A BLUETOOTH SCATTERNET FORMATION MECHANISM BASED ON TRAFFIC DISTRIBUTION IN AN INFRASTRUCTURE NETWORK

A BLUETOOTH SCATTERNET FORMATION MECHANISM BASED ON TRAFFIC DISTRIBUTION IN AN INFRASTRUCTURE NETWORK

BY
PING AI, B. ENG

A THESIS
SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

McMaster University

MASTER OF APPLIED SCIENCE (2005)
(Electrical and Computer Engineering)

McMaster University
Hamilton, Ontario

TITLE: A Bluetooth Scatternet Formation Mechanism Based on Traffic Distribution in An Infrastructure Network

AUTHOR: Ping Ai, B. ENG (University of Electrical Science and Technology of China)

SUPERVISOR: Dr. Terence D. Todd

NUMBER OF PAGES: x,73

Abstract

Wireless communication has been thriving in recent years. Developments in the hardware and software industries enable more and more devices to be embedded in wireless communication modules. All kinds of interesting applications based on wireless connections are emerging, demanding simple and efficient ways to inter-connect different devices. Bluetooth is an industry standard initially proposed by Ericsson, IBM, Microsoft and some other leading IT companies to meet this growing demand. Initially, it intended to provide universal low cost, low power, and low complexity wireless interface to various devices. Furthermore, it also proposed to provide the possibility of interconnecting a number of mobile devices to form a network. However, the details of network formation and operation have not yet been regulated. In this work, we will investigate Bluetooth enabled network formation issues (especially when the traffic patterns on the network are well known).

In this thesis, we use a small indoor area network model with a wired infrastructure network installed in the wall. A number of mobiles are distributed in the area and require inter-connectivity with each other and/or the outside world through multiple gateways. Unbalanced traffic in the network may result in hotspots leading to poor network throughput. Therefore, a centralized network formation algorithm is needed for Bluetooth networks to solve this problem.

This thesis proposes novel Network Formation based on a Traffic Distribution (NFTD) mechanism. This centralized mechanism co-ordinates the behavior of mobiles and is implemented on gateways (also called access points). It forms the network topology according to the traffic distribution so that the path length of hotspot flows can be limited in order to maximize the network capacity. Last but not least, infrastructure networks provide free high-speed links for mobiles to further increase network capacity. The proposed mechanism is a promising mechanism as supported by simulation results.

Acknowledgments

First of all, I would like to thank Prof. Terence D. Todd, my master thesis advisor. I thank him for his wonderful lecture on wireless communication that initiated my interests in this field. I thank him for his support, encouragement and research ideas.

I also appreciate the help and support of Ms. Cheryl Gies, the secretary of the Communication Research Laboratory, and other administrative staff of the department. Also, this thesis would not have been possible without my colleagues and friends Nader Fahmy, Ming He, Dr. Tao Lin, Mohammed Smadi, Yun Wu, Jingxin Xue, Feng Zheng, etc. I thank them for their valuable discussions and advice.

I also owe a huge debt of gratitude to my parents. They have always been with me whenever I needed them. I would like to thank my supportive husband, and colleague, Xiaofan Wang. He has kept me focused and confident during my toughest times, and I thank him for all his support, encouragement and love.

Finally, I would thank my adorable baby son Daniel, who makes me a better person with stronger will and more enthusiasm for life. In addition, I am very grateful to all my friends who help me take care of Daniel so that I can have enough time to complete my thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Personal wireless communication has been a popular field for decades. Nowadays, people find themselves equipped with more electronic devices than ever before; from PCs, printers, and digital cameras, to cellphones, PDAs and MP3 players. While people are enjoying the great convenience these devices bring to them, cables that are sometimes needed to interconnect them can require significant management overhead. Therefore, there is a need for a low complexity, low cost, and low power wireless replacement for these cables.

Bluetooth (which is named after the Danish King Harald) was first introduced in 1997 by Ericsson as a universal air interface aiming to replace all kinds of cables. In the following year, an organization called the Bluetooth Special Interest Group (SIG) was founded to define Bluetooth specifications. Now, thousands of companies from all around the world participate as members in the Bluetooth SIG. The promoter companies include many leading telecommunication and computing companies such as Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia, and Toshiba. They published their first Bluetooth standard version 1.0, including a specification book and an application profile book, in 1999. The recent version, known as version 2.0 + Enhanced Data Rate (EDR), was developed in late 2004. The standard is Royalty-free and Bluetooth chips operate in the 2.4G license-free Industrial, Scientific and Medical (ISM) band. These are two factors that guarantee the low cost of Bluetooth products. Note that this thesis is based on Bluetooth specification version 1.0, since most of the work had been done before the newer versions came out.

Although Bluetooth is conceived as a simple cable replacement technology, it also enables networking of many mobiles. Basically, there are two scenarios of Bluetooth networking. One of them is the ad hoc networking scenario, which is also called a meeting scenario in [27]. This kind of network is supposed to involve dozens of Bluetooth enabled mobile nodes in a meeting room or other small indoor area, where an ad hoc network could be setup automatically as mobiles are approaching each other. Then, applications, such as file transfer and ID exchanging, can run inside this kind of networks. In the other application scenario, Bluetooth is used as a last 10 meter connection solution. In this case, portable devices can access larger networks, either wired or wireless, through access points. A Bluetooth access point should be equipped with both a Bluetooth interface and an interface to a larger network, and work as a gateway between the larger network and the Bluetooth network. According to the standard, one Bluetooth interface can only connect to seven other Bluetooth interfaces at the most, resulting in the rest of the devices in close proximity of the access points multi-hops away from the access point. We use this access scenario as the network model in our work. In the Bluetooth standard, a network composing of nodes sharing one radio channel is called a piconet. Two or more piconets inter-connecting together form a larger network called scatternet. Inter-connection of piconets can be achieved by piconets sharing some common nodes that switch between channels from time to time. However, details of scatternet formation and construction of corresponding topologies are not given in the standard so far. Researchers have examined possible scatternet formation methods for ad hoc Bluetooth networking [21] [2] [24] [3] [26] [16] [5]. But no formation method has been proposed for the access scenario. In the access scenario, the capacity of the resulting scatternet would be a very important issue because a bottleneck might be easily created within the scatternet. For example, if a large number of mobiles are connected with one access point in a tree-shape topology, all outgoing traffic has to go through the access point. Then it is highly possible that the access point will become a bottleneck node. This is an evident issue as more portable electronic products are equipped with Bluetooth chips. Therefore, forming Bluetooth network for the access scenario is an interesting open topic, one which we will investigate in this thesis.

We consider a system with four Bluetooth access points and a group of mobiles expecting to connect to the wired LAN nearby through these access points in an indoor area. The area considered in this system is small, therefore any node can choose any access point. In other words, no access point is beyond the reach of any mobile node. Within this system, a scatternet has to be formed so that each mobile has at least one route to reach at least one access point. Among the traffic flows carried in the network, some might be between a mobile node and another mobile node, while others might be between a mobile and an access point. We assume that the traffic is relatively stationary. In addition, a network may already exist in the same area which may have been formed by any previously proposed scatternet formation algorithm. However, the network may not be operating very efficiently, and the capacity may be low because of a poor topology. In this case a scatternet formation mechanism is needed to reform the network and improve its capacity. This can be done with the proposed algorithm using the traffic information collected over long periods of time. The mechanism can be centralized and executed by any of the access points, since the they have all the necessary information about each mobile, such as its ID, position and average load.In this thesis, a centralized scatternet mechanism is proposed, which can form proper network topology based on the carried traffic pattern. Simulations are used to investigate the performance of the mechanism. Also, the resulting scatternet topologies are analyzed to see how high speed wired links connecting access points can benefit the scatternet.

The rest of the thesis is organized as follows. The next chapter introduces the characteristics of wireless signals and the concerns in designing wireless communication networks. It also provides the background on the development of wireless communication applications. Chapter 3 outlines the Bluetooth specification, mainly baseband protocols of the standard, including link definition and setup procedure. Chapter 4 explores networking issues in the Bluetooth standard and examines previous works of scatternet formation and routing. Our novel Network Formation based on Traffic Distribution (NFTD) mechanism is then presented. Chapter 5 houses the analysis of our proposed algorithm and simulation results are presented. Finally, we conclude in Chapter 6, and provide some directions for future research based on this thesis.

Chapter 2

Background Review on Wireless Communication

This section gives a background review on wireless communication. Section 2.1 presents characteristics of wireless signals and transmission channels, which lead to the challenges in designing wireless networks. Then we go over some typical wireless communication networks, including cellular networks in Section 2.2, the second and the third generation mobile systems in Section 2.3, and wireless local area networks and wireless personal area networks in Section 2.4. In Section 2.5, some major techniques for wireless communications are introduced. They are digital signal modulation techniques, multiple access techniques, and network topologies. Lastly, Section 2.6 describes the outlook of wireless communication systems in the near future.

2.1 Wireless Signals and Channels

For wired networks, radio waves carrying data travel through a stationary and predictable channel, such as a steel wire or a fiber optics cable. Signals can be protected by shielded outer materials wrapping the wire, avoiding most of the outside electrical interferences. Thus, wired signals can easily travel a very long distance along a certain path. With the development of fiber optics, this distance is increased to thousands of miles without any serious signal loss. However, wired networks can not provide the flexibility that wireless networks can provide. For example, setting up wireless networks removes the need for extensive cabling and patching, which is both time-consuming and expensive. Moreover, wireless networks enable people to stay connected whenever and wherever they are. Therefore, wireless networks have been attracting communication researchers and manufacturers over decades.

Despite these advantages, wireless communications encounter more difficulties than wired communications. For a wireless communication channel, there is always a radio transmitter at one end and a receiver at the other end. Data is transmitted through air carried by radio frequency (RF) waves. Physical characteristics of wireless transmission medium are always time-varying, causing radio waves carried to behave unpredictably or at best be modelled in complicated situation-dependent models.

To better understand the properties of air carrier, three major types of radio wave propagations are introduced. They are recognized as reflection, diffraction, and scattering. Reflection occurs at the borderline when a radio wave is propagating from one media to another with different electrical properties. Some of the radio waves reflect back instead of entering the second media. When there is an obstacle in the way of propagation, if the size of the obstacle is comparable to the wave length, the radio wave bends around it (which is diffraction). When a radio wave is obstructed by an irregular surface, it scatters. All the signals produced by the scattering, reflections and diffractions, together with the original signal, add up vectorially at the receiver to result in a signal with different amplitude, phase and even frequency from that of the original signal - emphasizing the aforementioned unpredictability.

Under the unpredictable environment, the major concern for wireless signals is fading, the change of signal strength. There are two kinds of fading: the first one is called fast fading, and the other slow fading. Fast fading is the result of multi-path propagation. Slow fading refers to signal power attenuation after travelling across the distance between transmitter and receiver. It is usually modelled as inversely proportional to some exponent of the transmitter-receiver distance. Both fading effects have been researched and modelled intensively [20].

Due to the nature of wireless signals and devices, wireless communication development faces three major challenges. They are data rate limitations caused by the multi-path characteristics of radio propagation, the difficulties associated with signal coverage within buildings, and the need for low power electronic implementations suitable for portable terminals [7]. In spite of all these difficulties, many wireless communication systems have been successfully developed to meet the huge market demand for decades. Wireless communication systems outperform wired systems when mobility and flexibility are major concerns. Until today, wireless communication has been one of the hot areas in communication research and application fields. The following three sections introduce some major wireless systems in the market, such as cellular networks, wireless local area networks, and wireless personal area networks.

2.2 Cellular Networks

Before the advent of cellular networks, bandwidth limitation was a serious constraint in designing public wireless mobile networks. For example, the first public mobile telephone network in the U.S. could only hold a very limited number of users at the same time. This is because the total available bandwidth was narrow, and each user had to occupy 120KHz of bandwidth [20]. The concept of "cell" and bandwidth multiplexing is one of the milestones in the development history of wireless communication technology. It successfully solves the problem of limited RF bandwidth. In practice, the market of one company is divided into small areas called cells, where normally one group of radio frequencies is used for uplink voice transmissions and another group of totally different frequencies is used for downlink voice transmissions. In each cell, there is a RF tower called a base station sending and receiving radios with controlled power. The radio power is managed so that the signals are strong enough for the base station to communicate with mobiles located within the current cell, and at the same time, weak enough not to be a source of interference to the other adjacent cells. Then in the non-adjacent cells, these group of downlink and uplink frequencies can be reused. As shown in Figure 2.1, f stands for the group of downlink and uplink frequencies used in the area of the center cell. Several cells away, f is reused in another cell efficiently, since there is enough distance from the first cell to avoid interference.

With this cellular network concept, limited bandwidth can theoretically satisfy the need of mobile communication in any size area.

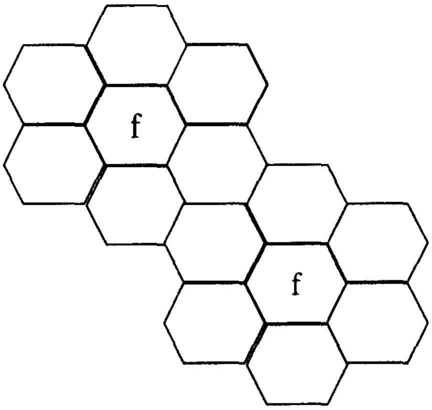


Figure 2.1: Frequency Reuse in Cellular Network

The typical cellular network appeared in the united states in the early 80s. It was called Analog Mobile Phone Service (AMPS). In AMPS, a total of 50MHz bandwidth were divided into 25 uplink channels and 25 download channels. Each channel was 30kHz [20]. Cellular networks in the other countries were all very similar to the AMPS.

Cellular networks from that time are now called the first generation of cellular networks. They are characterized by analog, meaning that voice and control signals are carried on an electrical wave using analog modulation. Beginning in the mid 1980s, the second generation of digital cellular networks gradually took over the market because of its advantages in better voice quality, higher system capacity, and more add-on services. The next section describes the second-generation and the emerging third-generation mobile systems.

2.3 The Second and the Third Generation Mobile Systems

The second generation (2G) cellular networks were extensively implemented in the world from late 1990s. In Europe, Global System for Mobile (GSM) was the first implemented digital cellular network. Other implementations of second generation cellular networks include Digital Advanced Mobiles Phone System (D-AMPS) and IS-95 CDMA in North America, and Japanese Digital Cellular (JDC) in Japan. Cordless telephone systems are also part of the second-generation mobile systems. Major technical standards for cordless phones are Digital European Cordless Telecommunications (DECT) and the Second-Generation Cordless Telephone (CT2), which were both developed in the Europe.

All these technologies differ greatly from the first generation cellular technologies because of their digital nature. Digital signals are much more reliable and robust to interferences compared to analog signals. Digital modulation methods, take frequency spectrum spread, for example, add reliability to radio signals with various frequencies as well. Actually, frequency spectrum spread is widely used in the 2G wireless networks, which to a certain extent solves the problem of limited bandwidth in the first-generation analog system. Simultaneously, the technical developments in hardware, especially the Very Large-Scale Integration (VLSI) technology have made the 2G mobile phones much smaller and much more affordable.

After cellular phones and other portable communication devices, such as PDAs with telephony functionality, became popular, the markets showed an increasing demand for better mobility and better services. In the mid 1980s, the term of the third generation (3G) mobile systems appeared. It targeted a mobile system with worldwide compatibility, enabling roaming to anywhere in the world without difficulty. It also provides much higher data transmission rates so that multimedia services, such as web browsing and video conferencing can be implemented. A couple of standards were already developed to carry out the 3G concept, such as the CDMA2000 and the Universal Mobile Telecommunication System (UMTS). The 3G technologies are reaching maturity today.

2.4 WLAN and WPAN

In the 1990s, wired local area networks were providing high speeds of hundreds of megabits per second, and became so popular that almost every business office and academic institution as well as some private citizens had it installed. At the same time, demand for wireless access was significantly growing due to two reasons. One was that some historic places could not afford to have holes in their walls to let cables be installed for wired local networks. Another reason was some personal devices such as notebooks and personal digital assistants (PDAs) became popular. They needed both convenience of local network access and its mobility.

To satisfy this new demand, the IEEE 802.11 specification was made as a standard that specified an "over-the-air" interface between a wireless client and a base station or access point, as well as among wireless clients. The corresponding wireless network is called the Wireless Local Area Network (WLAN), and the 802.11 standard is compatible with the IEEE 802.3 standard for Ethernet for wired LANs [10].

In the late 90s, significant hardware developments of a high level of circuit integration led to cheaper, smaller and "smarter" breed of electronic devices. WPAN refers to a network connecting all kinds of these "smart" electronic devices in close proximity, typically 10 meters. It can be a network including your desktop computer, the mouse, the keyboard, the printer, your PDA, etc. Or, it can be a network that connects all of your appliances at home. People can imagine numerous possibilities using WPAN technology. In May 1999, the IEEE WPAN Working Group was formed to work on a standard that supports WPAN. Bluetooth was adopted as part of their standards [12].

2.5 Wireless Communication Techniques

In this section, some of the techniques used by the aforementioned wireless communication systems are briefly introduced. These techniques include digital signal modulation techniques, multiple access techniques, and wireless network topologies. For in-depth knowledge on these topics, one can refer to [20].

