure 5.10. Node 1 solicits new nodes on the common channel, waits for a response, and
sends LRS packets to maintain synchronization with any nodes discovered. Node 2 waits for a back off period before responding with an IR packet. Finally, Node 2 synchronizes with Node 1 on Node 1’s receive code. Node 1 sends an RTS packet; Node 2 replies with the CTS and then data is transferred. Acquisition of Node 1 by Node 2 on the common channel is assumed to occur during the time Node 1 sends inquiry $I_1$. We assume that SDD can also respond directly to an inquiring node that sends an inquiry and tries to acquire the admission controlling node. We assume this process requires only a single acquisition period, ignoring the very short time needed to process the inquiry on the common channel. We define the join inquiry rate $R_{\text{join}}$ as before, noting that the total number of joins resulting from these join inquiries is the product of $R_{\text{join}}$ and $T_{\text{int}}$.

$$R_{\text{join}} T_{\text{int}}$$

\[ T_{\text{SDD\text{overhead}}} = 2t_{\text{acq}} + \text{backoff} + t_{\text{acq}}(R_{\text{join}}T_{\text{int}}), \]

with $T_{\text{int}}$ given in (5.4).

As described earlier, UWBJP does not require a node in the network to leave the ring and solicit for nodes within range that wish to join the network. Instead, each node in the network can listen on a common code channel for join inquiries. One node is preselected to respond to such inquiries. As shown in Figure 5.11, once the inquiry is received, Node 1 acquires Node 2 to pass information needed to join the ring. At this point, both nodes acquire other nodes as needed to reform the ring using the normal node code channels. The first acquisition time for Node 1 to acquire the signals in the joining channel does not count as overhead because the Joining Monitor receives and interprets transmissions on the channel without interfering with normal data transmission in the ring.
Once information is gathered about the node wishing to join, Node 1 acquires Node 2, sends joining information, and Node 2 then acquires another node to reform the ring. Once the ring is reformed, data transmission resumes. The sequence is repeated each time a new node is heard on the joining channel. The result for UWBJP is a time penalty of \(2t_{acq}\) for every joining operation.

![Figure 5.11: UWBJP joining sequence for discovery overhead calculation](image)

As is evident from this description, the discovery overhead for UWBJP depends on the average interval between nodes arriving that want to join the network. We assume that the node performing the admission control management function in the network responds to each joining inquiry sent by arriving nodes. Given an average joining inquiry rate of \(R_{join}\) nodes per second, the average joining inquiry interval is

\[
T_{join\ inquiries} = \frac{1}{R_{join\ inquiries}}. \tag{5.6}
\]

Setting acquisition times equal to \(t_{acq}\) as before, we write

\[
T_{Overhead}^{UWBJP} = \frac{2t_{acq}}{T_{join\ inquiries}}. \tag{5.7}
\]

We evaluated these discovery overhead estimates using network parameters used in [45] and provided in Table 5.1.6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel acquisition time</td>
<td>0.1 ms</td>
</tr>
<tr>
<td>Slot duration for device discovery</td>
<td>20 (\mu s)</td>
</tr>
<tr>
<td>Discovery short interval (T_{int,short})</td>
<td>10 ms</td>
</tr>
<tr>
<td>Discovery long interval (T_{int,long})</td>
<td>1 s</td>
</tr>
<tr>
<td>Average back off time</td>
<td>7.5 slot durations</td>
</tr>
</tbody>
</table>

As shown in Figure 5.12, we plotted discovery overhead as a percentage for each protocol versus the join rates. Because WTRP and SDD periodically solicit new nodes in the area, they require additional discovery overhead that is independent of the number of joining
inquiries sent by nodes that want to join the network. So even when no nodes proactively try to join the network, WTRP and SDD still require overhead for their solicitation procedures. For both WTRP and SDD we plotted overhead values for both short and long discovery intervals.

As can be seen from the plots, discovery overhead for all protocols is quite good. The worst performing protocol is WTRP when using a short discovery interval with an overhead of three percent for a zero joining rate with a slight upward slope with increasing joining rate. For the case when the WTRP and SDD protocols use the short discovery interval, we see in Figure 5.12 that UWBJP requires significantly less discovery overhead. When WTRP and SDD use the long discovery interval, UWBJP has slightly less discovery overhead.

