A Review of Bluetooth Technology Fundamentals and its Application to Scatternets

Tongyuan Wang

A Major Report
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science at
Concordia University
Montreal, Quebec, Canada

November 2002

0-612-77997-1

Abstract

A Review of Bluetooth Technology Fundamentals and its Application to Scatternets

Tongyuan Wang

Bluetooth is a newly proposed technology aimed to facilitate the realization of wireless personal networks or other short range networks using affordable Bluetooth chips with low power demand. The emergency of this technology has been getting very positive responses in both the industrial and the academic circles.

This report is focused on the Bluetooth fundamentals and a review of some selected research articles dealing with the Bluetooth technology. We first present a brief but broad view about the technological evolutions from the birth of the early Internet, to wireless networks and to the emergency of Bluetooth technology. Following is a review of the Bluetooth core specifications and its profile specifications. The summaries of the core specifications give us a general impression and fundamental knowledge about the Bluetooth technology, the related terminology and the relationships among the different protocol layers. The summaries of the profile specifications give us the rules to follow to apply the technology into various applications. We then include a review of some research papers dealing with several aspects of scatternets using the Bluetooth technology. The selected papers give us indications about the potential problems or constraints in the implementation of the Bluetooth technology into applications, and the possible solutions or improvements to them. The last part is a short conclusion, where the significance of studying the Bluetooth technology, the perspectives of the applications of the technology, are addressed.

Acknowledgement

I hereby would like to sincerely thank my supervisor Dr. Jaroslav Opatrny for his dedicated supervision and guidance in this report writing, from topic selection, literature information, to various comments and suggestions on my work. Special thanks should also be to the Computer Science Department of Concordia University, which provides students with a very supportive studying environment and facilities.

Tongyuan Wang

November 2002

Table of Contents

Introduction

The birth of network

Today's powerful and worldwide spreading **Internet** network can be seen as originating in 1960s with a group of scientists and researchers of **ARPA** (*Advanced Research Project Agency*) of the Defence Department of the U. S. whose motivation was to make as full use as possible of the computation resources of those days to meet the demands of numerous potential applications and computer users [14]. At that time it was a very significant topic to fully explore the capacity of the expensive main frame computers located only in a few central places to meet the demands of hundreds and thousands of computer users at their terminals or other less powerful computers nation-wide in the U.S. The first breakthrough was made in late 1969 with the birth of ARPA network or **ARPAnet** that connected four research sites initially and then extended to twenty more in the next year [14]. The success of this computer network system proved that it is possible to realize computer *resource sharing*, *timesharing* and *data communication* across distant locations.

From wired to wireless network, and to the Internet

The ARPAnet was a **wired network** that used telephone lines to transmit data. The wired network has advantages in handling signal attenuation, noise or interference in data transfer. On the other hand it faces some difficulties as well, for instance, in cable installation. To find an alternative, scientists attempted to create a **wireless network**, and indeed the birth and evolution of wireless network was following almost only one step behind the wired one.

In 1970, computer scientist Norman Abramson, recruited by the Information Processing Techniques Office of the ARPA, oversaw a design of **ALOHAnet** that used radio system to transmit data to a host computer in Hawaii [14]. Later, successor scientists found this radio ALOHA network fascinating and took it one step further by creating a **mobile network:** they mounted a computer on a vehicle and connected it with radio to a central host terminal.

The potential of this ALOHA system was recognized by the U.S Army which had it further developed into a **pocket radio network**. As the pocket radios have limitation in transmission distance, this was further developed into a *satellite-based network* – **SATnet**, which used satellites to connect computers in the U.S., and some in Europe [14].

Thus, in the early 1970s, there were three network systems existing: ARPAnet, ALOHAnet and SATnet, with different transmission modes. A very natural question was then considered, *how to connect all of these networks together?* A rational solution is to find a way to make those individual communication modes recognizable and inter-transmittable to one another. This eventually led to a proposal for the **TCP** (*Transmission Control Protocol*) that in turn led to the birth of the early *Internet*.

The first experimental TCP system was developed in 1977 and it successfully realized the data transmission among these three different kinds of networks. The next year, *Internet Protocol* (**IP**) and further the very well known **TCP/IP** system was developed which allowed the Internet to expand in a less expensive way while being more efficient. In 1983 the term "Internet" was officially used for all those networks using TCP/IP protocol system [14].