2.5.1 Digital Signal Modulation

Signals are carried on radio waves to propagate. The process of putting signals onto the carrying radio is called modulation. The earliest and simplest method is amplitude modulation (AM), which is the technique that makes the amplitude envelope of a radio wave change according to the signal. The first generation cellular networks used AM technique. However, large bandwidth-consumption and noise vulnerability together with other reasons led to the limitation of the capacity and quality of analog cellular systems.

Digital modulation was then introduced to offer many advantages over analog modulation. Examples include greater noise immunity, robust to channel impairments, easier multiplexing of various forms of information, and greater security. Basically, digital modulation technics can be divided into two categories, linear modulation and non-linear modulation. The former includes Binary Phase Shift Keying(BPSK), Differential Phase Shift Keying(DPSK), Quadrature Phase Shift Keying(QPSK), and Offset QPSK. The latter includes Binary Frequency Shift Keying(BFSK), Minimum Shift Keying(MSK), and Gaussian Minimum Shift Keying(GMSK). There are other modulation techniques combining linear and non-linear technics together, such as M-ary phase Shift Keying(MPSK), M-ary Quadrature Amplitude Modulation(MQAM), and M-ary Frequency Shift Keying(MFSK). Gaussian Frequency Shift Keying (GFSK) is adopted by the Bluetooth standard because of its spectral efficiency.

In the mid 90s, spread spectrum modulation was proposed. This modulation is quite different from other digital modulation methods in the fact that it aims to employ a bandwidth that is several orders of magnitude greater than the minimum required signal bandwidth [20]. More importantly, a large number of users can use this relatively large bandwidth simultaneously without any significant interference if spread spectrum modulation is assigned for signals of each user. At this point, spread spectrum modulation is far more bandwidth efficient than other modulation techniques. This is one of the major reasons that it has been regulated by the Federal Communications Commission (FCC). Spread spectrum is to be used in the Industrial, Scientific and Medical (ISM) radio band, which is free of license and crowded by various radio systems.

Based on how and what pseudo-noise (PN) sequences are used, spread spectrum modulation can be grouped into two kinds of techniques - Direct Sequence Spread Spectrum(DS-SS) and Frequency Hopping Spread Spectrum(FH-SS). DS-SS system spread the baseband data by directly multiplying the baseband data with a pseudo-noise sequence that is produced by a pseudo-noise generator. In FH-SS system, frequency of data signal hops among a set of available frequencies, and the frequency hopping sequence is pseudo random. FH-SS has a relatively lower cost than DS-SS. The popular IEEE 802.11 standard uses DS-SS as its signal modulation method, and the Bluetooth standard, which is another popular new wireless technology, uses FH-SS [9]. For details of the spread spectrum modulations, please refer to [20].

2.5.2 Multiple Access

Similar to wired networks, multiple access is an important issue in wireless networks. Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) are two simple yet efficient ways to do multiple access, as they are used in wired networks. There are some other techniques developed only for wireless networks, such as Code Division Multiple Access (CDMA) and Space Division Multiple Access (SDMA).

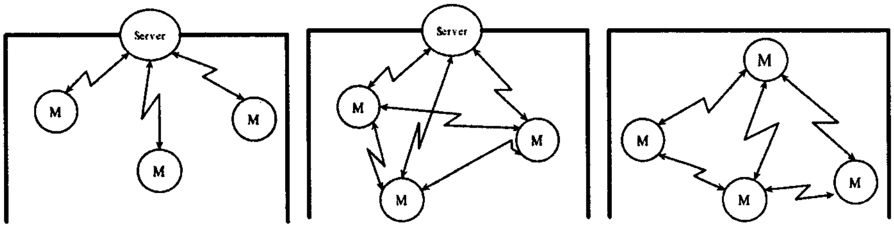
CDMA refers to the multiple access technology that signals are multiplied by different chip sequences. As long as the chip sequences are all orthogonal, these coded signals can be transmitted in the same radio channel and get extracted by the receiver. It is used in the 2G mobiles system in North America called IS95, and has even become the alias for this cellular system.

SDMA is a multiple access technology that utilizes the spatial separation of the users in order to optimize the use of the frequency spectrum. Together with the so-called smart-antenna technology, it can reuse frequencies in one cell by dividing it into several sections, which are sufficiently separated.

In practice, some hybrid combinations of these multiple access technics can be used to achieve better efficiency in some applications.

2.5.3 Wireless Network Topologies

Cellular networks have centralized topologies, shown in Figure 2.2(a), which are a kind of common topologies used in WLAN and some other wireless networks. It is robust and easy to maintain. However, this topology could be inefficient in capacity and delay, if most data traffic is not necessarily directed through base stations. Another more distributed topology is shown in Figure 2.2(b), which allows data transmission between any two mobiles instead of being relayed by a central server. The topology in Figure 2.2(c) presents a kind of wireless network without server, named ad hoc networks. All mobiles in an ad hoc network spontaneously discover and negotiate with each other before they inter-connect. Similar to the one shown in Figure 2.2(b), every mobile in Figure 2.2(c) has the ability to route data traffic for others.



(a) central (b) distributed (c) ad-hoc

Figure 2.2: Popular Wireless Network Topologies

2.6 Outlook of the Future

Two trends are noticeable in development of wireless networks. One is looking for a solution for high-speed wireless backbone networks that can carry multimedia data, such as video flows. The other, on the contrary, is a trend of a so-called last-meter solution that provides short-range and low-power connectivity to various electronic devices.

As an example of the high speed trend, the 3G cellular network technologies are maturing and moving towards implementation in many countries. Introduced in section 2.3, the 3G cellular networks have the capability to transmit multimedia traffic. In recent years, to achieve better transmission quality, a concept called the fourth generation (4G) cellular network emerged. It refers to technologies that provide more bandwidth than the 3G systems, and therefore enhanced and smoother multimedia transmission can be carried on it.

To implement the concept of the 3G and the 4G communications, IEEE has been working on several standards. The standard for Wireless Metropolitan Area Networks (WMANs) IEEE 802.16 is a broadband wireless network standard that provides up to 134 Mbit/s data transmission rate. Its aim is to provide a solution for fast local connection to network. IEEE 802.20 group is working on the standard for Mobile Broadband Wireless Access (MBWA), which can provide broadband and support high mobility to vehicular wireless networks. [11]

On the other hand, short-range low-power picocelluar networks have been on the track of fast development. In 2002, the Bluetooth specification was adopted by the IEEE WPAN group into their standard 802.15.1 as the 1Mb/s WPAN solution in the universal unlicensed band. IEEE 802.15.2 was then proposed in 2003 to address issues about coexistence of WPANs with other wireless devices operating in unlicensed frequency bands. In the same year, IEEE P802.15.3 was introduced to provide high rate WPAN standard for multimedia and digital imaging. Meanwhile, 802.15.4 standard was introduced as the specifications for low-rate wireless personal area networks (LR-WPANs). For details and latest development of WPAN technology, please refer to [12].

As standards are coming out, applications are booming to meet the tremendous demand from markets all over the world.

Chapter 3

Background Review on Bluetooth Technology

This chapter provides the background knowledge of Bluetooth technology. Section 3.1 gives a brief overview of Bluetooth development and the radio characteristics and protocol stacks offered by Bluetooth. Interfaces to other upper layer protocols are also presented in this section. Section 3.2 describes Bluetooth baseband, with emphasis on link establishment. Finally, Section 3.3 and Section 3.4 review the link management protocol and the Logical Link Control and Adaptation Protocol (L2CAP) respectively.

3.1 Overview

Bluetooth technology has emerged as a cable replacement solution. It aims at providing a universal air interface between a wide selection of electronic devices, such as computers, cellphones, digital cameras, speakers, printers, etc., physically separated by tens of meters. In 1999, a Special Interest Group (referred to as SIG hereafter) headed by industrial key players, such as Ericsson, IBM, Microsoft, and Intel, distributed Bluetooth specification version 1.0 royalty-free. Then, version 1.1 was released in 2001 correcting some errors in version 1.0 and version 1.2 released in 2003 with minor modifications on the audio link part [23].

The radio of a Bluetooth interface works at 2.4GHz in licence-free ISM band.

It is normally available in most countries in the world, thus reducing the cost and allowing wide deployment of the Bluetooth enabled products. Bluetooth spectrum ranges from 2.4000GHz to 2.4835GHz, and is divided into 79 channels of 1MHz each. FH-SS is adopted to utilize this bandwidth with efficiency and relatively low cost when compared to other spread spectrum technologies. Low power and robustness in a high interference radio environment are two of the major concerns when this technology was originally developed. This is because Bluetooth works in the regulated yet crowded ISM band. To control power consumption, Bluetooth defines three modes called sniff, hold, and park mode. Nodes operating in these three modes switch between "active" and "sleep" states using different schemes in each mode. Under normal operation and for the typical 10-meter transmitter-receiver distance, Bluetooth radios only consume 10mW of power, but can be increased up to 100mW for longer distance.

As mentioned earlier, cheaper and smaller Bluetooth chips make an interesting candidate for embedded devices, allowing new exploitation of new areas such as Wireless Personal Area Networks (WPANs). Ad hoc networking is not well described in Bluetooth specification so far. However, it is possible for a number of Bluetooth devices to be connected together, by one or more of them switching among different Bluetooth channels in a time sharing manner. These connected Bluetooth nodes sharing more than one 1MHz channels form a wireless network that is defined as a *scatternet* in the specification. So far, details of scatternet formation and operations are outside the scope of the present release of Bluetooth specifications. This issue remains as an open topic especially when the network performance relies on the network topology.

In the Bluetooth specification, protocols are divided into three Groups: transport protocols, middleware protocols, and applications. The protocol stack is shown in Figure 3.1.

Low level transport protocols are composed of two parts, the radio and the baseband. They can be regarded roughly as the physical layer and the link layer in the OSI seven-layer network model [13]. In the Bluetooth radio part, transmission frequencies, channel coding method, transmission power level, and other parameters are

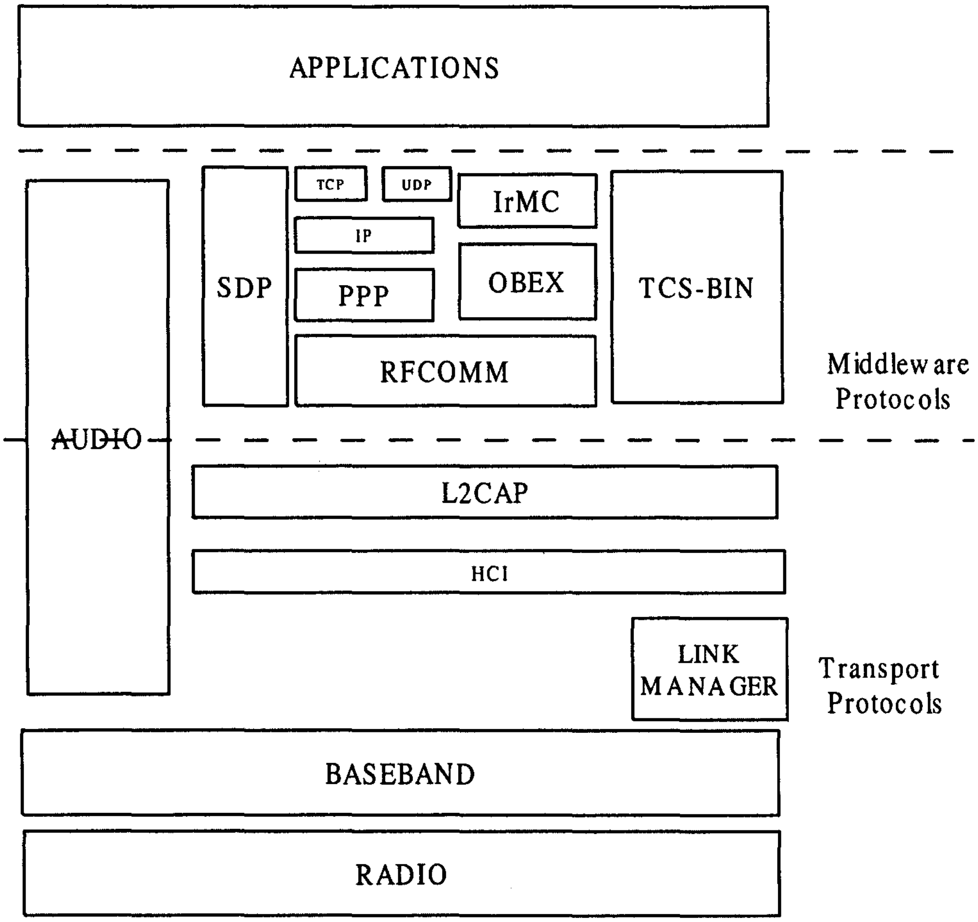


Figure 3.1: Bluetooth Protocol Stack

defined. Above the radio part, the baseband protocol regulates the packet format and transmission. Radio and baseband protocols are usually implemented in hardware inside a Bluetooth product, residing in one or two chips. Normally, there is another protocol layer called link management protocol in the Read-Only-Memory (ROM) of the product. Its accomplishments: setting up and breaking down the baseband link, and negotiating the grade of service and power level. The L2CAP is another upper level protocol in the transport protocol group. It aims at making Bluetooth transport protocols transparent to the applications. Notice that in Figure 3.1, there is a Host Controller Interface (HCI) layer below L2CAP layer. It is proposed to provide interoperability among host devices and Bluetooth modules. HCI is not a required part in the specification.

Middleware protocols include three Bluetooth specific protocols and other adopted protocols, such as TCP/IP, UDP, and Infrared Object Exchange (OBEX) protocol. They are all regarded as the networking layer in the terms of the OSI model. Those Bluetooth specific protocols include the Service Discovery Protocol (SDP), a serial port abstraction named RFCOMM protocol, and a telephony control protocol named TCS-BIN. SDP is used for two communicating peers to find each other - with one of them being a particular service provider (server) and the other being the client. This searching procedure is required because Bluetooth links are ad hoc- established. RFCOMM is used to facilitate the use of serial communications over Bluetooth wireless links. TCS-BIN is designed to support telephony functions. Since Bluetooth is proposed to provide a universal interface to connect devices using existing protocols, it is crucial to guarantee that current applications designed over these existing protocols can still operate in a Bluetooth-enabled device. IrDA interoperability protocols include OBEX and Infrared Mobile Communication (IrMC) for IrDA standard. Please refer to [19] and [9] for details of the middleware protocols.

Bridging by middleware protocols, most currently popular communication applications are ensured to be compatible with Bluetooth transportation layer. Applications based on Point-to-Point Protocols (PPP) can run over RFCOMM protocol. Applications based on TCP/IP protocols can access Bluetooth radio through PPP as well. Moreover, as the Bluetooth technology becomes mature, numerous new applications designed especially for it can surface very soon.

Besides data flows, audio flow can also be loaded on Bluetooth channels, such as voice and low-quality music. As shown in 3.1, audio flow is directly transferred upon Bluetooth baseband, without going through middleware protocols as data traffic.

In the next section, we focus on the baseband protocol, one of the major protocols in Bluetooth protocol stacks.

3.2 Bluetooth Baseband

Bluetooth baseband is a protocol layer right on top of the Bluetooth radio layer. Its functions include clock management, connection establishment, medium access control, power control, and security.

In Bluetooth radio, the frequency can hop very fast at normally 1600 times per second (in some states, 3200 times per second). Therefore, transmission time is divided into time slot of 0.625 msec each, during which the radio frequency does not change. In most cases, a baseband Protocol Data Unit (PDU), called a baseband packet, is transmitted within one time slot. Occasionally, a long baseband packet transmission could span 3 slots or 5 slots. However, during a transmission, the frequency remains unchanged even when the packet is longer than 0.625 msec. The frequency hops according to a pre-generated long frequency sequence, which is determined by the chip's address and its clock.

Baseband protocols support two types of links called Asynchronous Connection-Less (ACL) link and Synchronous Connection-Oriented (SCO) link. They are used for data transmission and audio Transmission, respectively. ACL links support transmission rates up to 1Mbps, while SCO links support 64Kbps voice transmission. Since the two ends of a Bluetooth link must operate with the same frequency hopping sequence and the same phase, they have to go through a certain procedure called *inquiry* to identify a common frequency sequence and frequency hopping phase. Then a *page* procedure is used to exchange information and setup an ACL link. An ACL link has to be established before a SCO link is set up between the two peers. One of the two ends of the link assumes control of the link, which is then called a master, while the other is called a slave. This master-slave structure is necessary since Bluetooth links are totally ad hoc. However, master-slave concept only exists in the baseband layer, and is transparent to peers at the upper layers.

A master can connect up to seven slaves, sharing one channel. Nodes sharing a channel are called a piconet. Time Division Duplex (TDD) and Time Divide Media Access (TDMA) are used in a piconet. The master polls a slave first and then that slave is allowed to transmit in a number of the following time slots to the master.

Bluetooth chips are generally embedded in small mobile devices, which are only equipped with limited power, such as battery power. Thus, power consumption becomes one of the major concerns in the standard. To save power, different modes are defined in the specification to force idle nodes into the sleeping state.