![Figure 5.12: UWBJP outperforms WTRP and SDD using the long or short discovery intervals](image)

5.2 Leaving a Ring

Ideally, network link management protocols should handle at least two instances of node departure - departure with notification and departure without notification. Departure with notification implies a node transition planned ahead of time with a protocol that allows graceful handling of the departure of the node from the network. Departure without
notification is normally associated with emergency action or node failures. Next, we discuss how the WTRP, Bluetooth, and SDD protocols handle nodes leaving the network with and without notification. The analysis revealed that there are not many different ways for nodes to leave the network, so we did not pursue performance analysis of the network leaving protocols.

5.2.1 WTRP Leaving Method

For WTRP, we consider the two cases of nodes leaving the ring with and without notification. Leaving with notification implies a planned departure when a node “knows” ahead of time that it is leaving the ring. Leaving without notification would normally imply some fault with a node has occurred that abruptly stops the ability of the node to communicate with the network. We will describe how the WTRP protocol handles each case, beginning with leaving with notification.

As mentioned previously, nodes may also leave WTRP networks without notification, for example, if a fault occurs that stops a node from communicating with its neighbors.
In Figure 5.13, assume Node 2 develops a fault and can no longer communicate. As discussed earlier, WTRP employs implicit acknowledgement for token transmissions. That is, after a node sends a token, it considers the transmission successful when it hears any packet with the same ring address as its own. When Node 1 no longer detects through implicit acknowledgement that Node 2 is present in the ring, it looks up the MAC address of Node 2’s successor in a connectivity table (maintained by the Connectivity Manager in each node [35]) and sends a set_predecessor token containing its own MAC address to that node to re-close the ring.

5.2.2 Bluetooth Leaving Method

As in WTRP, Bluetooth implements relatively simple protocols for nodes leaving a piconet. Again, we will describe the protocols for the two cases of leaving with notification and leaving without notification.

As shown in Figure 5.14, Node 2, a slave node detaches with notification by sending the LMP_detach protocol data unit to Node 1, the piconet master. A simple link level ARQ acknowledge completes the detachment. Because each piconet is a physical star with the master at the center, no additional connections need to be formed when a node detaches. Bluetooth has additional protocol features to handle the situation when the master node wishes to detach, including a way for nodes to switch their relative master and slave statuses. For leaving without notification, both master and slave nodes are equipped with a link supervision timer. If timeout period supervisionTO elapses without a device receiving a valid packet, the device will reset the link. This procedure is straightforward and no additional actions are needed such as reconfiguring the piconet to some new topology.
5.2.3 SDD Protocol Leaving Method

The SDD protocol, being primarily a device discovery protocol, does not provide a method for nodes to leave the network with notification. However, [45] briefly describes how a failed node would be detected and handled using SDD. Referring to Figures 5.3 and 5.15, if Node 1 does not receive a CTS packet within response interval $T_{response}$, it will attempt to resend the RTS packet for at most $m$ times. If there is still no response from Node 2, Node 1 will decide that Node 2 has failed or left the area. The protocol does not state whether or not Node 2 is maintained in Node 1’s neighbor list.

Figure 5.15: SDD detects failed or out of area nodes by multiple failures to respond to RTS packets

5.2.4 Analysis of Leaving Protocol Suitability for UWB Networks

In Section 5.1.4 we discussed many of the characteristics of the WTRP, Bluetooth, and SDD protocols that also apply when considering nodes leaving the UWB ring. As we saw, the leaving protocols are simpler than the joining protocols. In this section, we will discuss the shortcomings of these leaving protocols with regard to UWB networks. For the case of leaving with notification, WTRP’s reliance on implicit acknowledgement is a problem because UWB nodes cannot "listen" for messages from other nodes unless they are synchronized with those nodes. In the UWB network described, each the transmit and receive links for each node are synchronized only with the receive and transmit links of the node’s up-ring and down-ring neighbors. Explicit forms of acknowledgement will be required for the UWB ring network leave protocol.

The same limitation holds true for leaving without notification. In WTRP, nodes discover their down-ring neighbor is missing by listening for implicit acknowledgements. The UWB ring network leave protocol will need another method to determine when neighbor
nodes have gone missing.

On the other hand, the simple \textit{LMP}\textunderscore\textit{detach} and baseband \textit{ARQ Acknowledgement} used in the Bluetooth node disconnect request process can be easily applied to the UWB network. However, additional actions that are not required for Bluetooth piconets will be needed to re-form the UWB ring. For nodes leaving without notification, the UWB ring network can adopt something similar to the Bluetooth link supervision timer to recognize when nodes have left the area or suffered communications failures. In fact, the process of discovering a lost node in the UWB ring is much easier than in a piconet because the node downstream of the failed or wayward node will experience a complete absence of all traffic on its incoming link.