Certainly, many other contributions had been made before the emergency of the Internet, in both hardware and software domains, e.g. the invention of the *modem* by Bell Telephone Laboratory in 1958; the *gateway* machine resulting from the pioneering work of Catenet in 1973 for routing and handling data transfer between different networks; the establishment of *Ethernet* as a **local area network** (**LAN**), etc [14].

Other two important aspects that support and guarantee the proper operation of the Internet are the **protocol** systems like the **OSI** (*Open System Interconnect*) standards and the **security** measurements like *encryption* techniques, they are always accompanying and evolving with the Internet development throughout the history [14].

Without such innovations and techniques, it would be hard for us to imagine that today's application of and surfing on the Internet could become so popular that almost any people, young or aged, professional or no, would take it as his/her intimate tool to communicate virtually anyone else in the world.

From fixed Wireless to mobile wireless network

We see that the modern Internet was originally developed from the combination of wired and wireless (principally *fixed wireless*) network. These two types of network are alternative to one another in data transfer.

In terms of the main data transfer *backbones* or *carriers* over a long distance, wired net has been more dominant in the later stage especially due to the development of *fibre optic* technology. Fibre optic can transfer data with low *signal attenuation* and *noise interference*, and most notably, very high capacity – one strand of fibre optic can carry many *Peta* (quadrillion) bits of data per second [22]. Such capacity is hard to reach by the wireless network, and deployment of the wireless network normally includes the use or implementation of satellite and other transceivers and relays, etc. which are costly and time consuming.

However, for a short distance, especially in commercial districts of cities, fixed wireless network has its advantage in passing large volume of data over the air without the demand for cables (fibre optic) installations constrained with digging or other municipal or aesthetics restrictions. Today's fixed wireless net can provide capacity of over 100 *Mbps* (mega bits per second) [22].

A common characteristic of the wired and fixed wireless network is that they transfer data from and to the end users at *fixed access points*, e.g. desktop PCs. This fixed access issue posts some limitations to the efficient and convenient use of the data transferred, which is where the **mobile wireless network** plays the role.

On one hand the mobile wireless network can work as an extension or a complementary to the modern main networks (*Internets* or *intranets*), such that the user does not have to sit at a fixed point to access the net and get information delivered or received. Up to now, this is the main stream of currently considered and used mobile wireless network, especially used for *mobile telephony*. On the other hand mobile wireless communication network itself can be an independent communication network, and the earliest one was already developed some decades ago, e.g. for police use or other uses.

There are various mobile wireless network models over the world. In terms of facilities and functionality, the mobile wireless network can be broadly classified into two: *full fledged networks* and *ad hoc networks.*

Full-fledged or full-infrastructured mobile wireless networking:

Full-infrastructured mobile wireless network [22][23] is currently the network most people are familiar with, since it is basically the network used by mobile phones. This system is equipped with all the infrastructure needed for connection of the mobile nodes of the network, like *mobile switching center* (*MSC*), *radio transceiver*, and the like, as so called fully infrastructured. A conceptual architecture of this type of wireless (telephone) network can be depicted as below in Figure 0.1.

Firstly, "Mobile" means the *node* (telephone set for example) can be moveable with a user roaming, this in turn means the battery power of the mobile node shall not be high due to the health constraints to the user. Then the low power condition means the range of the radio transmission of the mobile node shall not be very large.

Thus, an area covered by the mobile data transmission system over radio is typically divided into a number of *cells*, each cell covering an area up to a few hundreds of meters wide corresponding to the mobile node power capacity and other conditions, like noise and barriers to the radio. As shown in Figure 0.1, each cell then needs a fixed **radio transceiver,** also called **base station**

*(***BS***)* to communicate with the **mobile nodes**. The base stations of all the cells are then connected to a main **mobile switching center** *(***MSC***)* through the radio, thus forming a network. The base station works as gateway or bridge to transfer data received from the MSC to the mobile units/nodes in its reachable proximity. The MSC then may connect to a main external network, like the *public switched telephone network (PSTN), integrated service digital network (ISDN)*, or the Internet [16], [22]. That is the way for a mobile node to be able to communicate to other node over the world through its connection to the BS within its current cell. When the mobile unit gets into another cell, there is a connection *handoff* to occur between the previous base station and the new one.