The next section explains different states and modes defined by the Bluetooth specification. After that, we present the medium access control mechanism in section 3.2.2. Then, section 3.2.3 describes the process of link establishment.

3.2.1 States and Modes

There are two basic states for Bluetooth nodes. When a node is engaged in a connection, it is in the connected state; otherwise, it is in the standby state. Standby state is the default state for a Bluetooth device. In this state, the device only keeps its clock running in a low-power mode. When a node is active in connected state, it consumes 1mW of power in a normal 10m distance transmission. However, a node may not always be transmitting because the traffic is heavy. To conserve power, it is necessary to make the nodes less active when there is no data for transmission. Depending on how active the node is involved in the connection, four modes are defined as active, sniff, hold, and park (with their activity decreasing in that order).

In the active mode, a node listens to the channel all the time, consuming the most power, typically 1mW. When a node decides to go to the sniff mode, it negotiates a sniff period with its master first and then stops listening to the channel. After that, that node only wakes up once every sniff period to check if there is any packet addressed to itself. If there is no data pending, it keeps listening for a certain number of time slots and then goes back to sleep; otherwise it becomes active again. If there is a long period of time with no data transmission between a slave and its master, this slave can enter hold mode. This state change process can be forced by master or be requested by the slave and then agreed by the master. When the slave is in hold mode, only the ACL link between the slave and its master is broken; the SCO link between them remains connected. Unlike those three modes above, nodes in park mode give up their active member addresses and can only synchronize with the master by getting broadcast packets. Putting some of the slaves into the park modes can increase the capacity of the piconet, as discussed in [15].

3.2.2 Medium Access Control

Bluetooth supports point-to-multipoint communication besides point-to-point communication. Up to seven slaves are allowed in a piconet, sharing the same channel resources. TDD is used for medium access control and the scheduling is managed by the master. Each active slave in the piconet has a 3-bit address assigned by the master. Basically, each slave gets some time slots in a time cycle, and the master only communicates with a slave during the time slots assigned to that slave. Once timeout occurs for the time slots assigned for one slave, the master switches to the next slave. Nodes in the park mode give up their active addresses so that it is possible to have more than seven slaves in one piconet, while the number of active nodes are still smaller than or equal to seven. It is one of the methods to improve the piconet capacity, as shown in Figure 3.2. The master node, marked as M in the center, has ten slave nodes in its piconet. Seven of them are in active mode and marked with letter A, while the other three slaves marked with letter P are parked at that time - temporarily unable to communicate with the master. After a period of time, the master could wake up the parked nodes, and pick three other nodes to be parked. Thus, all its slaves have some time scheduled to talk with the master and can have their data transmitted.

Clearly, the scheduling scheme used by masters is a crucial issue in piconet operation. It is unspecified in the standard and is currently an active area of research. The

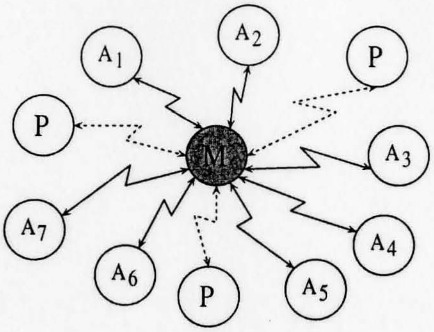


Figure 3.2: Park Mode

naive round robin scheduling is one of the scheduling schemes that can be used in Bluetooth piconet. However, it is not efficient in both throughput and power saving. S. Grag et al improved this round robin scheduling by introducing sniff modes into piconet [8]. Sniff interval and serving time are adjusted according to the backlogs of every master-slave pair. About 14% power reduction and significant throughput improvement are achieved. Focusing on piconet throughput and packet delay, M. Kalia et al proposed two other scheduling policies [14]. Both policies allocate time slots for master-slave pairs according to the binary backlog states of the master and the slave. In [4], A. Capone et al proposed several more practical scheduling schemes based on backlog awareness. In [17], T. Lin et al considered multi-slaves in a piconet simultaneously in its sniff scheduling schemes instead of treating each slave independently. S. Baatz et al explored scheduling schemes in scatternet in [1]. Since scheduling is not the primary interest in this thesis, please refer to these papers for further details.

3.2.3 Link Establishment

As mentioned at the beginning of this section, two procedures are related to link establishment in Bluetooth baseband. Inquiry procedure is for two nodes to find each other. Page procedure is used to setup an ACL link after the channel's frequency hopping sequence and phase are known.

In the inquiry procedure shown in Figure 3.3, node A inquires and node B listens to the channel. Both nodes operate on the same frequency hopping sequence, which is derived from a General Access Code (GAC) [9]. Node A sends an ID packet every 0.3125 msec, doubling the normal speed of 1600 hops per second. Node B listens to the channel, changing the frequency every 1.28 second, This frequency hopping speed is much slower than that in the inquiring state. However, node B does not keep listening throughout the 1.28 seconds. Instead, it only listens for a period of time $T_{w\_inquiryScan}$. The typical value for the parameter $T_{w\_inquiryScan}$ is 11.25 msec. Once both nodes happen to operate on the same frequency and scanning node B gets an ID packet from the inquiring node A, we say a "contact" occurs. When the first "contact" occurs, shown in Figure 3.3 as $T1$, the node in the inquiry scan state does not response with an ID packet immediately. This is because there might be other nodes undergoing the same situation at the same time. Collision would happen if more than one node send their ID packets to the same listening node simultaneously. To avoid the potential collision, node B backs off for a random number of time slots after the first "contact" occurs with node A. This random number ranges from 1 to 1023. Then, after node B resumes the inquiry scan process and the "contact"" occurs again, shown in Figure 3.3 at $T2$, it sends a FHS packet exactly 0.625 msec after the "contact" moment, at $T3$ in the figure. The FHS packet contains the sender's address, clock, and other important information. The inquiring node may continue inquiry, because there might be more than one node searching for connections.

Once the necessary information is received, nodes A and B might decide to setup an ACL link with each other. Then they have to enter the page procedure, during which node A enters the page state and node B enters the page scan state. Note that a page procedure does not have to follow an inquiry procedure immediately. The information obtained by nodes during an inquiry procedure can be stored into a local database and then nodes can enter a page procedure later. Figure 3.4 gives an example of a page procedure. Similar to the inquiry state, node A sends an ID packet every 0.3125 msec, changing its frequency every 0.3125 msec. Different from the inquiry procedure, the frequency sequence at node A is derived from the address and estimated clock of the prospective communication peer node B. On the other hand, node B has to be in the page scan state. In the page scan state, node B listens

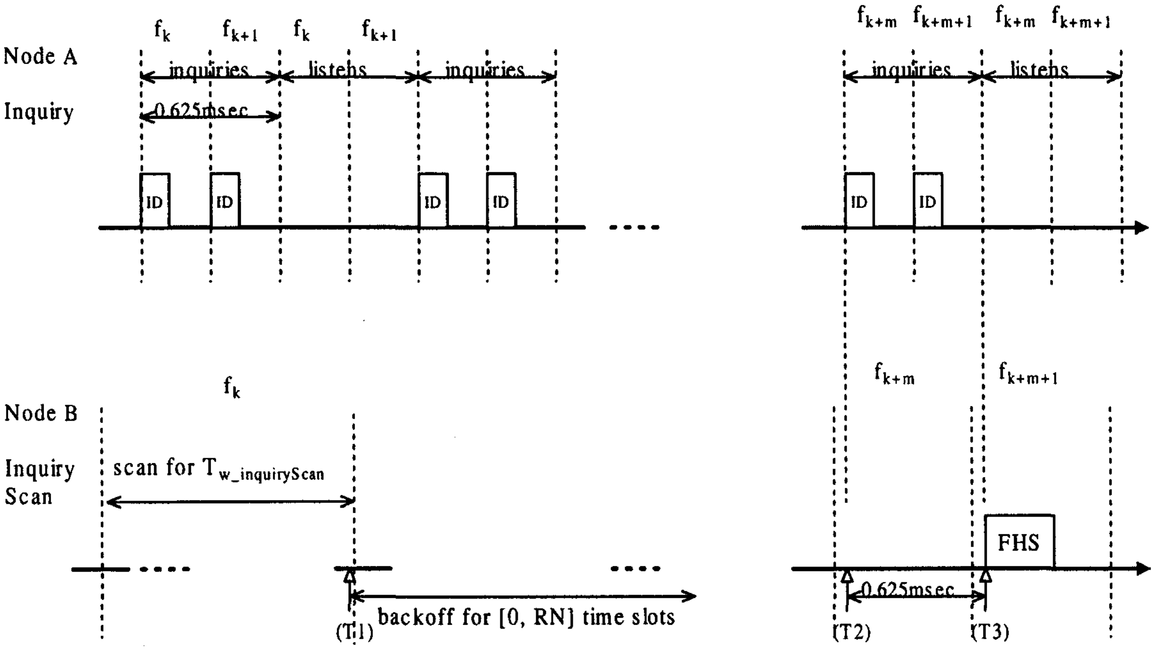


Figure 3.3: Inquiry Procedure

to the channel, changing its frequency normally every 1.28 seconds. Its frequency sequence is probably the same as that of node A. The frequency sequence is derived from node B's address and clock. Once a "contact" occurs at time T_1 shown in the figure, node B begins to synchronize its clock to node A. Then, 0.625 msec later, node B sends an ID packet to node A. Node A gets this packet and sends its FHS packet in the next time slot. The FHS packet includes some necessary information that node B needs to join the piconet, such as the Device access Code (DAC) and node B's 3-bit active member address. Again, node B responds by sending another ID packet notifying node A that the FHS packet is received. Finally, from time $T2$, both nodes change to the frequency sequence derived from the address and clock of the master, and begin to transmit data.

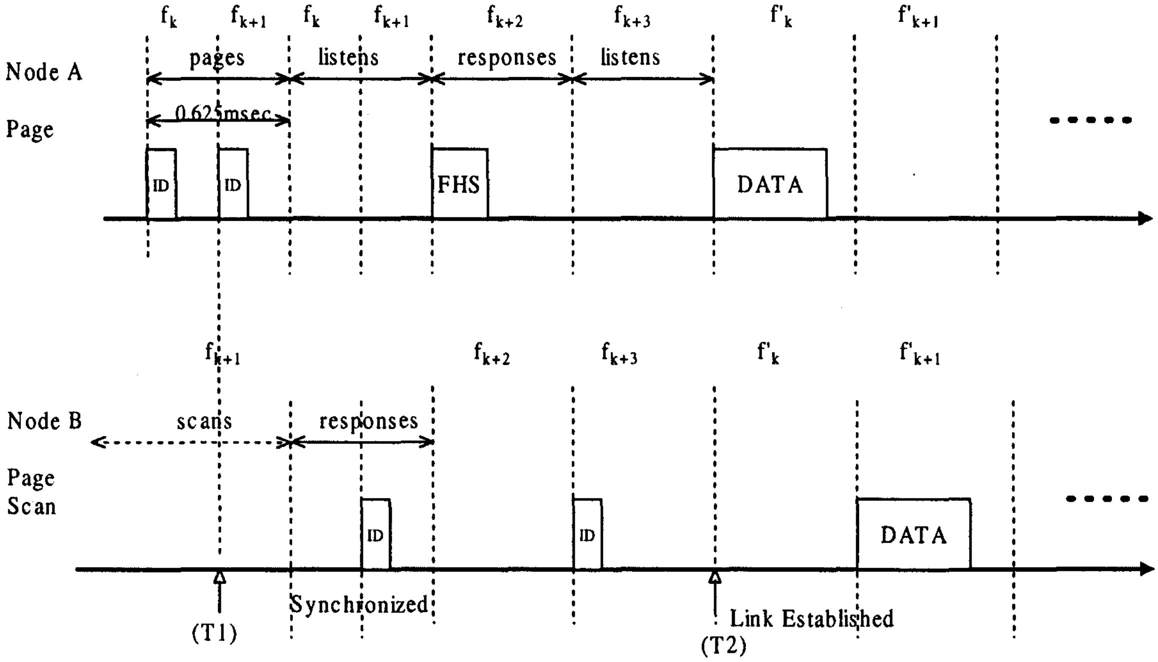


Figure 3.4: Page Procedure

A node in the page scan state first carries on a scan for 1.28 second. Typically, it uses 16 frequencies around the frequency it guesses. Normally, a "contact" should

Node State	Frequency Hopping Sequence Derived From	Access Code
Inquiry	GIAC and Own Clock	GIAC
Inquiry Scan	GIAC and Own Clock	GIAC
Page	Paged Node's Address and Estimated Clock	DAC
Page Scan	Own Address and Clock	DAC

Table 3.1: Parameters in Inquiry and Page

occur in this 1.28 second period if the "guess" is efficient. Otherwise, a "contact" should occur in the next 1.28 seconds when the other 16 frequencies are used in page scan. Thus, a page procedure does not exceed 2.56 seconds. In other words, it can be much faster for a "contact" to occur during a page procedure than during an inquiry procedure. (The worst case for the length of the inquiry procedure to last 10.24 seconds [22].)

The crucial parameters in inquiry and page procedures are summarized in Table 3.1.

3.3 Link Manager

As shown in Figure 3.1, there is a layer right on top of the baseband layer, named the link manager. A protocol called Link Management Protocol (LMP) implements the functions in this layer. This protocol is responsible for QoS bandwidth allocation and voice traffic. It also authenticates the device pair by generating and storing an authentication key. Power control is one of the functions of link manager as well. It manages the mode of the baseband, forcing it into a less active mode, saving power when necessary.

3.4 Logical Link Control and Adaptation Protocol

The upmost layer in the transportation protocol group is called logical link control and adaptation protocol (L2CAP), as shown in Figure 3.1. Unlike other protocols in the transportation group, it is normally implemented in software. It aims at hiding the details of the baseband PDU and transmissions from upper-layer protocols. It helps segmenting/asssembling data from/to upper layer, since PDUs in the application layer are always much longer than Bluetooth baseband packets and not suitable for transmission in baseband directly. Also, it facilitates the maintenance of QoS admission controls. In L2CAP, communication tunnels between two entities are called channels. (while they are called links in the baseband.) Two ends of a channel are equal without any master and slave roles.

After reviewing the basic concept and protocols of Bluetooth technology, we address the issue of Bluetooth networking in the next chapter and then propose our Network Formation based on Traffic Distribution (NFTD) mechanism.

Chapter 4

Bluetooth Scatternet Formation

This chapter proposes the Bluetooth Network Formation based on Traffic Distribution (NFTD) mechanism in the infrastructure network. Before going into the details of the mechanism, we first introduce the terms of piconet and scatternet defined in the Bluetooth standard. Some previous work on the topic of Bluetooth scatternet formation is then presented. In Section 4.2, we describe the network model considered in this thesis, address the challenges of network formation problem in the model and some viable solutions. After that, the NFTD mechanism is presented in detail in Section 4.3. Last, in Section 4.4, a network formation algorithm called Traffic Unaware Algorithm (TUA) is introduced. The TUA works similar to the most of formation algorithms proposed in literature. It is used to compare with the NFTD mechanism in Chapter 5.

4.1 Bluetooth Scatternet

Despite the initial aim of simply replacing cables between electronic devices, the demand for inter-piconet connectivity arises as more and more devices become equipped with Bluetooth chips. As stated in Chapter 3, any node in a piconet is able to communicate with other nodes in the same piconet, either directly (between a master and its slaves) or indirectly (between slaves via their master). The maximum number of active nodes in one piconet is limited to eight. Moreover, the specification also allows multiple piconets to inter-connect and form a larger network defined as a scatternet.

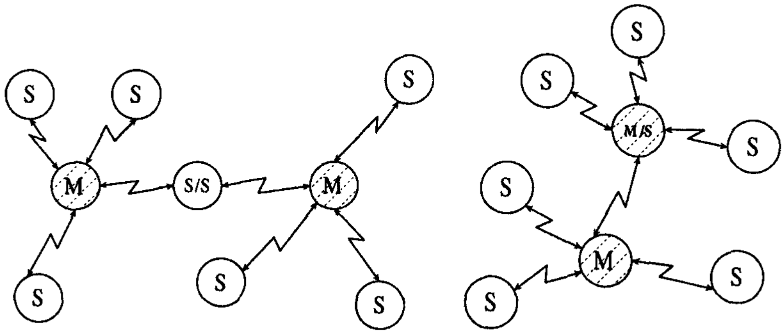
The usage of scatternets in an access scenario can be two-fold. One is to expand capacity of an access point. For example, more than seven devices are expecting transmission through an access point simultaneously. Installing multiple Bluetooth chips in the access point would be an option to solve this problem. However, interferences have to be managed very carefully [6] [25]. Otherwise, multiple chips can not work together properly in very close proximity. Thus, another practical way is to form a scatternet among these devices and the access point. The other usage of scatternets is access point range extension. If the devices are not all within the radio range of an access point and still in need of transmissions with that access point, forming a scatternet can solve the problem, since any node in the scatternet can communicate with the access point directly or through some intermediate nodes. In this thesis, we focus on the capacity expansion usage.