Considering that SDD does not describe a detachment with notification protocol, we will discuss only the node departure without notification described for the SDD protocol. Recalling that SDD bases decisions on missing or out of area nodes on failing to respond to RTS packets for \(m\) repetitions. Neglecting time to transmit the LRS and RTS packets shown in Figure 5.15, this means that Node 1 will discover that Node 2 is missing in a time period of approximately \(mT_{\text{response}}\). For parameters given for an SDD simulation study in [45], this equates to 1.2 ms - a considerable amount of time for a UWB network with 110 to 480 Mbps link capacity.

### 5.2.5 Proposed UWB Ring Network Leaving Protocol

As implied above, we propose a protocol that supports nodes leaving UWB ring networks that is simple and involves explicit rather than implicit acknowledgements. For node leaving the ring that can provide advance notification, the UWB Leaving Protocol (UWBLP) is shown in Figures 5.17 and 5.16. When Node 2 wants to leave the network, it sends a \textit{prepare for leaving} packet to Node 1 with the PN code for Node 3, its successor in the ring. Note that this packet must travel around the ring to reach Node 1. It is possible, but not necessary, to allow other nodes in the ring to read this packet in order to prepare for Node 2’s departure. Node 1 sends a simple acknowledgement of the \textit{prepare for leaving} packet back to Node 2 and then drops the link and begins the acquisition process with Node 3. Node 2 also drops its connection to Node 3 when it receives the \textit{prepare acknowledgement} from Node 1. Node 1 completes acquiring Node 3 and the ring is re-constituted.

In the case of a node leaving without notification, refer to Figures 5.18 and 5.19 and assume that Node 2 has gone missing. Node 3 finds this out because for some **supervisory**
time, Node 3 has received no packets from Node 2. If link level synchronization procedures for the UWB network require a periodic keep alive or beacon signal, Node 3 may discover that Node 2 is missing through lower level protocol stacks. In any event, when Node 3 discovers Node 2 is gone, it sends a set_successor packet to Node 1 containing its own address as shown in Figure 5.18(b). Note that the set_successor packet must traverse the entire ring to reach Node 1. Node 1 then acquires Node 3 as shown in Figures 5.18(b), (c) and Figure 5.19. UWBLP borrows the supervisory time from Bluetooth and the set_successor token (or packet) from WTRP to provide a simple way to handle nodes leaving the UWB ring without notice.

Figure 5.18: When Node 2 disappears without notice, the proposed UWB protocol can easily re-form the ring

We need additional mechanisms to recover when more than one node leaves the
network without notification. As can be seen in Figure 5.20, it is not enough for nodes to be able to detect when their immediate predecessors are gone. Nodes also need to be able to detect when their successors have left the ring. If Nodes 2 and 6 leave the ring simultaneously without notice, Nodes 3 and 5 immediately detect the loss because they do not receive any packets at all. However, if they try to send set_successor packets around the ring to the predecessors of the lost nodes as shown in Figure 5.20(b), these packets will not get through due to the loss of the nodes that normally would have forwarded the packets. Here we see the need for a different timer in each node to keep track of the last packet received that originated in that node's successor. So, when Node 3 sees no packets originating from Node 2 for a specific period of nothing_heard_from_successor_time, Node 3 assumes Node 2 has left the network and acquires Node 3 as shown in Figure 5.20(c). Node 4 follows the same operational pattern.

Figure 5.20: We need a successor timer to continue when two nodes leave the network without notification.

Even more complicated ways exist for nodes to leave the ring without notification. For example, two consecutive nodes may drop out of the ring. Or, half of the ring network may drop due to link failures. Such complex failures require even more management protocol.
features. One way to deal with the consecutive node drop problem is to use additional timers in each node. Consider the ring network in Figure 5.21(a). If Nodes 2 and 3 drop without notification, Node 4 detects the loss of traffic from Node 3 and sends a \textit{set\_successor} packet to Node 2. Each downstream node attempts to forward the packet and sets a \textit{nothing\_heard\_from\_Node2} timer to keep track of whether Node 4 succeeds in getting Node 2 to become its predecessor. Nodes further along the path towards Node 2 set shorter timers than nodes that are closer to Node 4. This ensures that the first \textit{nothing\_heard\_from\_Node2} timer to expire will be Node 1 as shown in the Figure 5.21(b). Node 1 then acquires Node 4, assuming that Node 2 has also dropped from the network, as shown in Figure 5.21(c). When Node 1 acquires Node 4, it sends a packet to shut off the \textit{nothing\_heard\_from\_Node2} timers running in the other nodes.