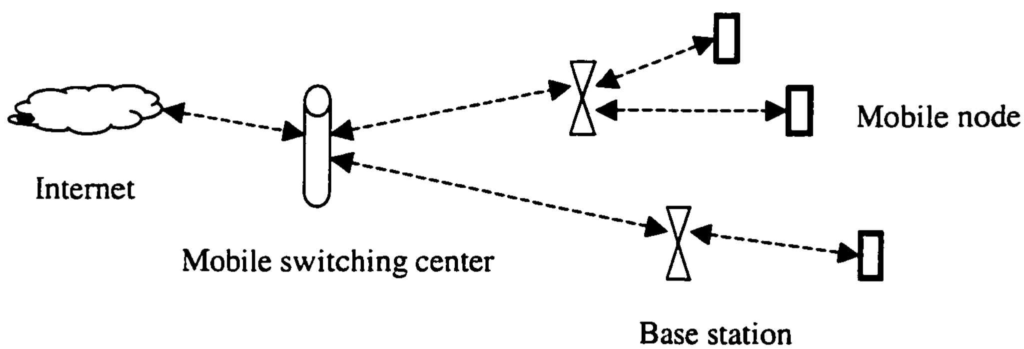

Fig. 0.1. Conceptual architectural diagram for full-fledged mobile wireless network

Ad hoc or infrastructureless mobile wireless networking:

Unlike the full-fledged infrastructure network, this network does not possess substantial infrastructures, even without fixed radio transceiver. Rather, it consists only of mobile units, like phone, laptop computer, PDA (*personal digital assistant*), and the like, without any cables or existing base stations to interconnect them. All the nodes can move in an arbitrary manner and can connect to other units automatically based on certain rules. Each mobile node itself acts as a bridge on one hand to transfer data and maintain a route, and on the other hand to receive data from other mobile nodes [4], [23].

Up to the time this present report was written, there has not existed a substantial ad hoc network over the world, while it has been studied extensively in both the research and the industrial circles. Formal proposals have also been produced for the design and implementation of the ad hoc networks. Presently there are three major standards proposed for ad-hoc networks, they are:

IEEE 802.11 [25],

Hiperlan/2 [26], and

Bluetooth [20], [21].

The IEEE 802.11 and Hiperlan/2 are standards for **Wireless Local Area Network** (**WLAN**). The Bluetooth is a standard for **Wireless Personal Area Network** (**WPAN**).

The emergency of the Bluetooth – one of the expected realizations of the ad hoc wireless network

As just mentioned, Bluetooth is one of the dominant proposals for the realization of ad hoc wireless networks, especially for WPAN.

Like many other technologies, the birth of Bluetooth was propelled by expected market demands and based on other relevant technological developments especially in Internet and wireless phone systems. Over the last decade, we have seen a rapid market development in various personally used devices and instruments like cell phones, walkmans, laptop computers and PDAs. We have also seen various increasing demands for *remote controls*, *virtual meetings*, and so on. From technological point of view, on one hand digital technologies over radio have been developed rapidly especially in wireless telephony, on the other hand those personal devices or household electric appliances are still used in conventional ways (either used separately or connected with various cables). Business and industrial people then have observed that there is a huge business potential in making such devices interconnected and inter-communicable wirelessly.

As early as 1994, a concept debuted in the Ericsson laboratories, its primary goal was to supply users with cheap, robust, flexible and low power consumption chips that can replace cables and interconnect devices in short range based on the existing technologies. This concept was eventually named **Bluetooth** after the legendary Scandinavian king Harald Bluetooth, who united Denmark and Norway in the 10th century [17], [22].

In 1998, Ericsson Mobile Communications, Intel, IBM, Toshiba and Nokia Mobile Phones formed the *Bluetooth Special Interest Group* (*SIG*) whose intention was to create this new technology and promote it on the market. Afterwards, more companies joint in rapidly, two years later almost 2000 companies became members of the **Bluetooth SIG** and are now working on the development and promotion of Bluetooth products [17], [22].

The very positive responses and expectations to the Bluetooth proposal are well justified because of its prominent technological strength. As stated before, a realistic and promising application of the Bluetooth technology is in *personal area network* (**PAN**) in which various personal use devices like mobile phone, laptop or palmtop computer, etc., can be organized and connected as a net to make various information transitions or manipulations. Further, Bluetooth can serve not only for network applications, but also can be used for other independent usages. E.g. a very expected application of the Bluetooth chips is to replace the cables or physical links among or within various devices or equipment, like the cable between a computer and its keyboard, or a printer. There are many other potential usages of the Bluetooth chips or cards or other types, such that the so called "non-wired" world shall be no longer for just fictional stories. The present report is focused on the Bluetooth fundamentals and a review on some selected research articles. The report is organized as follows.