The rest of the section is organized as follows. Section 4.1.1 introduces the piconet and the scatternet definitions in the Bluetooth specification, and outlines the open issues of scatternet formation. In Section 4.1.2, related work is presented.

4.1.1 Scatternet in Bluetooth Specification

Inter-piconet connection is enabled in the specification by allowing a node to participate in multiple piconets. As shown in Figure 4.1, the node that is active in two piconets acts as a bridge between the two piconets, resulting in a network consisting of multiple piconets called a *scatternet.* The roles of the bridge node in different piconets vary depending on the way the piconets are connected. In the scatternet shown in Figure 4.1(a), the bridge is a slave in both piconets. In the scatternet shown in 4.1(b), the bridge is a master in one piconet but a slave in the other. These two scatternets are among the simplest scatternets, since only two piconets are involved and also only two roles are held by the bridge node. The specification does not define an upper limit on the number of roles a node can have and the number of piconets a node can participate in. There is even no further description about scatternet topology and routing, which can be complicated, since any topology is possible as long as the size of each piconet is constrained to eight nodes.

While the emergence of scatternets seems to be practical in the near future, the issues of scatternet topology formation and scatternet routing still remain undefined in the specification. Numerous papers tried to address it in different angels and aspects. Some previous works among them are briefly reviewed in the next subsection.



(a) slave bridge (b) master bridge

Figure 4.1: Scatternet Definition

4.1.2 Scatternet Topology and Formation

In [18], G. Miklos et al from Ericsson examined the relation between scatternet parameters and performance. Two different centralized scatternet formation algorithms were proposed and one of them was traffic dependant. An algorithm to form tree-shape scatternets was discussed in [26]. However, a tree topology is not robust in case of node failure and might not be efficient in routing. In [16], a distributed algorithm was presented to minimize the roles of bridge nodes and the number of piconets, so that a relatively high performance of the resulting piconets can be achieved. Another distributed algorithm was introduced in [24], which also succeeded in controlling the scatternet diameter (the longest hop counts between two nodes in the scatternet) and the size of piconets. Moreover, it developed a valuable metric to judge scatternet performance. It was called average shortest path, which was the average shortest path-length (hop count) among all node pairs in a scatternet. This metric reflected average routing length so that mean packet delay could be restrained. It is adopted in this thesis as an important scatternet performance matric in Chapter 5. In [3], separated piconets (called *bluestars*) were generated in phase I and then in phase II, bluestars were treated as virtual nodes to form a scatternet (called *blueconstellation*). The concept of virtual nodes is valuable and adopted in this thesis as well.

4.2 Scatternet Topology in Infrastructure Network

In our work, we focus on the Bluetooth access scenario, which is illustrated in Figure 4.2. A wired infrastructure network and a number of Bluetooth-enabled devices reside in a small area, usually indoor. The bold lines around the walls are wired infrastructure network. It can be any kind of wired network, such as a Local Area Network (LAN), providing broad bandwidth and stationary connections between access points along the wall. On each corner of the indoor area, there is a Bluetooth access point. The access points are equipped with both a Bluetooth interface and a wired network interface. The circles marked with letter B are Bluetooth mobile devices. They are distributed in this area and need to exchange data with devices outside the area through the access points. They might demand data exchange among themselves occasionally as well.

In order to access outside networks, all mobiles have to connect to one of the access points in the area by one or more hops. If less than eight mobiles are associated to an access point, these mobiles and the access point form a piconet. If eight or more than eight mobiles are associated to one access point, they together form a scatternet. Unlike the meeting scenario, scatternet formation of the mobiles are not necessarily completely ad hoc, because access points can be of great help in the process of scatternet formation and data transmissions after that. Firstly, an infrastructure network provides ready-to-use links for the access points to quickly exchange all kinds of information about the mobiles in the area, such as their IDs, positions, traffic loads. Thus, a centralized scatternet formation algorithm can be executed by any access point given that information. Secondly, the high speed links between the access points can also be used as free links for data transmission, since their bandwidth is hundreds

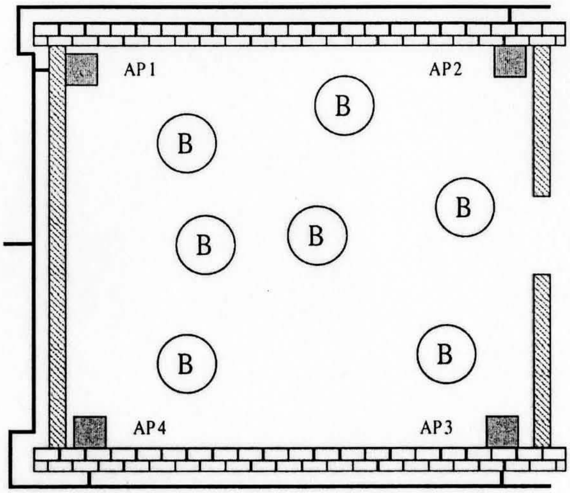


Figure 4.2: Infrastructure Network Scenario

or thousands of Mbps, which is large when compared to 1Mbps of Bluetooth ACL link bandwidth. As stated in previous sections, no algorithms have been proposed to address the network formation problem in a Bluetooth infrastructure network. Thus, a novel mechanism is needed to solve this problem.

From the point of view of traffic distribution, the access scenario can be characterized as a traffic pattern with some hotspots. Each hotspot consists of an access point and the mobile devices that are accessing the outside network through this access point. Besides access demands, some mobiles can also demand data exchange with any other mobiles in the area. Some mobile devices can even act as a server to provide services to a couple of other mobile devices. Thus, the unbalance existing in the inter-mobile traffic can potentially lead to hotspots in the traffic distribution. Therefore, the network formation mechanism proposed in this thesis takes the traffic distribution into consideration to achieve a relative high network capacity.

The model investigated in the following paragraphs assumes that all mobiles and the access points in this scenario are within the radio range of each other. There is no obstacle between any two nodes, and no hidden terminal effects are considered. To further simplify the problem, mobiles are assumed to be relatively fixed in position, which is reasonable since the emphasis of this thesis is to find a proper topology for a certain 2-dimensional distribution pattern of nodes. Given the demand traffic flows between any mobile pair, the following section gives a centralized algorithm to form scatternets with high capacity and relative low packet delay.

There are two major strategies in the algorithm. Firstly, mobiles with high volume of traffic among them are grouped in one piconet if possible. Otherwise, high traffic flows have to go through more hops and burden the intermediate nodes along the routes, so that capacity of the whole network could be significantly impaired. Secondly, once piconets are formed according to traffic hotspots, they are considered as virtual nodes. All traffic flows from a mobile in one piconet to another mobile in a different piconet are handled as flows between two virtual nodes. These virtual nodes then form "piconets" without overlapping. After the "piconet" formation repeating a certain number of iterations, there should be only two virtual nodes left, and connecting them together produces a full connectivity network. The details of the proposed Network Formation based on Traffic Distribution (NFTD) mechanism are explained in the next section.

4.3 Network Formation Mechanism Based on Traffic Distribution

The mechanism consists of a main procedure and two subroutines. Descriptions and pseudo-code are provided in the following subsections. First, Section 4.3.1 describes the main procedure and introduces some terms used in this section. Other supported subroutines are explained in Section 4.3.2 and 4.3.3 respectively.

4.3.1 Main Procedure

The main procedure is to call the *cluster* subroutine repeatedly, until a fully connected network is generated. The cluster subroutine is to group isolated network components, which is addressed in detail in Section 4.3.2. To better describe the algorithm, we define several terms. At the beginning of the l-th iteration in the main procedure, the isolated network components are called level-$(l-1)$ nodes. The set of level l nodes is presented by $N(l)$. The number of level-l nodes is denoted by $N_c(l)$. In a mathematical equation to follow, $N_c(l) = |N(l)|$. For example, at the beginning of the 1st iteration, all mobiles are separated. So the isolated network components are the isolated mobile nodes, which are called level-0 nodes. Then, during the 1st iteration, the cluster subroutine is called to group nodes into a number of piconets. Thus, at the beginning of the 2nd iteration, the isolated network components are the piconets and they are considered to be virtual nodes. These virtual nodes are called level-1 nodes. At the end of the 2nd iteration, level-1 nodes are grouped to new virtual node that are called level-2 node. The main procedure ends when $N_c(l) = 1$, which means all nodes are fully connected. The cluster subroutine guarantees that $N_c(l) < N_c(l-1)$, as explained in Section 4.3.2. Thus, the main procedure converges when l is large enough.

The following is the pseudo-codes for the main procedure.

```
main():
    set $N_c(l)$ = number of real nodes
    set $l = 0$
    while ($N_c(l) > 1$)
    begin
        call subroutine $Cluster(N(l))$
        $l = l + 1$
    end
```

4.3.2 Cluster Subroutine

The cluster subroutine is to group level-l nodes together to form a number of new level-$(l+1)$ nodes. In other words, this subroutine takes $N(l)$ as input, connect components in $N(l)$, and generates $N(l+1)$. Since at least two components in $N(l)$

are connected, $N_c(l+1) \leq N_c(l) - 1$. Therefore, $N_c(l+1) < N_c(l)$, guaranteeing the convergence of the main procedure in Section 4.3.1.

The pseudo-code for the cluster subroutine is at the end of this subsection. Here we introduce the terms used in the pseudo-code first. N_i is the node with ID i in the current level, whose *role* can be NONE, MASTER or SLAVE. Ld_i is the estimated load on a specific node N_i. FNS represents the set of "free" nodes. A node is "free" when it is not connected to any other node and its role is NONE. T represents the traffic matrix for nodes in the current level. $T_{i,j}$ is the amount of traffic between nodes N_i and N_j. S_{max} is the maximum number of slaves a master can have in a piconet.

In the pseudo-code, Line 1 to Line 5 initialize FNS and calculates Ld_i for each i. From Line 6, the subroutine begins to connect the level-l nodes in $N(l)$.

Line 6 and Line 7 determine how many nodes should be in a new cluster. Level-0 nodes are real mobile nodes and clustering them forms piconets. Therefore, a level-0 node can connect S_{max} nodes to form a cluster. However, to avoid large clusters in the higher levels, only two virtual nodes are allowed in a cluster, and one virtual link has to be setup between them. For example, in level 0, if there are 20 nodes and up to 6 nodes are allowed in a piconet ($S_{max} = 5$), the subroutine builds $\lceil \frac{20}{6} \rceil = 4$ level-1 nodes. Among them, 3 of them are piconets consisting of 6 mobiles each and the fourth is a piconet with only 2 mobiles in it. Then, by calling cluster the routine again, these 4 level-1 nodes forms $\lceil \frac{4}{2} \rceil = 2$ level-2 nodes, since only two virtual nodes are allowed in one cluster.

The cluster subroutine builds new virtual nodes one by one. Once a new level-$(l+1)$ node is built, the level-l nodes in it are all removed from the free node list FHS. Thus, only free nodes are considered to form the next new level-$(l+1)$ node. The decision of which nodes should be grouped into a new level-$(l+1)$ node is made according to the traffic pattern between all the level-l nodes in FNS.

Line 8 and Line 9 choose the master for the cluster, which is the node with heaviest load (total outgoing and ingoing traffic to all other nodes). In case of a tie, a node is randomly picked. (This rule for breaking a tie is used in the rest of the mechanism.) Then, in Line 10, the master node is removed from FNS.

Line 11 to Line 16 choose this master's slave(s) according to traffic and guarantee that nodes that have relatively high traffic with a master get priority of being associated to that master than to other nodes. The selected slave node is then removed from the *FNS*. Line 17 calls the *connect* subroutine presented in Section 4.3.3 to actually make links between master and its slave(s).

The following is the pseudo-code for the cluster subroutine.

```
    Cluster(N(l)):
1   $FNS$ = N(l)
2   for each node $N_i$ in $FNS$
3       begin
4       set $Ld_i$ = total traffic between $N_i$ and all other nodes in $FNS$
5       end
6       if $l = 0$ then set $n = S_{max}$
7           else set $n = 1$
8       set $msr\_id$ = ID of the node in $FNS$ that has the heaviest estimate load
9       set role of $N_{msr\_id}$ = MASTER
10      remove $N_{msr\_id}$ from $FNS$
11      set $Counter = 0$
12      while $Counter < n$ or no more unmarked node in $FNS$
13          begin
14          set $id$ = ID of the node in $FNS$ that has the heaviest traffic with $N_i$
15          set role of $N_{slv\_id}$ = SLAVE
16          remove $N_{slv\_id}$ from $FNS$
17          call subroutine $Connect(l - 1, msr\_id, slv\_id)$
18          $Counter$ increased by 1
19          end
20      end
```

4.3.3 Connect Subroutine

This subroutine connects two level-l nodes, and forms a level-$(l+1)$ node. The IDs of two nodes i and j, together with l are the three inputs of the connect subroutine. If l is 0, it makes an ACL link between them. Otherwise, it sets up a virtual link between virtual nodes. The pseudo-code for the subroutine is presented at the end of this subsection

To better describe the procedure, we call a virtual link between two level-l nodes a level-l link. Therefore, a level-0 link is a real ACL link. A level-l virtual link between virtual nodes A and B is actually implemented by one or more real ACL links between two real nodes included in A and B respectively. The connect subroutine is also responsible for setting up the real ACL links when it sets up a virtual level-l link ($l > 0$). This is done by recursive calls.

Before we move to the details of the subroutine, some terms are introduced. C is the connectivity matrix for the level-0 nodes that describes the resultant network topology. $C_{i,j} = 1$ means that there is an ACL link between N_i and N_j, with N_i be the master and N_j be the slave. Otherwise, $C_{i,j} = 0$. $N_l(l)$ denotes the maximum number of links allowed between any two level-l nodes.

In the pseudo-code, line 1 to line 4 deal with connection for 2 level-0 nodes. In this case, an ACL link is setup by setting the corresponding entry in the connectivity matrix to 1.

Line 5 to line 9 deal with connections between two level-1 nodes, which are piconets. In this case, a node in the two piconets has to be selected to act as a bridge. Generally, there are two ways to do this, as shown in Figure 4.3. If a bridge is chosen in the way that it acts as a master in piconet A and a slave in piconet B (called a M/S bridge in this thesis), all the activities in piconet A have to be paused once the M/S bridge switches to piconet B. This is obviously not efficient. Thus, in this algorithm, only S/S bridges are allowed. Back to Figure 4.3, for the above reason, the possible links to connect piconet A and piconet B are M_A to all slaves of piconet B and M_B to all slaves of piconet A. The traffic flows on these possible links are compared, and the link with the heaviest flow is chosen to be the link to build.

Lines 10 to line 18 take care of the case for connections between two level-l nodes

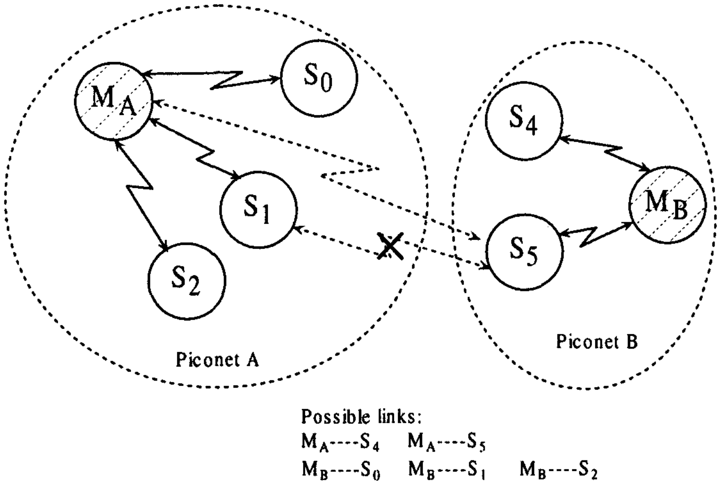


Figure 4.3: Possible Links Between Two Piconets

$(l > 1)$. Line 12 to Line 15 determine the number of level-$(l-1)$ links needed to be made between members of these two level-l nodes. Notice that the higher the level is, the more real mobiles are included in a virtual node. The number of links is designed to increase as the level goes higher. Therefore, we set $N_l(l)$ to 1 in Line 15. As always, these links are chosen according to the traffic distribution. There is a special case when S_{max} equals to six. As shown in Figure 4.4, piconet 3 and piconet 4 will be saturated if two level-1 links between two level-2 nodes are brought up. Consequently, all the four piconets will be saturated, because piconet 1 and piconet 2 are already saturated. Then, when vitual node 1 and virtual node 2 are regarded as one virtual node A in level-3, and same thing happens to another virtual node B in level-3 as well, A and B cannot be interconnected. Therefore, to avoid this problem, $N_l(l)$ is always set to 1 in each level above level-0 when $S_{max} = 6$, as Line 13 does.