![Diagram](image_url)

Figure 5.21: When two consecutive nodes leave without notification, supervisory timers can support ring recovery

A similar approach can be used to recover when many consecutive nodes drop out of the ring. Consider the ring network in Figure 5.22(a). If Nodes 2 through 5 drop without notification, Node 6 detects the loss of traffic from Node 5 and sends a \textit{set\_successor} packet to Node 4. Each downstream node attempts to forward the packet and sets a \textit{nothing\_heard\_from\_Node4} timer to keep track of whether Node 6 succeeds in getting Node 4 to become its predecessor. Nodes further along the path towards Node 4 set shorter timers than nodes that are closer to Node 6. This ensures that the first \textit{nothing\_heard\_from\_Node4} timer to expire will be Node 1 as shown in the Figure 5.22(b). Node 1 acquires Node 6, assuming that Nodes 2 through 4 have dropped from the network. When Node 1 acquires Node 6, it sends a packet to shut off the \textit{nothing\_heard\_from\_Node4} timers running in the other nodes. Figure 5.22(c) shows Node 1 sending a packet defining the new ring order, which performs the dual purpose of informing nodes in the ring of the new topology as well as informing them to zero their \textit{nothing\_heard\_from\_Node4} timers.
Figure 5.22: Even when half of the nodes leave without notification, supervisory timers can support ring recovery.

There may be other more complicated circumstances from which the ring network must recover that require additional management protocol features. Enumerating all such situations is beyond the scope of this thesis. Our purpose has been to outline protocols for basic operation of this UWB ring network.

5.2.6 Summary

We researched three protocols supporting nodes joining and leaving ad hoc networks. These protocols were WTRP, Bluetooth, and SDD. WTRP and Bluetooth had limitations when used to support the UWB networks described in the thesis. Major problems included using active solicitation of new members, use of implicit acknowledgement for some packet deliveries, and a requirement to exchange many packets or tokens to add a new node to the network. Surprisingly, even SDD which was designed to support device discovery in UWB networks had shortcomings. Chief among these were an active solicitation policy for new members and protocols that end up requiring nodes to communicate in a time division method even though use of unique TH codes allows all network nodes to transmit and receive data simultaneously. We defined new protocols UWBJP and UW-BLP for allowing nodes to join and leave a UWB ring network. These protocols avoid the drawbacks of WTRP, Bluetooth, and SDD, but incorporate some of their features, such as time-based decisions for deciding when nodes have gone missing.

Our performance analysis for discovery overhead revealed that UWBJP performed with the least overhead. UWBJP outperformed the other protocols by a larger margin when short discovery intervals were used. When long discovery intervals were used, UWBJP per-
formed with slightly less overhead than WTRP and SDD. However, the discovery overhead of all protocols—even those with active member solicitation—was in all cases less than four percent over the range of joining rates analyzed.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

Given a traffic matrix specifying the average packet delivery rate between nodes and an aggregate capacity bound for all links, we studied the effects of topology, variable link capacities, and link acquisition times to determine how to minimize average delay in networks made up of simple UWB nodes. Our conclusions are as follows:

- For a specified traffic matrix, $T$, and an aggregate capacity bound, $B$, link capacities can be determined for a UWB ring network that minimize average delay.

- For a specified traffic matrix, $T$, and an aggregate capacity bound, $B$, there is probably a ring order that can be specified to minimize average delay. Finding the best ring order is NP-complete, so we must use heuristic methods for determining the best ring order.

- For a simple three node UWB network, switching between nodes incurs more delay than forwarding packets unless node acquisition times are very short.

- Adding single direction links to a UWB ring can improve delay performance without requiring significant additional complexity in all nodes.

- For a simple ten node UWB network, forwarding packets in the ring incurs less delay than using one node to switch traffic between the upper and lower halves of the ring. For very small acquisition times, the delay difference is negligible.
• We developed protocols to allow nodes to join and leave the ring that perform better for UWB networks than three protocols currently proposed or used in wireless networks.

6.2 Future Work

This thesis was primarily concerned with simple networks of UWB nodes having a single input and a single output link. Future research in this area should consider the case of UWB nodes with M incoming and N outgoing links. Topics of interest include finding node topologies to minimize delay, investigating the effect of aggregate capacity bounds on the network, and modeling the effect of node to node acquisition times on network performance. Future work might also consider multi-hop UWB networks where all nodes are not in the same physical domain. Developing management protocols for these complex networks is also a topic of interest.
Bibliography


106