Part 1 is a brief review on Bluetooth core specifications and its profile specifications. Through the summaries of the core specifications we can get a general impression and fundamental knowledge about the Bluetooth technology, the related terminology and the relationships among different protocol layers. Through the summaries of the profile specifications we can get the rules to follow to apply the technology into various applications.

Part 2 is a literature review on some of the research papers dealing with the Bluetooth technology. Through this part we can get indications about the potential problems or constraints in the implementation of the Bluetooth technology into applications, and the possible solutions or improvements to them. More importantly, we can learn from these papers about the approaches and methodologies used by the researchers in how to reach those solutions.

The last part is a short conclusion to follow, where the significance of studying the Bluetooth technology, the perspectives of the applications of the technology, are addressed.

Part 1. Bluetooth Protocols Review

1.1 Technology overview

The basic mission of the Bluetooth SIG is to provide people with license-free technology for interconnection of devices, and all the Bluetooth SIG members have the right to produce and sell the Bluetooth chips and use the common logo. To reach these objectives with rather open constraints, the Bluetooth technology then must have some rules or standards to follow.

In a high-level point of view, the requirements for Bluetooth chips are rather simple, they include:

(a) the Bluetooth transceivers shall work around the *Industrial, Scientific and Medical* (ISM) radio frequency band 2.4GHz, with relatively high speed, up to 721 kbps, or nominally 1 Mbps.

(b) The Bluetooth chips shall possess other properties, like: low cost, around $5.00 US per unit, low power consumption; short transmitting range, 10 meters (extended 100 m); open standard; and small sized and light weighted [11], [4].

Typically, simplicity in high-level view means that lot of work must be done on the lower levels and behind the curtain. The above appealing objectives could be reached in different ways in a low-level view with different manufacturers or vendors, and there is thus an unavoidable demand to reinforce the interoperability of the products from different vendors. This calls for the establishment of the Bluetooth protocol system like any other electronic communication technology does.

Currently available Bluetooth specifications are in its 1.1 version and released in 2001 [20] [21]. The specifications are divided into two files:

(a) *Bluetooth 1.1 Specifications Core* [20], and

(b) *Bluetooth 1.1 Profiles* [21].

The core file defines the technical fundamentals of the Bluetooth products, i.e. how the technology works. The profiles file defines the rules for various applications of the technology to follow, i.e. how the technology is used. In the next two sections we summarize the specification core and the profiles.

1.2 The core specifications – how the technology works?

The core specifications are a stack of protocols with layered architecture. Because of the space limit of this paper, only the representative layers: Radio, Baseband, LMP, HCI, L2CAP, RFCOMM and SDP are summarized here.

The Bluetooth protocol architecture can be simply depicted as in Fig. 1.1. We notice that the Bluetooth protocol stack is not just a copy of the known OSI protocol system [11]. This difference basically follows from the consideration that the technology should reflect the special characteristics and applications of the Bluetooth and render it to work in a more efficient way. We can also understand that the upper layers of the protocol stack may be adaptable to different environments or applications, e.g. to *Wireless Application Protocol (WAP)* system, while the lower layers are quite invariable as they are more fundamental [11], [22].