Table 4.1 makes a summary of number of links in each level. We present an example to calculate the total number of links for a network. Assume there are twenty nodes and up to five nodes are allowed in one piconet ($S_{max} = 4$). First round, $\lceil \frac{20}{5} \rceil = 4$ level-1 nodes are built and totally $4 \times 4 = 16$ links are setup. In

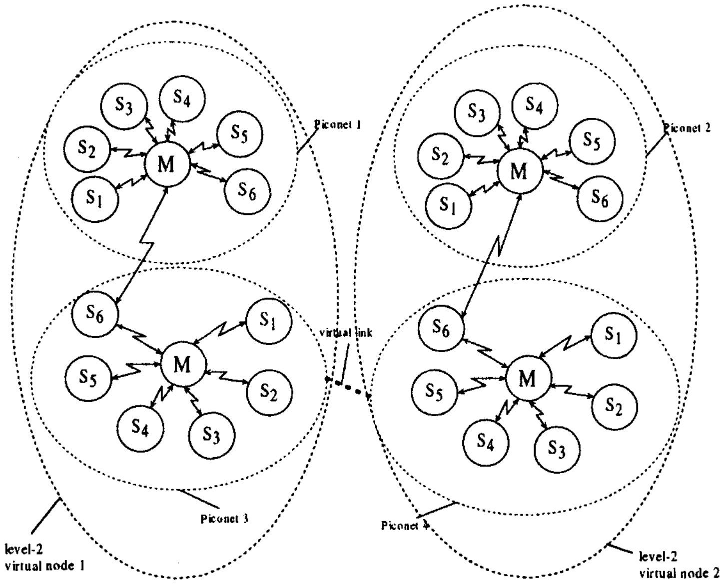


Figure 4.4: A Special Case, $S_{max} = 6$

level	number of lower layer links between 2 nodes
$l(l > 1)$	1 if $S_{max} = 6$, otherwise l
1	1
0	1

Table 4.1: Summary of Number of Links in Each Level

the second round, every two level-1 nodes are grouped together and one level-1 link is setup for each pair. One ACL link is setup by the connect subroutine for each level-1 link. Now, two more ACL links are brought up. Then, at the beginning of the third round, there are two level-2 nodes. Thus, two level-2 links need to be setup. The algorithm looks into these two nodes and find two level-1 nodes in each of them. Then two level-1 links have to be setup among these four level-1 nodes. At last, one ACL link is built for each of these two level-1 links, which leads to two more ACL links in the network. Totally, $16 + 2 + 2 = 20$ ACL links are setup to make a fully connected network.

The following is the pseudo-code for the connect subroutine.

```
Connect($l$, $i$, $j$)

    if $l = 0$
        begin
        set $C_{i,j} = 1$
        end
    elseif $l = 1$
        begin
        set $L^0_{ab}$ = the heaviest possible ACL link
            between $N_i$ and $N_j$
        set $C_{a,b} = 1$
        end
    else
        begin
        if $S_{max} = 6$
            set $N_l(l) = 1$
        else
            set $N_l(l) = l$
        set $L^{l-1}_{a_ib_i}$ = the set of heaviest $N_l(l)$ possible links in level $(l-1)$
            between $N_i$ and $N_j$
        for each link in this set, call subroutine $Connect(l-1, a_i, b_i)$
        end
```

4.4 Traffic Unaware Algorithm

In order to evaluate the proposed NFTD mechanism stated above, an algorithm that is unaware of loaded traffic is explained in this section. It is called Traffic Unaware Algorithm (TUA herein) in this thesis.

To form a scatternet, the TUA first randomly groups nodes into separate piconets according to a designated piconet size S_{pico}. The master of each piconet is randomly chosen as well. Then, these piconets are regarded as virtual nodes. Each of them are randomly connected to another one, and every pair of these connected virtual nodes is regarded as a new virtual node. This virtual node pairing procedure is executed recursively until all nodes are finally connected.

The scatternet made by the TUA satisfies the Bluetooth piconet size constraint while guaranteeing full network connectivity. These are the main concerns and objectives of the algorithms presented in many previously published papers [2] [24] [3] [26]. In addition, TUA is simple and practical. Therefore, the TUA can be used as the typical Bluetooth network formation algorithm that is unaware of the traffic distribution loaded on the network.

However, since TUA is totally blind to loaded traffic on the scatternet, nodes in the same traffic hotspot are likely to be put in different piconets. Large flows between these nodes burden intermediate nodes along the path and impair the network capacity. By comparing with TUA, performance of the NFTD mechanism is analyzed in detail in the next chapter.

Chapter 5

Simulation Results and Discussions

A number of computer simulations are done to evaluate the performance of the NFTD mechanism proposed in this thesis. All simulations are programmed using C language and the results are discussed in detail in this chapter. The rest of the chapter is organized as follows. First, Section 5.1 describes the assumptions made for the simulations and the metrics used for performance analysis. Some terms and simulation parameters are also given in the section. Section 5.2 and Section 5.3 investigate the performance of the NFTD mechanism with two different types of traffic distributions respectively. One of them uses the server-type hotspots and the other uses the uniform-type hotspots, which are described in Section 5.1. For each type of traffic distribution, the impact of varying chosen network parameters is analyzed in the following subsections. These parameters include the number of nodes, a traffic parameter, hotspot and piconet sizes. Use of links between access points is also studied under different types of traffic distributions in Section 5.2.3 and Section 5.3.4 respectively. At last, Section 5.4 concludes this chapter.

5.1 Simulation Assumptions and Performance Metrics

The objective of the NFTD mechanism is to increase the capacity of the resultant network. A metric called *congestion load* is used to evaluate Bluetooth network capacity. We define the congestion load as the load on the bottleneck node of the network, defined as the node with the heaviest load among all the nodes in the network. Equation 5.1 shows how the load of a particular node i, which is represented by Ld_i, is calculated.

$$Ld_i = \sum_{j \neq i} T_{i,j} + \sum_{k \neq i} T_{k,i} + \sum_{m,n \neq i} 2 \times TT_{m,n} \tag{5.1}$$

In Equation 5.1, T is the traffic matrix. The first two terms are flows from and to node i. In the third term, $TT_{m,n}$ represents any transit traffic flow via node i from nodes m to node n. The flows are multiplied by 2 because bandwidth of node i has to be occupied at the ingress and egress flows.

Assume that each entry in the T is relatively small so that no node in the network is overburdened. However, if the traffic flows are scaled up, the bottleneck node is the first one to reach 1Mbps link capacity, so the *congestion load* reflects the capacity of the whole network. In other words, if the same traffic pattern is loaded on two different Bluetooth networks, the one with lower *congestion load* should has higher capacity. Thus, the *congestion load* is an important metric in our study.

Another metric is the Average Path Length (APL) adopted from [24]. APL is defined by Equation 5.2.

$$APL = \frac{\sum L_i}{N_p} \tag{5.2}$$

In Equation 5.2, L_i stands for the hop count of the path for a particular node pair i, N_p denotes the total number of node pairs in the network. The network formation algorithm guarantees full network connectivity, and, thus there is always a valid path between any pair of nodes. The path for a certain node pair is chosen by the shortest path algorithm, which is used as the routing algorithm used in the simulations.

In the simulations, a number of nodes are randomly distributed in a $7m \times 7m$ area, which guarantees all the node are within 10m range of each other (we assume that two nodes can communicate with each other directly if the distance between them is less than or equal to 10m.) We categorize traffic flows between nodes into two categories based on a certain threshold, denoted as T_{thrd}. A given traffic flow is called a *hotspot flow* if it is heavier than T_{thrd}. Otherwise, it is called a *non-hotspot flow*.

The value of T_{thrd} has great impact on the network performance. Even though some observations have been made in Section 5.2.2 and Section 5.3.2, how to choose a proper value for T_{thrd} is out of the scope of this thesis. To maintain simulation simplicity, we assume the value of T_{thrd} is 10 kbps. Without losing generality, we only generate two types of traffic flows. One is 15Kbps and belongs to the hotspot flows. The other is smaller than 1Kbps and belongs to the non-hotspot flows. These traffic flows are not uniformly distributed and, thus, form an un-balanced traffic distribution.

As we discussed in previous chapters, un-balanced traffic distributions generate hotspots inside a Bluetooth network. A hotspot is defined as a set of nodes that have relatively high traffic flows among them. Its size is defined as the number of nodes in this node set. Two types of hotspots are discussed in this chapter. The first one is called server-type hotspots and the other is called uniform-type hotspots.

A server-type hotspot associated with a server node i is denoted by $ST(i)$. The traffic flow between node j and node k in this node set, denoted as $T_{j,k}$, follows Equation 5.3. In other words, all traffic flows between node i and other nodes in $ST(i)$ are hotspot flows while others are non-hotspot flows.

$$T_{j,k} \begin{cases} = 15Kbps, & j = i \; or \; k = i \\ < 1Kbps, & j \neq i, k \neq i \end{cases} \tag{5.3}$$

A uniform-type hotspot including node i is denoted by $UT(i)$. The traffic flow between any nodes j and k in $UT(i)$ follows Equation 5.4. In other words, all traffic flows between any pair of nodes in this hotspot are hotspot flows.

$$T_{j,k} = 15Kbps \qquad j,k \in UT(i) \tag{5.4}$$

In practice, the size of $ST(i)$ or $UT(i)$ is different for different i. To simplify our simulation, we assume that all hotspots have the same size, represented by S_{hot}. We also assume that there are totally $\lfloor \frac{N}{S_{hot}} \rfloor$ hotspots among N nodes. We investigate the performance of the NFTD mechanism for server type and uniform type of traffic distributions in Section 5.2 and Section 5.3 respectively.

The maximum size of the piconets allowed in the network is denoted by S_{pico}. Hence, $S_{pico} = S_{max} + 1$, where S_{max} is defined as the maximum number of slaves allowed in a piconet. Smaller S_{pico} leads to more piconets if the number of nodes N is fixed. When the number of piconets increases, serious mutual interferences might arise and impair the network capacity. However, this kind of radio interferences are not modelled in the thesis. This suggests future work.

For TUA simulation, every plot in the curves is an average of 500 runs. The simulation runs explained in the following sections are all under these assumptions and use the prescribed notations. Their results are evaluated by *congestion load* and APL.

5.2 Server-Type Hotspots

This section summarizes simulation results with only server-type hotspots. The following subsections are organized as follows. Section 5.2.1 investigates performance of the NFTD mechanism when the number of nodes varies. Section 5.2.2 studies the impact of difference between the values of non-hotspot flows and hotspot flows. Section 5.2.3 examines benefit of wired links between the access points. Impact of hotspot and piconet sizes is also observed in Section 5.2.5 and Section 5.2.4 respectively.

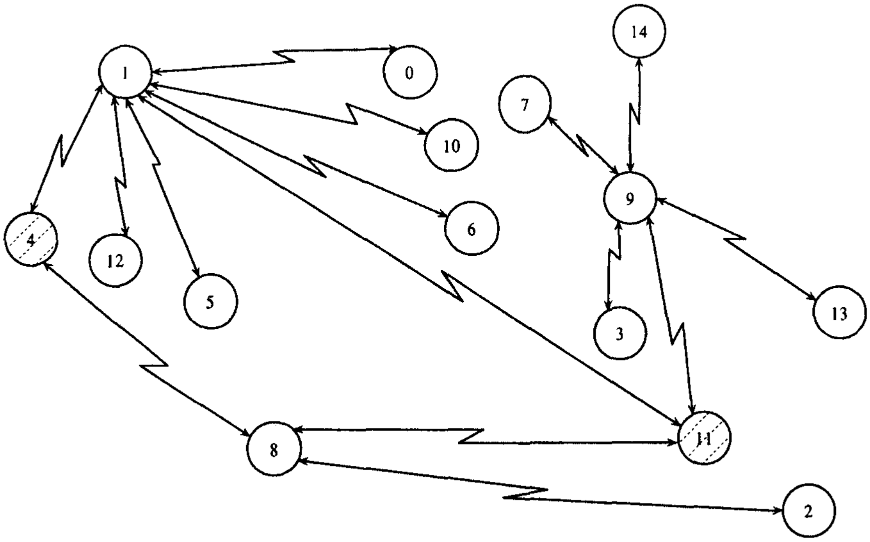
5.2.1 Impact of Number of Nodes

In this subsection, simulation results are presented to show the performance of NFTD when the number of the mobile nodes in the area varies. Congestion loads and APLs are plotted in the figures to evaluate the performance of NFTD and TUA.

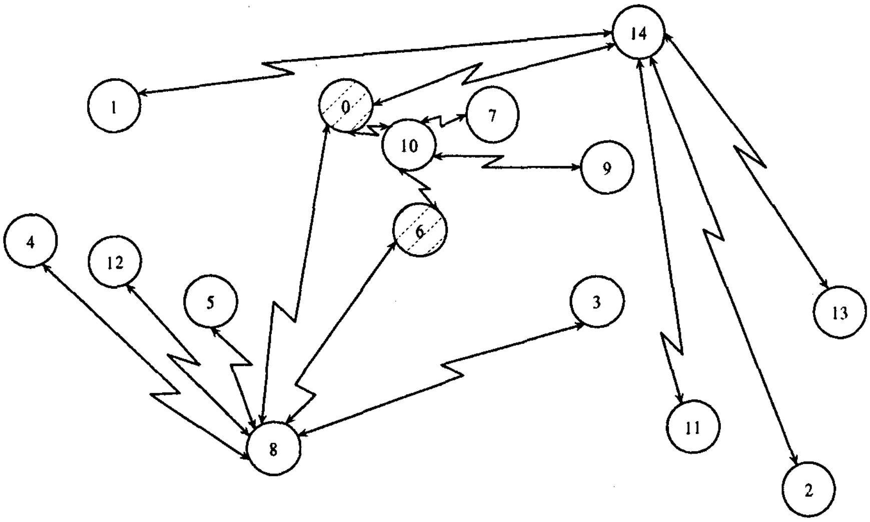
Since NFTD mechanism and TUA form different topologies, we use an example to demonstrate the difference. Figure 5.1 illustrates two topologies by NFTD and TUA respectively. There are fifteen mobile nodes where node 1 and node 9 are acting as servers providing information to exactly five client nodes each. In this example, the five clients of node 1 are nodes 0, 5, 6, 10, and 12 whereas the five clients of node 9 are nodes 3, 7, 11, 13, and 14. Figure 5.1(a) shows the topology of the scatternet formed by the NFTD mechanism. The lines in the topology figures are the ACL links between masters and their slaves. Slaves with more than one master are bridging slaves. They switch among the piconets they are involved in for packet forwarding. By using the NFTD mechanism, it can be seen that two end nodes of a heavy traffic flow are mostly to be included in the same piconet, reducing the probability to route this flow via inter-piconet links. Figure 5.1(b) shows a scatternet that TUA forms. Most of the hotspot flows are routed through a multi-hop path. For example, the hotspot flow $T_{9,13}$ has to go through node 10, node 0, and node 14, making the path length of four hops. Consequently, node 0 happens to be the bottleneck node in the TUA's scatternet, which has 266.7Kbps of congestion load. On the other hand, node 1 happens to be the bottleneck node in the NFTD mechanism's scatternet, with only 88.1Kbps of congestion load.

Figure 5.2 shows congestion loads as the number of nodes ranges from 5 to 60 using the NFTD mechanism and TUA. In the simulations, we assume that non-hotspot flow, t_{low}, is 0.1Kbps and the size of hotspots, S_{hot}, is four.

First, it can be seen that the scatternet formed by the NFTD mechanism has much lower congestion load than that of the TUA, matching our expectation of capacity improvement with the NFTD mechanism. It is reasonable because of high probability of clustering those nodes in one hotspot in the same piconet. It also can be seen that all curves increase as the number of mobiles in the area increases. This is mainly because more mobiles lead to more hotspots under the assumptions of our simulation,



(a) NFTD



(b) TUA

Figure 5.1: Topologies Made by NFTD mechanism and TUA, $N = 15$, $S_{hot} = 6$, $S_{pico} = 6$, Server-Type Hotspots

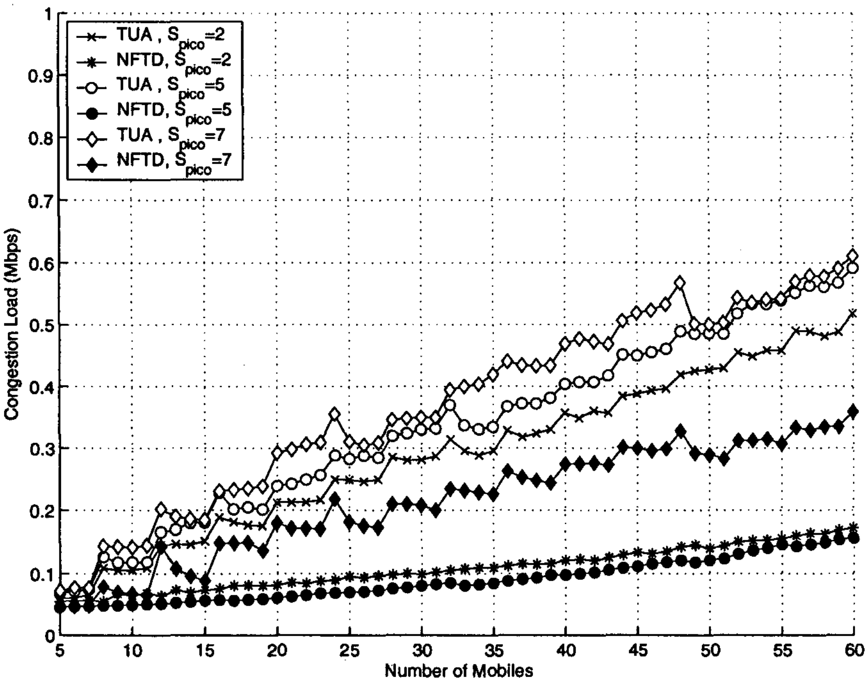


Figure 5.2: Congestion Load vs. Number of Nodes, t_{high} = 15Kbps, t_{low} = 0.1Kbps, S_{hot} = 4, Server-Type Hotspots

resulting in increased total traffic load on the scatternet. From this point of view, using the NFTD mechanism, which is based on the traffic distribution, the resultant scatternets can hold more mobiles than TUA.

$$APL_h = \frac{\sum HL_i}{N_h} \tag{5.5}$$

Moreover, as mobiles in the same hotspot are grouping together, packet delay in the hotspot can be restricted to a relatively small value. Note that since there is no specification defined on scheduling in Bluetooth, we are unable to model the per hop propagation delay. We assume that there is a constant per hop propagation delay. Therefore, the end-to-end delay for a flow is proportional to the hop count along its path. In the simulations, We use the average hop count to map the end-to-end delay, as defined by Equation 5.5. In this equation, HL_i stands for the hop count of the path for a hotspot flow i. N_h represents the total number of the hotspot flows. Figure 5.3 plots the curve of APL_h with different number of mobiles. As we can see, node pairs with large traffic are only one hop away using the NFTD mechanism. On the contrary, hotspot flows go through a much longer path in the scatternets formed by TUA.