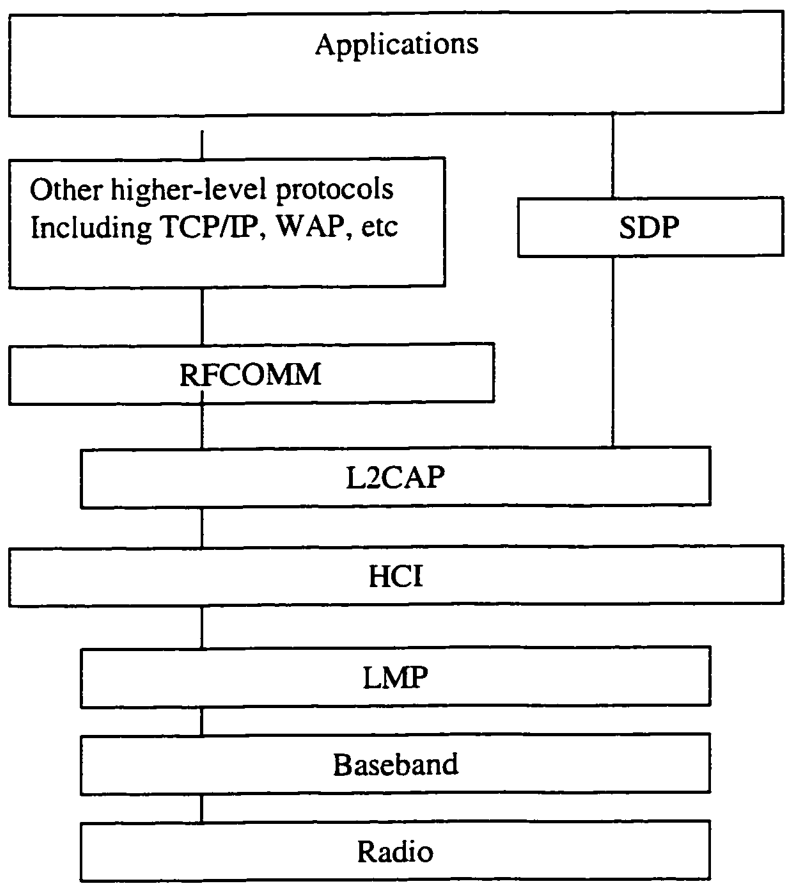

Fig. 1.1 Bluetooth technology protocol stack (refer to [11])

1.2.1 Radio

The first or lowest protocol layer is the physical layer concerning the *radio frequency* (**RF**) spectrum. As mentioned before, the Bluetooth technology uses the ISM 2.4GHz RF band. The advantage of using this band is that it is free (unlicensed) and also globally available, thus facilitating the universal deliverability of the Bluetooth products. This band takes 83.5 MHz on the RF spectrum spreading from 2.400 to 2.4835 GHz in most of the Bluetooth SIG member countries. This band is then divided into 79 channels, with 1 MHz for each channel. In France, Japan and Spain only 23 channels are currently available there since these countries do not have that much bandwidth left freely around the 2.4 GHz range.

The Bluetooth uses *fast frequency hopping* (**FFH**) and *time division duplex* (**TDD**) mode in sending data. The *frequency hopping rate* is 1600 per second, each *time slot* is 0.625 ms (*milliseconds*), and the data packet is transmitted in length of 1, 3 or 5 time slots.

Other important parameter for a radio transmission is its power level, with regulation and health constraint factors, the Bluetooth devices power falls and can optionally vary in three classes:

Class 1: for long range transmission up to 100 m, and maximum output 20dBm;

Class 2: for ordinary range transmission up to 10 m, and maximum output 4dBm.

Class 3: for short range transmission up to 10 m, and maximum output 0dBm.

The 0dBm is also the nominal power level of the Bluetooth radio interface. The radio transmission modulation is chosen to use simple *Gaussian Frequency Shift Keying* (**GFSK**): binary one is represented by a positive frequency deviation, binary zero by a negative deviation.

For receiving, a receiver should have a sensitivity level, in terms of *bit error rate* (**BER**), 0.1%, which means the power level must be –70dBm or higher.

1.2.2 Baseband

The *Baseband* is the physical layer which sits just above the Radio layer and is responsible for the tasks contained in the physical layer and some other tasks in higher layers of the OSI/ISO model, such as:

Link control,

hopping selection,

synchronization,

data transmission,

error correction,

logical channels division,

data whitening (scrambling), and

security.

The physical channel is a sequence of the referred 79 or 23 RF spectrum of 1 MHz each around the radio band of 2.4 GHz through a **pseudo-random hopping**. Two or more and up to eight Bluetooth devices can use the same channel, forming a **piconet**. In each piconet there is one and only one **master** while there can be one or up to seven **slaves.** The position of master and slave can be alternated if in need. The hopping sequence is unique for a specific piconet and is determined by the Bluetooth **device address** of the master. The phase of a hopping sequence is determined by the master's clock. The channel is divided into time slots and each slot corresponds to an RF hop frequency. As there are 1600 hops each second, it comes out each time slot is 0.625 ms long.

The master and the slave alternatively transmit data through TDD. The master shall start its transmission in even-numbered time slots only, and the slave shall start in odd-numbered slots only. The packet start should be aligned with the slot time start.

There have been defined five *logical channels* for Bluetooth to transfer different types of information: Two channels *Link Control* **LC** and *Link Manager* **LM** are used in the link level. Other three channels are *User*

Asynchronous **UA**, *User Iso-synchronous* **UI** and *User Synchronous* **US** channels to carry user information.

The Baseband handles two types of physical links:

Synchronous Connection-Oriented **SCO** and

Asynchronous Connection-Less **ACL** links.

The SCO link is a symmetric *point-to-point* link between a master and a single slave in a piconet. The master maintains the SCO link by using reserved slots at regular intervals, taken as a *circuit switched* type. The SCO link was designed mainly to carry voice information in speed up to 64kbps. For voice encoding, either *pulse-code modulation* (**PCM**) or *Continuous Variable Slope Delta modulation* (**CVSD**) is used [11]. The master can support up to three simultaneous SCO links while slaves can support two or three SCO links. SCO packets are never retransmitted, as it was assumed for real-time transmission.

The ACL link is a *point-to-multipoint* link between the master and all the slaves participating in a piconet, and is intended for packet transfer, both asynchronously and isochronously (i.e. time-sensitive). The asynchronous channel may support up to 721kbps downlink and 57.6kbps uplink or symmetrically 433.9 kbps in both directions. The master can use unreserved slots for the SCO links to establish an ACL link on a per-slot basis to any slave, including the slave already engaged in an SCO link. Only a single ACL link can exist. For most ACL packets, packet retransmission is applied to assure data integrity. This is a big difference, in comparison to other technologies like Ethernet or ATM, where retransmission is not applied and it is up to higher layers (for example TCP) to take care of retransmission if needed. In the ACL link, a slave is allowed to occupy the slot(s) to send packet(s) if and only if the master informed it so in the preceding slot. This solution is the Bluetooth *medium access control mechanism*: master decides who shall access radio link at a time.

Other physical parameters including the Bluetooth *device addressing* and *packet formats* are also defined in the Baseband layer.

The main functionality of the Baseband layer is the **channel control**, and will be elaborated here with moderate details. The channel control is realized and embodied in its network – piconet and scatternet formation and then in the data transmission schemes.

Piconet formation and connection establishment:

There are *seven states* for adding slaves or make connections in a piconet. They are: *page, page scan, inquiry, inquiry scan, master response, slave response and inquiry response*. These states can be grouped into two major states: *Standby* and *Connection*.

The **standby** state of a unit is the default low power state where only the native clock is running and there is no interaction with any other device.
In the **connection** state, the master and slave can exchange packets, using the *channel (master) access code* and the *master clock.* The other states like page, inquiry, etc., shall be further mentioned in the following context.

In a **Connection Setup** (Inquiry/Paging) process, there are two cases: one case is when nothing is known about a remote device. In this case both inquiry and page procedures have to be followed. The other case is when some details are known about a remote device, then only page procedure is executed.

The **inquiry procedure** enables a device to discover which devices are within its transmission range, and to determine the addresses and clocks for the devices. It works like this: a source unit sends out inquiry packets and gets into the **inquiry** state. A destination unit may happen to be in the **inquiry scan** state when it can receive the inquiry packets. The destination unit will then send an inquiry reply to the source and enter the **inquiry response** state.

Once the inquiry procedure completed, the **paging procedure** follows immediately and a connection can be established. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock estimate will accelerate the setup procedure. A unit that establishes a connection will carry out a page procedure and will automatically be the master of the connection. The procedure works as follows:

The *source device* pages the destination and gets into **Page** state;

The *destination* receives the page and gets into **Page Scan** state;

The destination sends a reply to the source and gets into **Slave Response** state;

The source sends a *Frequency Hopping Synchronization* (**FHS**) packet to the destination and gets into **Master Response** state;

The destination sends it's second reply to the source and gets into **Slave Response** state;

The destination & source then switch to the source channel parameters.

The **Connection** state starts with the master sending a **poll packet** to verify that slave has switched to the master's timing and channel frequency hopping, and the slave can respond with any type of packet. The master schedules the transmission based on traffic demands to and from the different slaves. In addition, it supports regular transmissions to keep slaves synchronized to the channel. A Bluetooth device in the Connection state can be in any of the four modes: *Active, Hold, Sniff* and *Park.*

In the **active** mode, a Bluetooth unit actively participates on the channel. Active slaves listen in the master-to-slave slots for packets. If an active slave is not addressed for some time, it may *sleep* until the next new master transmission.