5.2.2 Impact of Traffic Parameter

In this section, simulations are done to examine the capacity of NFTD mechanism when the traffic pattern changes. To be more specific, we want to see the impact of the traffic loads on the system performance. We fix the transmission rate for the hotspot flow to 15Kbps and vary the transmission rate for non-hospot traffic flows. We define the ratio of the low transmission rate versus the heavy transmission rate as α, which is shown in Equation 5.6. Note that if α is close to 1, there is no major difference between hotspot flows and non-hotspot flows. If α is close to 0, the hotspot flows becomes the dominating traffic flows in the network.

$$\alpha = \frac{t_{low}}{t_{high}} = \frac{t_{low}}{15Kbps} \leq 1 \tag{5.6}$$

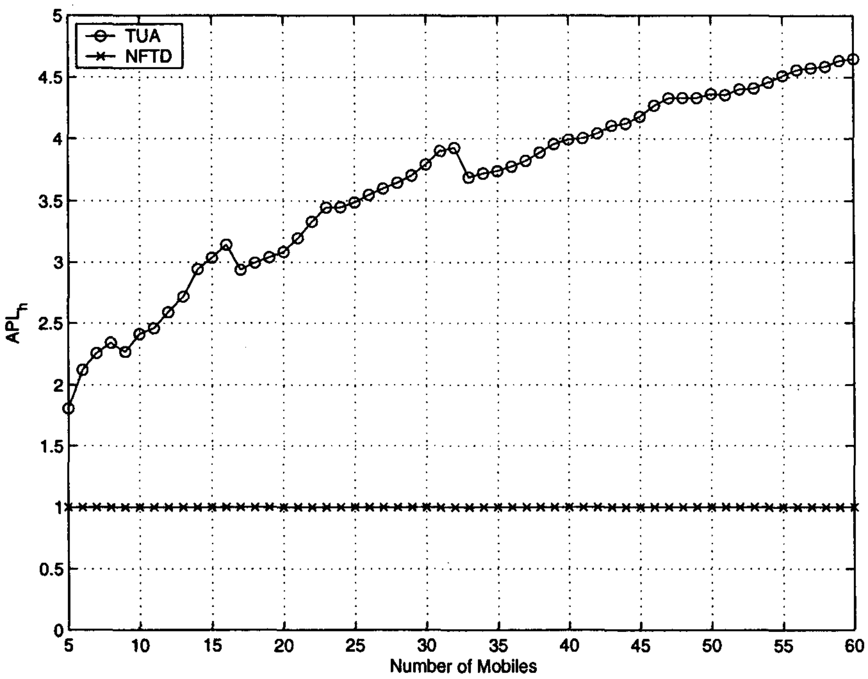


Figure 5.3: APL_h vs. Number of Nodes, $S_{hot} = 4$, $S_{pico} = 4$, Server-Type Hotspots

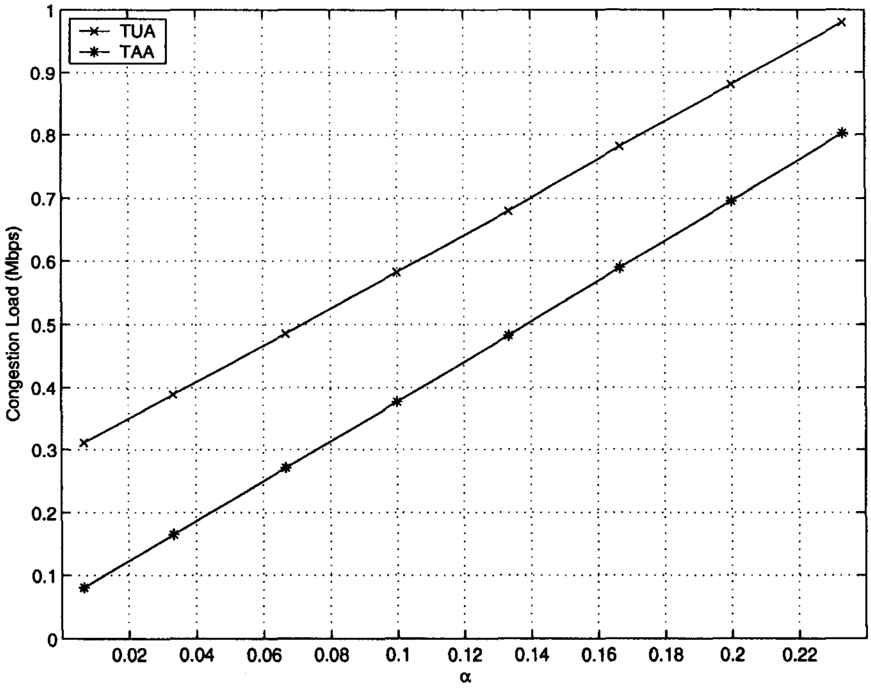


Figure 5.4: Congestion Load vs. α, $t_{high} = 15Kbps$, $S_{hot} = 5$, $S_{pico} = 5$, $N = 20$, Server-Type Hotspots

In this simulation, the transmission rate of the non-hotspot flows is increased gradually from a very low value by slowly increasing α. Again, notice that the traffic running in the scatternet is increasing as the non-hotspot flows are going up. Figure 5.4 shows that the NFTD mechanism has much higher capacity since its congestion load is much lower compared to the TUA. The TUA curve almost reaches the Bluetooth bandwidth limit of 1Mbps when α is 0.233, while the NFTD mechanism curve does not reach that point, meaning that scatternets formed by NFTD mechanism can hold more traffic.

We simulate across a large range of α values. As we can see in Figure 5.4, the gap between the NFTD and the TUA curves slightly decreases as the value of α increases. This is because with a larger α, the non-hotspot flows become heavier and the difference between hotspot flow and non-hotspot flow fades away. When there is no difference between these two types of traffic, the resulting formation of NFTD mechanism should be very close to that of TUA.

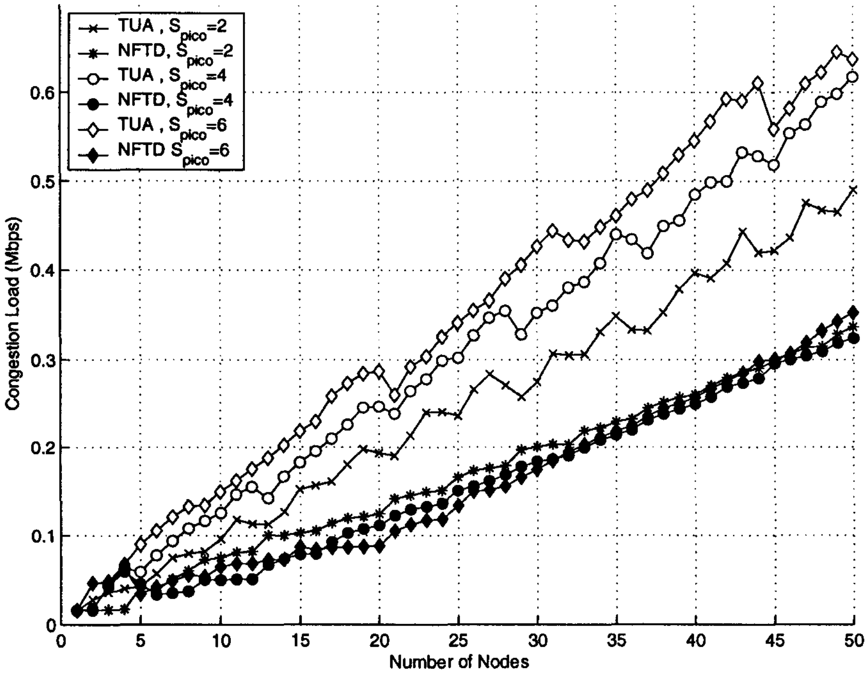


Figure 5.5: Congestion Load vs. Number of Nodes, Pure Access Scenario, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$

Another simulation experiment is to examine an access scenario, where all mobiles in the area only desire to exchange data with outside nodes through 4 access points. This is a special case of server-type hotspots. In this case, servers in the hotspots are the access points and every mobile device is the client of one of the access points. Figure 5.5 compares the performance of the NFTD mechanism and the TUA when the number of mobiles and piconet size vary. NFTD mechanism consistently outperforms TUA. For example, when there are 30 mobiles in the area and piconet size are limited to 5, congestion load of the network setup by NFTD mechanism is only 42.8% of that by TUA, suggesting that the proposed NFTD mechanism is able to support higher network load compared to TUA. There is no additional cost imposed by NFTD. Hence it is a promising mechanism for Bluetooth network formation.

5.2.3 Impact of Infrastructure Links Between Access Points

In this subsection, simulations are done to investigate whether using high-speed links between access points help increase the network capacity. Presumably, a flow can choose a path going through the infrastructure links instead of through multi-hops of mobiles in order to lighten the burden of those intermediate mobiles. In the shortest path routing algorithm used in the simulations, path weight of infrastructure links are set to 0 while that of Bluetooth links are set to 1, so that infrastructure links get higher priority.

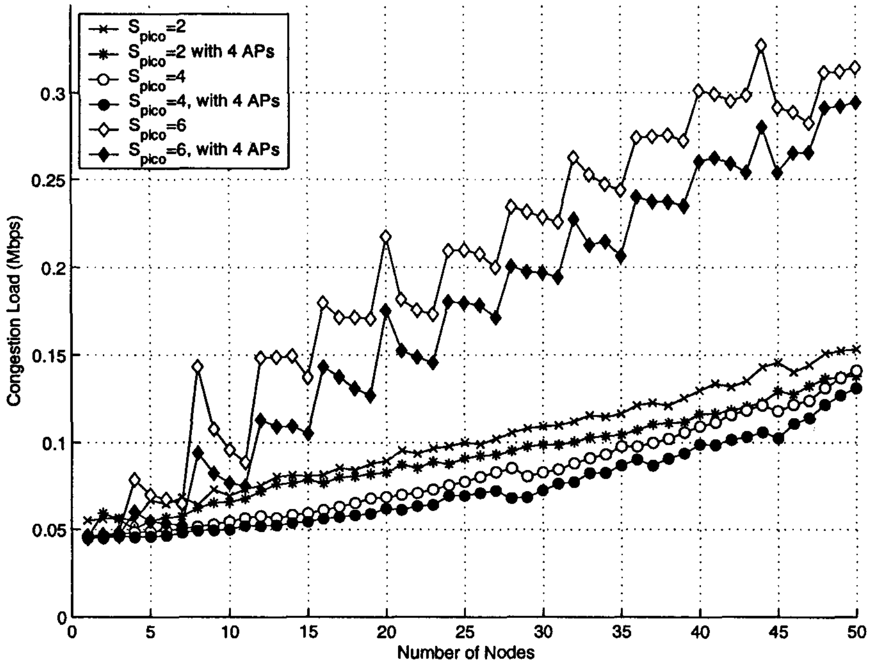


Figure 5.6: Congestion Load vs. Number of Nodes, 4 Access Points, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Server-Type Hotspots

Figure 5.6 shows the congestion load of network formed using the NFTD mechanism, when server-type hotspots exist in traffic. It can be seen that utilizing high-speed links between access points leads to capacity improvement, regardless of any changes in the number of mobiles and the size of piconets.

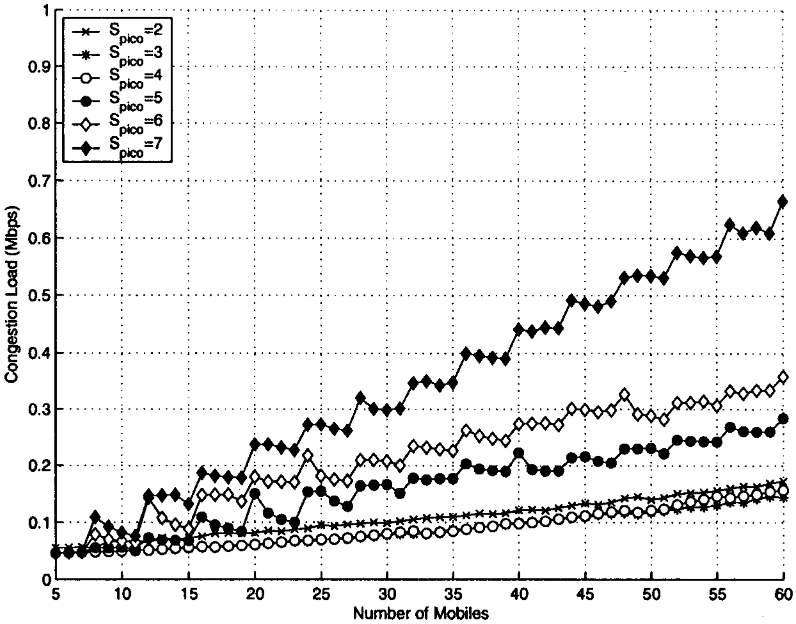
5.2.4 Impact of Piconet Size

As we can see in Figure 5.2, the allowed maximum number of slaves for a master impacts scatternet capacity as well. The number of slaves in a piconet is restricted to be smaller than or equal to seven according to the Bluetooth standard. However, two separated piconets with seven slaves each, cannot be able to inter-connect. This is because a piconet with seven slaves cannot accept more slave, and in this case, at least one slave in one piconet has to join the other piconet to act as a bridge if they are to inter-connect.

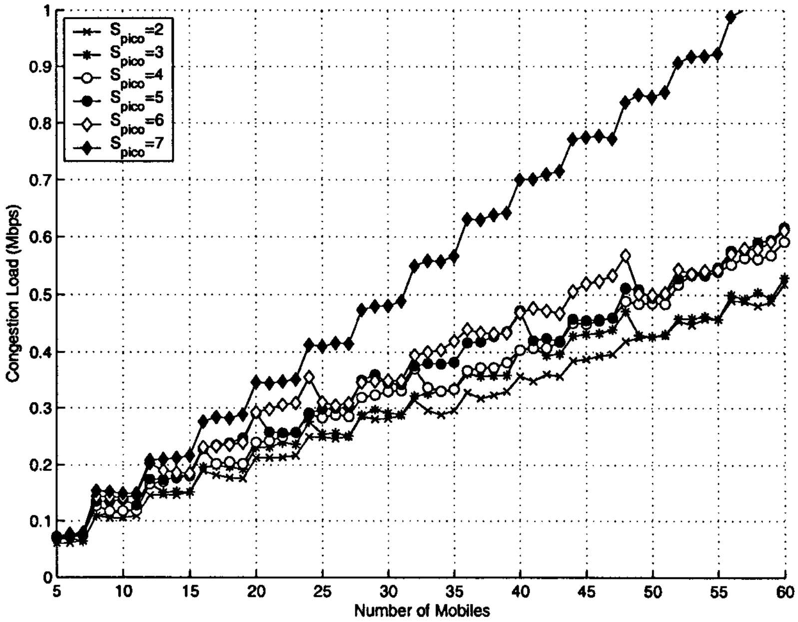
Figure 5.7 shows the scatternet capacity. For scatternets built by TUA, they tend to have higher capacity as the size of piconet decreases. This implies that more hotspot flows can take only one hop route to reach their destination and in turn, lower aggregated link throughput can be achieved.

The similar trend can also be seen in the results of NFTD. Generally, a smaller S_{pico} leads to a lower congestion load. However, when S_{pico} equals to 4 in Figure 5.7(a), the networks formed by NFTD have the lowest congestion loads. The reason is that we set S_{hot} to be 4, which is equal to the hotspot size. According to NFTD, this setting guarantees that node in the same hotspot are grouped into one piconet as possible. When traffic loads using Equation 5.1, the third term is very small for every node i. Thus, traffic load on every node is relatively low, leading to relatively low congestion load.

We also show the APL_h curves in Figure 5.8 with different S_{pico} values. In the case that nodes in one traffic hotspot are grouped into the same piconet with the sever be the master, heavy traffic flows are only running between the master and its slaves. According to Equation 5.5, HL_i is always equal to 1. Therefore, APL_h is 1 when the optimal piconet size is used, while other piconet sizes yield definitely longer



(a) NFTD



(b) TUA

Figure 5.7: Congestion Load vs. Number of Nodes, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Server-Type Hotspots

paths, matching our expectation for NFTD in the previous paragraph.

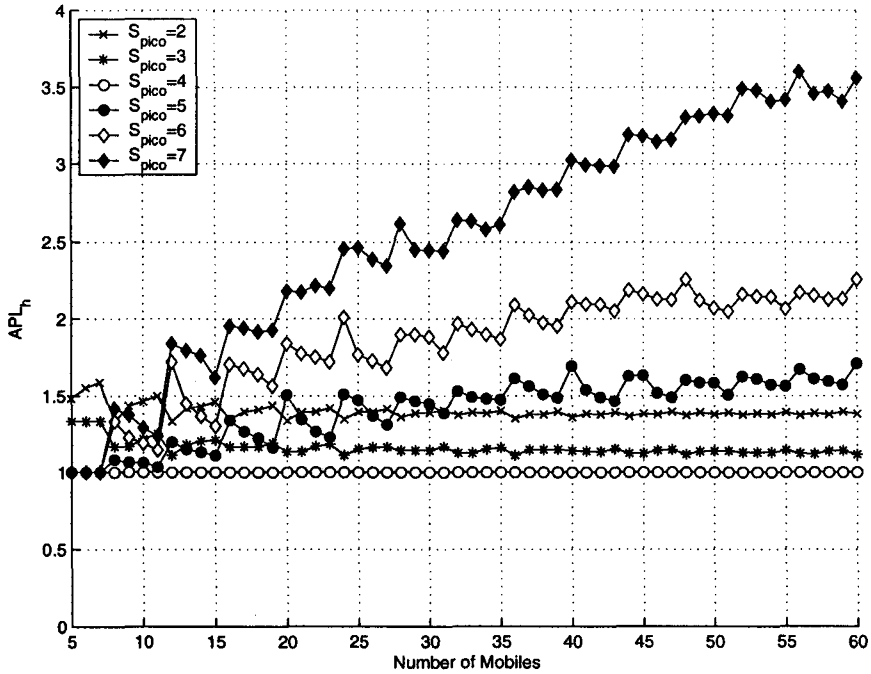


Figure 5.8: APL_h vs. Number of Nodes, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Server-Type Hotspots, NFTD

5.2.5 Impact of Hotspot Size

In this experiment, a constant piconet size of 5 is used and 20 mobiles are placed in the area. Network capacity is examined when traffic with different hotspot sizes are applied to the network. From Figure 5.9, it can been seen that the hotspot size affects the capacity of scatternets in a similar way as α does. The scatternets formed by the NFTD mechanism again outperforms those formed by TUA. Unlike the curves with different α values, the curves in Figure 5.9 does not show a clear relationship between the hotspot size and the performance, suggesting a potential work on this topic.

Now, all simulations for the server-type hotspot are discussed. The next section focuses on the other type of hotspot namely the uniform-type.

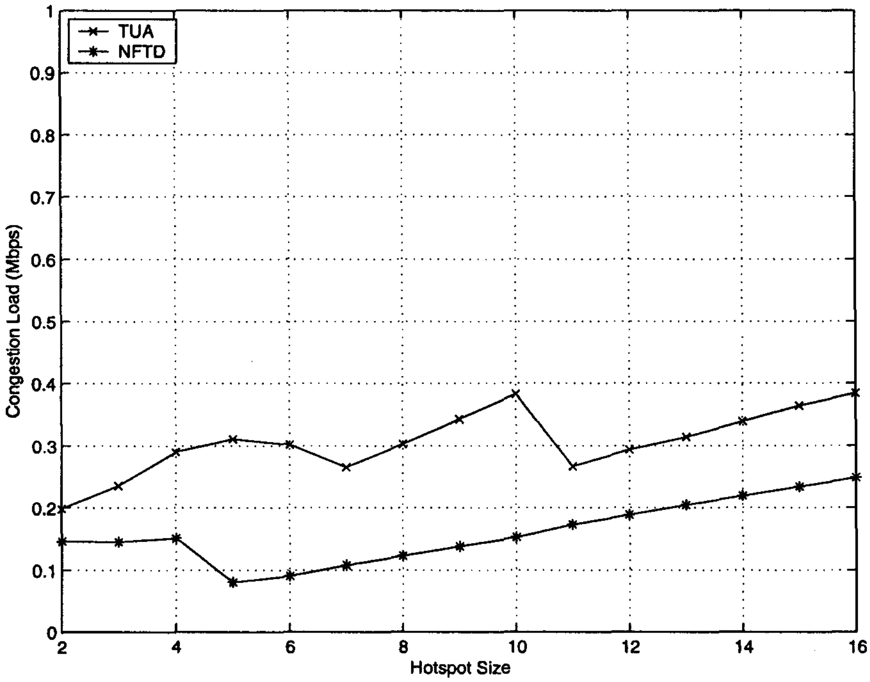


Figure 5.9: Congestion Load vs. S_{hot}, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $N = 20$, $S_{pico} = 6$, Server-Type Hotspots

5.3 Uniform-Type Hotspots

This section presents simulations with only uniform-type hotspots in the loaded traffic in the resulting networks. The following subsections are organized as follows. Section 5.3.1 investigates performance of the NFTD mechanism when the number of nodes varies. Section 5.3.2 studies the impact of the difference between the values of non-hotspot flows and hotspot flows. Section 5.3.4 examines benefit of wired links between the access points. Impact of hotspot size and piconet size are also observed in Section 5.3.5 and Section 5.3.3 respectively.

5.3.1 Impact of Number of Nodes

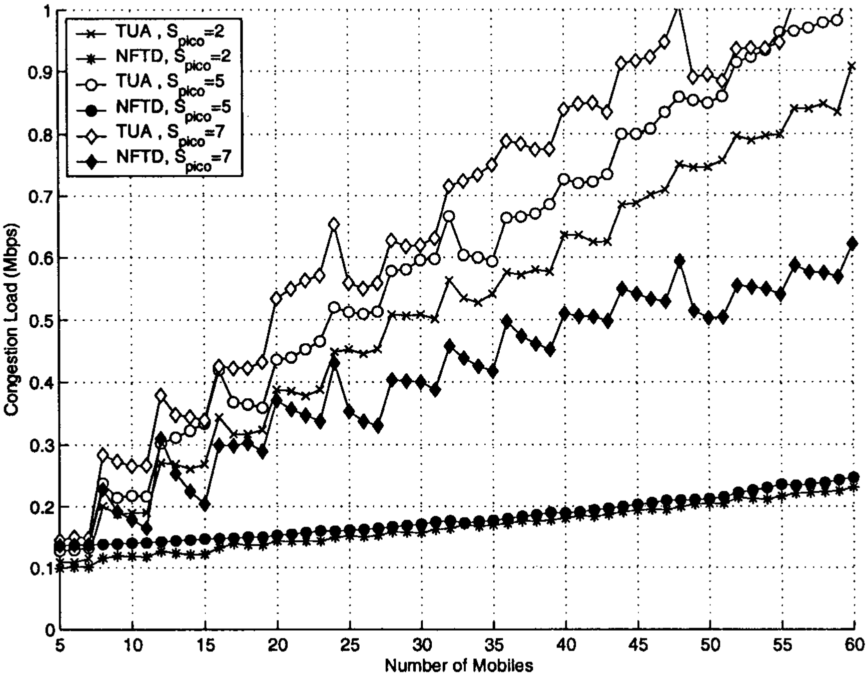


Figure 5.10: Congestion Load vs. Number of Nodes, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Uniform-Type Hotspots

We investigate the performance of the NFTD mechanism and the TUA with different number of nodes in this section. The NFTD mechanism again outperforms TUA in this case. As shown in Figure 5.10, no matter how large the number of nodes in the area is or how the maximum piconet size varies, the scatternets formed by NFTD mechanism always have lower congestion load compared to the scatternets formed by TUA, showing that a better capacity is achieved by using traffic adaptive algorithm.

However, it should be noticed that the performance gain is not as significant as we observe in Figure 5.2. This is reasonable for traffic with uniform type hotspots. As nodes in the same hotspot are clustered together in the same piconet, most of the heavy traffic flows within the hotspot have to go through the master of the piconet and intensely increase the load on the master. This is not a problem in Section 5.2.1, because the nature of server-type hotspots leads to very low traffic flow between slaves. However, as we can see in Figure 5.10, clustering nodes in the same hotspots still significantly reduces inter-piconet traffic, leading to higher performance for NFTD mechanism compared to TUA.

The improvement can also be illustrated by APL_h, as shown in Figure 5.11. In the case that nodes in one hotspot are grouped in the same piconet by NFTD mechanism as possible, the APL_h can be made very close to or equal to 1. For scatternets made by TUA, the APL_h is much longer and increases significantly when number of mobiles increases. Therefore, the proposed NFTD mechanism can significantly improve the network performance.

5.3.2 Impact of Traffic Parameter

Similar to Section 5.2.2, we present the curves of NFTD and TUA for different traffic patterns in Figure 5.12. The NFTD mechanism shows higher capacity than TUA. In this figure, the difference between these two curves is more significant when compared to the corresponding graph in the server type hotspot case in Section 5.2.2 (Figure 5.4). Basically, this suggests that increasing the value of each high traffic flow has a stronger impact on the total traffic flow in this type of hotspot.

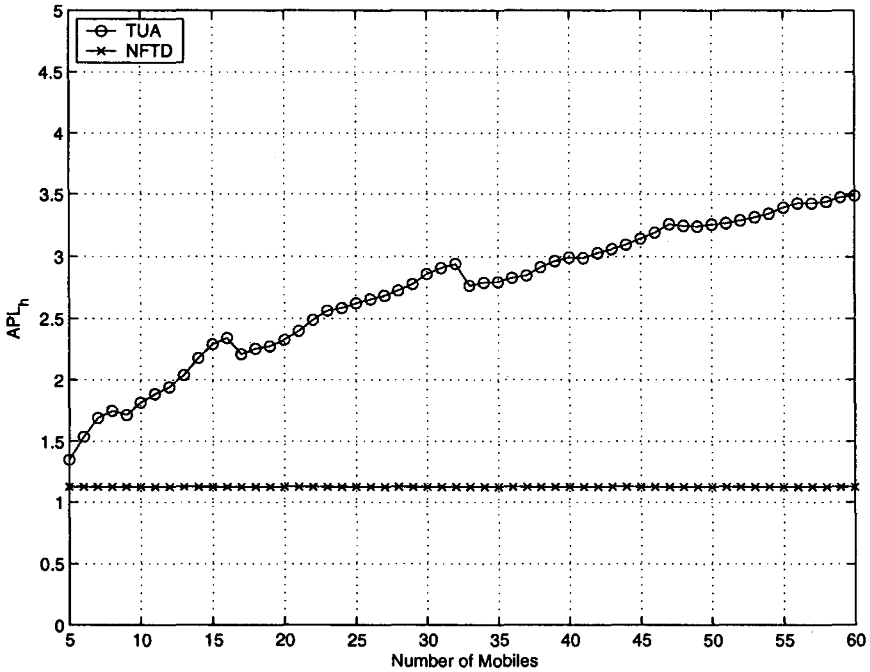


Figure 5.11: APL_h vs. Number of Nodes, $S_{hot} = 4$, $S_{pico} = 4$, Uniform-Type Hotspots

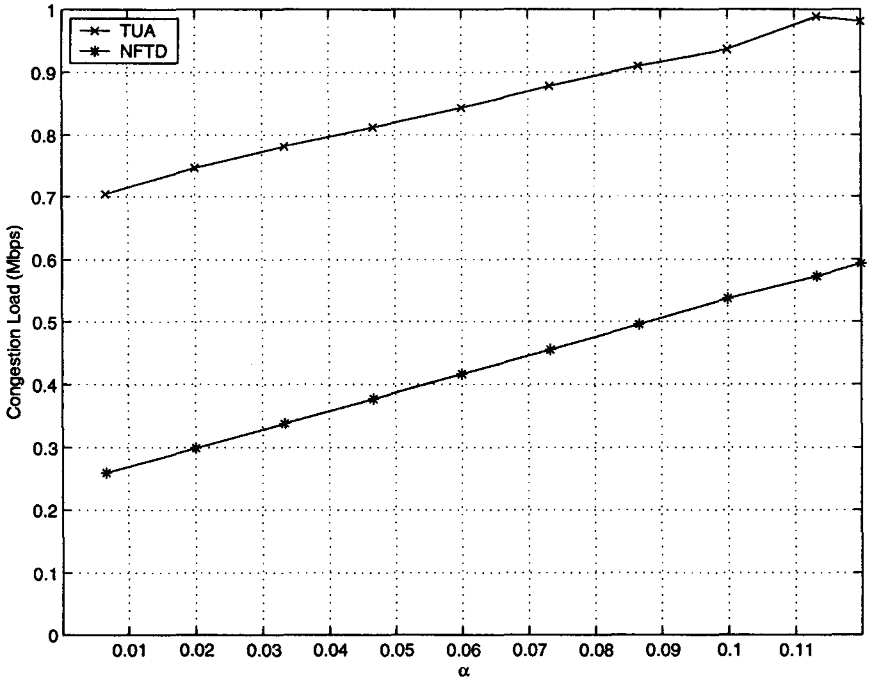


Figure 5.12: Congestion Load vs. α, $t_{high} = 15Kbps$, $S_{hot} = 5$, $S_{pico} = 5$, $N = 20$, Uniform-Type Hotspots

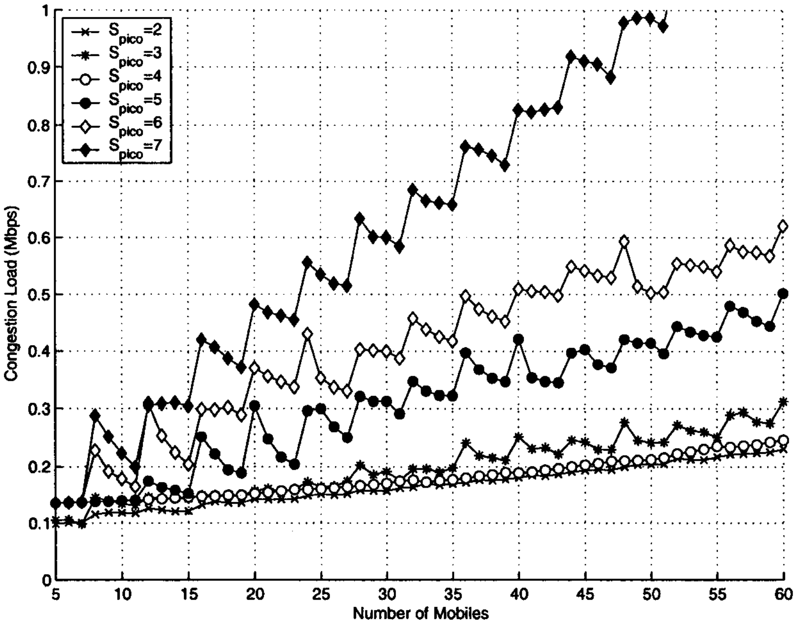
5.3.3 Impact of Piconet Size

Figure 5.13 shows how the piconet size affects the NFTD algorithm and the TUA. From Figure 5.13(b), it can be seen that decreasing the piconet size increases the capacity of the scatternet formed by TUA. The reason is the same as in Section 5.2.4. The curves for NFTD is shown in Figure 5.13(a). Again, an optimal piconet size exists when the NFTD mechanism is used. The optimal piconet size is four according to the S_{hot} value used. Therefore, the curve with $S_{pico} = 4$ provides nearly the best performance as compared to other S_{pico} curves. Interestingly, we can see that the curve with $S_{pico} = 2$ also provides the optimal performance compared to other values of S_{hot}. This is because the number of links increases when the value of S_{pico} is small. Therefore, hotspot flows can be distributed evenly over more links, leading to low congestion load. Moreover, the more piconets are in the small area, the more interferences will exist as mentioned in Section 5.1. However, we have not modelled the physical transmission and interference in the simulations and this can be part of future work.

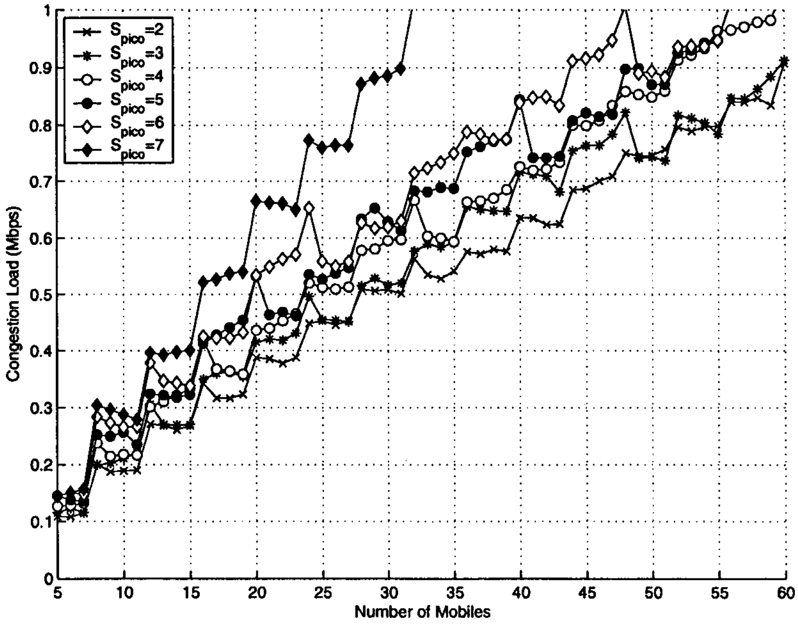
Note that with $S_{pico} = 2$, we are expecting longer APL_h than that of $S_{pico} = 4$. It is because only when $S_{pico} = 4$, a hotspot is grouped into one piconet. This leads to $HL_i = 1$ for each flow i in Equation 5.5, which is the smallest possible value. As we can see in Figure 5.14, the APL_h curve for $S_{pico} = 4$ outperforms others, including $S_{pico} = 2$, matching our expectation and suggesting one to use the optimal S_{pico} value instead of 2. Also, a smaller piconet size leads to more piconets, and the probability of two links using the same frequency in one hop increases. That leads to more radio interference, and capacity will definitely suffer. Hence, letting the piconet size be the same value as the hotspot size might be a better option.

5.3.4 Impact of Infrastructure Links Between Access Points

Figure 5.15 shows the curves of congestion load when loaded traffic has uniform-type hotspots and there are four access points in the area. Similar to Figure 5.6, presented in Section 5.2.3, we can arrive at the same conclusion. Utilizing high-speed links between access points leads to noticeable capacity improvement.



(a) NFTD



(b) TUA

Figure 5.13: Congestion Load vs. Number of Nodes, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Uniform-Type Hotspots

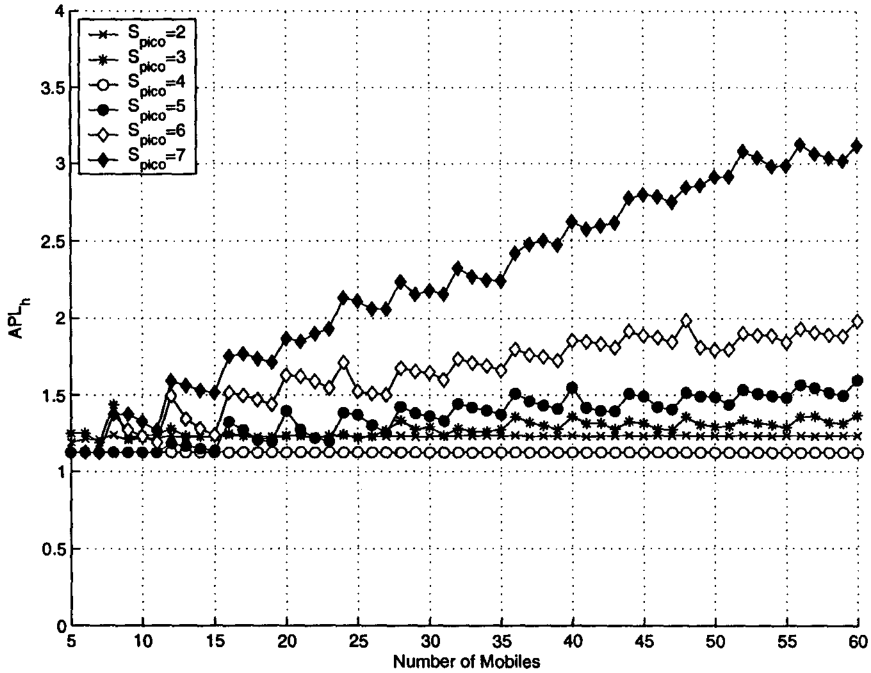


Figure 5.14: APL_h vs. Number of Nodes, $t_{high} = 15Kbps$, $S_{hot} = 4$, Uniform-Type Hotspots, NFTD

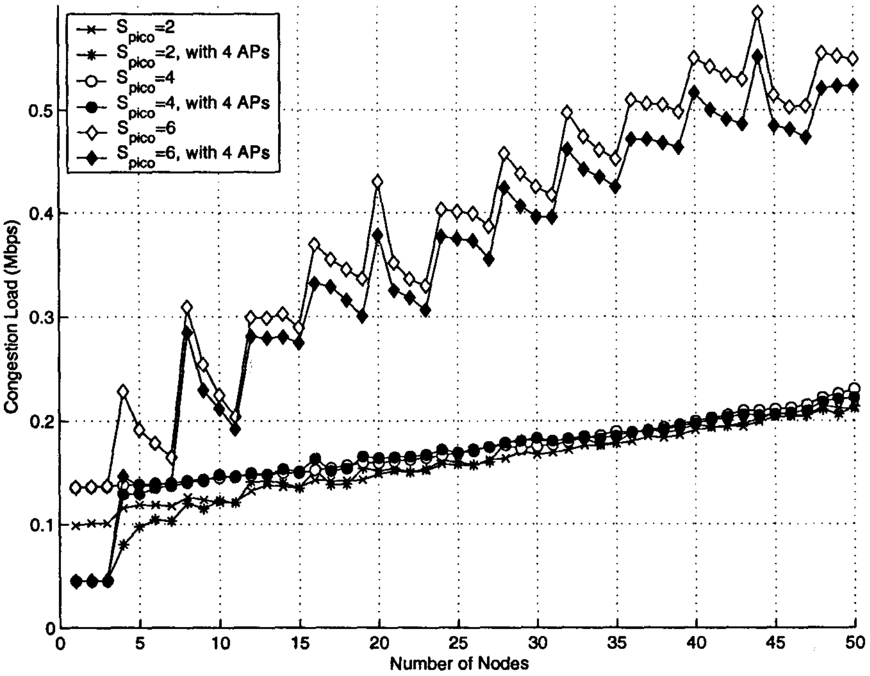


Figure 5.15: Congestion Load vs. Number of Nodes, 4 Access Points, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $S_{hot} = 4$, Uniform-Type Hotspots, NFTD

5.3.5 Impact of Hotspot Size

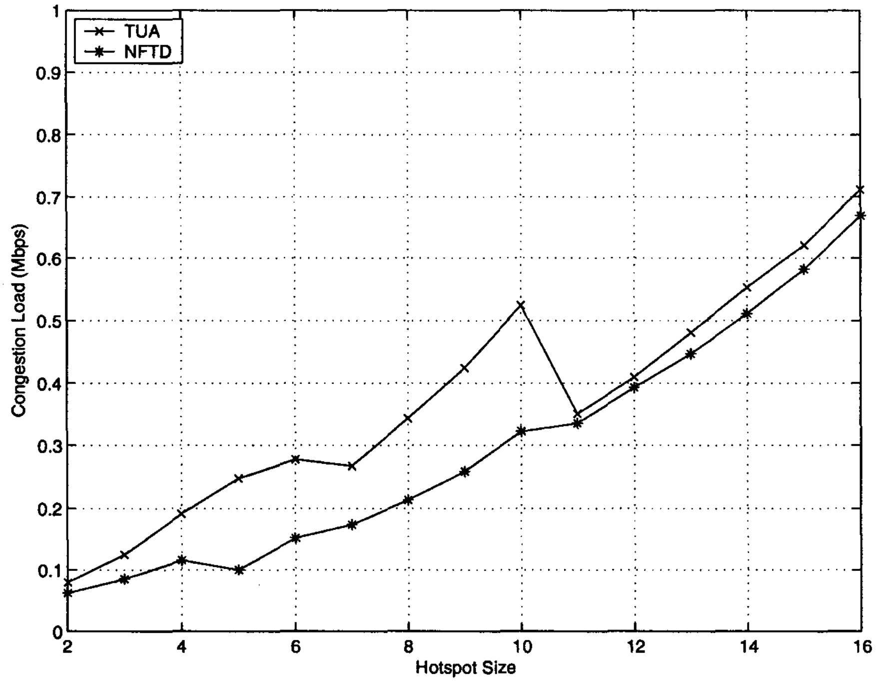


Figure 5.16: Congestion Load vs. S_{hot}, $t_{high} = 15Kbps$, $t_{low} = 0.1Kbps$, $N = 20$, $S_{pico} = 4$, Uniform-Type Hotspots

This section demonstrates the congestion load curves versus the hotspot size. As shown in Figure 5.16, the NFTD curve is always below the TUA curve, showing that the NFTD mechanism performances better than TUA via higher capacity. Notice that the difference of the two curves is smaller when compared to Figure 5.9 in Section 5.2.5. This is because the NFTD mechanism performs a little better for server-type hotspots than for uniform-type hotspots, as analyzed in the second paragraph in 5.3.1.

5.4 Discussion of Results

The NFTD mechanism proposed in this thesis achieves the design goal, which is to increase the network capacity when forming Bluetooth scatternets in an infrastructure network for a given traffic load. It succeeds in using the traffic distribution information to form Bluetooth networks with relatively higher capacity compared to a straightforward network formation algorithm, namely TUA. Two types of traffic distributions are examined. One of them is server-type hospots and discussed in Section 5.2. The other is uniform-type hotspots and discussed in Section 5.3. Simulation results show that the network capacity is significantly improved in both cases with NFTD. Packet delay is expected to be relatively small when compared to the networks formed by TUA.

The impact of some scatternet parameters on the performance of the resulting network are examined in both cases as well. Firstly, there exists an optimal size for piconets, especially in the traffic distribution with server-type hotspots. Its value is the same as the hotspot size in the traffic distribution in simulations. Secondly, the performance of NFTD is always better than that of TUA regardless of the hotspot size. Finally, the NFTD mechanism significantly outperforms the other algorithm when the hotspot flows are much heavier than the non-hotspot flows in terms of transmission rate.

Last but not least, high speed wired links connecting access points can provide help in improving the network capacity and decreasing the packet delay, so long as a proper path routing scheme is used to direct flows over them.

To summarize, the NFTD mechanism proposed in this thesis is practical and efficient. Note that optimizing piconet size can achieve better network capacity. How to optimize the piconet size is a promising future research problem.

Chapter 6

Conclusions

Bluetooth is one of the most promising wireless communication technologies used to replace wires for voice and data exchange. Besides the simple universal connectivity it provides, it also has networking capabilities. The Bluetooth access scenario is a very popular networking scenario since fast wired and wireless networks are very common in indoor environments. Since there is no specification on how Bluetooth networks should be formed, Bluetooth networking is receiving increasing interest from both industry and academic groups. Infrastructure networks can facilitate network formation and operation in two ways. First, they can collect information about all Bluetooth nodes in the area, and exchange this information through the existing fast-speed wired or wireless network. Then, they can decide on an optimized network topology for the Bluetooth nodes so that the network formation algorithm is centralized and maintainable. Secondly, they can provide alternative paths for Bluetooth nodes that are hundreds of times faster than Bluetooth data links. By doing this, the capacity of resulting network can be further increased.

We have explored the Bluetooth network formation mechanism in this thesis. This thesis focuses on the above scenario where dozens of Bluetooth-enabled mobiles are in a small indoor area, and access points are deployed to provide fast local network access. A Bluetooth Network Formation based on Traffic Distribution (NFTD) mechanism is proposed to build up a scatternet that guarantees full connectivity. It is a centralized algorithm that can be implemented by one of the access points.

Compared to other algorithms unaware of loaded traffic, the algorithm forms scatternets with the traffic information and provides the maximum network capacity. Two types of traffic patterns are considered; one is with separate client/server hotspots, where there is one server node providing services to every other nodes in the hotspot. The scatternet created by the NFTD mechanism has more than 50% higher capacity than those created by a typical algorithm that is unaware of traffic, namely TUA. The other type of traffic pattern is with peer-to-peer hotspots called uniform-type hotspots, where every node pair inside a hotspot has a larger amount of traffic than the traffic between two nodes that are not in the same hotspot. The NFTD mechanism is also capable of increasing scatternet capacity, but not as much as that with the server-type of hotspots. This is because Bluetooth piconet has a star-shaped topology, which matches the server type of hotspots.

There is still future work left for the Bluetooth network formation issue in access scenarios. First, work needs to be done to simulate the link establishment process, piconet and scatternet scheduling in the baseband. Overhead in the baseband for control and scheduling also needs to be considered in simulations, and will lead to more accurate results. Different indoor topologies can be explored, where new scenarios can emerge - such as some mobile nodes outside the coverage area of the others. In this case, members of a piconet can not be chosen only according to the traffic distribution as in the proposed NFTD mechanism, because two nodes that have hotspot flow between them might be out of radio range of each other. Also, piconet size optimization under a more complex traffic, such as a hybrid-type hotspot patterns, can be considered an interesting extension of this work.

Bibliography

[1] S. Baatz, M. Frank, C. Kuhl, P. Martini, and C. Scholz. Bluetooth Scatternets: An Enhanced Adaptive Scheduling Scheme. 2002.

[2] S. Basagni and C. Petrioli. Degree-constrained Multihop Scatternet Formation for Bluetooth Networks. In *Global Telecommunications Conference, IEEE*, pages 195–200, November 2002.

[3] S. Basagni and C. Petrioli. Multihop Scatternet Formation for Bluetooth Networks. In *Vehicular Technology Conference, IEEE 55th*, pages 424–428, May 2002.

[4] A. Capone, M. Gerla, and R. Kapoor. Effient Polling Schemes for Bluetooth Picocells. In *ICC*, 2001.

[5] S. I. Dharia and D. P. Agrawal. A Novel Distributed Resource-aware Scalable Scheme for Scatternet Formation. In *The 11th IEEE International Conference on Networks*, pages 659–664, Sept.-Oct. 2003.

[6] D. Famolari. Link Performance of an Embedded Bluetooth Personal Area Network. In *Proc. of the IEEE International Conference on Communications*, June 2001.

[7] John Gardiner and Barry West Editors. *Personal Communication Systems and Technologies.* Artech House, 1995.

[8] S. Grag, M. Kalia, and R. Shorey. MAC Scheduling Policies for Power Optimization in Bluetooth: A Master Driven TDD Wireless System. In *VTC*, 2000.

[9] Bluetooth Special Interest Group. *Core-Specification of the Bluetooth System 1.0b.* Bluetooth Special Interest Group, 1999.

[10] IEEE. http://grouper.ieee.org/groups/802/11/, 2005.

[11] IEEE. http://grouper.ieee.org/groups/802/dots.html, 2005.

[12] IEEE. http://ieee802.org/15/, 2005.

[13] ISO. https://www.iso.org/, 1994.

[14] M. Kalia, D. Bansal, and R. Shorey. Data Scheduling and SAR for Bluetooth MAC. In *IEEE VTC*, pages 716–720, Tokyo, 2000.

[15] M. Kalia, S. Grag, and R. Shorey. Efficient Policies for Increasing Capacity in Bluetooth: An Indoor Pico-Cellular Wireless System. In *IEEE VTC*, 2000.

[16] C. Law and K.Siu. A Bluetooth Scatternet Formation Algorithm. In *Global Telecommunications Conference, IEEE*, November 2001.

[17] Ting-Yu Lin and Yu-Chee Tseng. An Adaptive Sniff Scheduling Scheme for Power Saving in Bluetooth. 9(6):92–103, Dec. 2002.

[18] G. Miklos, A. Racz, Z. Turanyi, A. Vlko, and P. Johansson. Performance Aspects of Bluetooth Scatternet Formation. In *Mobihoc*, 2000.

[19] Brent Miller and C. Bisdikian. *Bluetooth Revealed.* Prentice Hall, NJ, 2001.

[20] T. Rappaport. *Wireless Communications.* Prentice Hall, 1996.

[21] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed Topology Construction of Bluetooth Personal Networks. Infocom, April 2001.

[22] R. Shorey and B. A. Miller. The Bluetooth Technology: Merits and Limitations. In *ICPWC*, 2000.

[23] Bluetooth SIG. https://www.bluetooth.org/, 2005.

[24] Zhifang Wang, R.J. Thomas, and Z. Haas. Bluenet - A New Scatternet Formation Scheme. In *System Sciences, Proceedings of the 35th Annual Hawaii International Conference on*, January 2002.

[25] J. Xue and T.D. Todd. Basestation Collaboration in Bluetooth Voice Networks. In *IEEE Workshop on Wireless Local Networks*, Tampa, Florida, November 2001.

[26] G. V. Záruba, S. Basagni, and I. Chlamtac. Bluetrees-Scatternet Formation to Enable Bluetooth-based Ad Hoc Networks. In *Communications, IEEE International Conference on*, pages 273–277, June 2001.

[27] S. Zurbes, W. Stahl, K. Matheus, and J. Haartsen. Radio network performance of Bluetooth. *IEEE Personal Communications*, 7(1), February 2000.