In the **sniff** mode, a slave device listens (synchronized) to the piconet at reduced rate and enters power-saving modes, thus reducing its duty cycle.

In the **hold** mode, a slave device listens to the piconet with only its internal timer running, and enters other deeper *power-saving* modes. When the unit transits out of hold mode, data transfer is restarted instantly.

In the **park** mode, a device is still synchronized to the piconet but does not participate in the traffic and saves power. Parked devices have given up their MAC addresses (according to the specifications, the MAC address is valid only when a slave is active on the channel) while occasionally listen to the traffic of the master to re-synchronize and check on broadcast messages.

Scatternet formation and data communication:

A group of piconets that are interconnected is called a **scatternet**. Within the same scatternet, each piconet still keeps its own independent channel hopping sequence and phase determined by the respective master, and the packets carried on the channels are preceded by different channel access codes determined by the master addresses too.

The rules that govern the scatternet formation are rather simple and open: two piconets can be connected together through a device who acts as a bridge or gateway and belongs to these two piconets. Consequently a unit can participate in two or more piconets, and its (role) presence in those different piconets is through applying the time multiplexing. Such bridge/gateway device can act as a slave in several piconets, but can act as a master in one piconet only.

If needed, an existing master or slave may wish to swap roles, and this can be done in two steps:

The first step is the TDD switch between the considered master and slave, followed by the piconet switch from the old one to a new one.

The second step is then to let other slaves of the old piconet be transferred to the new piconet after they have got the FHS packets, and they will then use the new piconet parameters defined by the new master and the piconet switch is completed.

Other functions of the Baseband, like error correction, flow control, synchronization, and security, are not elaborated here because of the space limitation of this paper. See [20] for further details.

1.2.3 Link Manager Protocol (LMP)

The *LMP* mainly takes care of link set-up, authentication, link configuration and the security. It uses the services of the underlying *Link Controller* (**LC**) to fulfill all the tasks.

The LMP is working principally through various **PDUs** (*protocol Data Units*), which are sent from one device to another in a connection. The link manager PDUs are either Mandatory (must be supported), or Optional (optionally supported). The PDUs are sent as single-slot packets, and they have higher priority than user data and can pass immediately through the link even if it is congested. The tasks or the types of the PDUs are numerous, they can be roughly classified as:

- Response: in various device detection, and link setups;
- Pairing of devices;
- Power control;
- Clock synchronization between master and slave;
- Switching the connection modes: hold, sniff and park;
- Authentication;
- Encryption;
- Paging;
- Connection establishment;
- Quality of service negotiation;
- Testing;
- Error handling ;

1.2.4 Host Controller Interface (HCI)

As implied by the term interface, the *HCI* provides a uniform method and a command interface to the Baseband controller and link manager, and access to the Bluetooth Baseband capabilities. The HCI can be seen existing across three sections, the Host, the Transport Layer, and the Host Controller.

The "**Host Controller**" means the HCI-enabled Bluetooth device, its main component is the *HCI Firmware* which implements the HCI Commands for the Bluetooth hardware by accessing Baseband commands, link manager commands, hardware status registers, control registers, and event registers.

The "**Host**" means the HCI-enabled Software Unit, its main component is the *HCI Driver* that receives asynchronous notifications of HCI events and then parses the received event packet to determine which event occurred.

The Host Controller "Transport Layer" refers to several layers that may exist between the *HCI driver* on the host system and the HCI firmware in the Bluetooth hardware. These intermediate layers provide the ability to transfer data without intimate knowledge of the data being transferred.

The **HCI Commands** provides a uniform method of accessing the Bluetooth hardware capabilities. These commands include mainly:

(a) HCI Link control commands, which provide the Host with the ability to control how the Bluetooth piconets and scatternets are established and maintained through the link layer connections to other Bluetooth devices.

(b) HCI Policy commands, which are used to affect the behavior of the local and remote LM, and to provide the Host with methods of influencing how the LM manages the piconet.

(c) The Host Controller & Baseband Commands, which provide access and control to various capabilities of the Bluetooth hardware, the Link Manager, and Baseband. These commands can be used to modify the behaviour of the local device.

The **Informational Parameters** are the parameters which are fixed by the manufacturer of the Bluetooth hardware and thus cannot be modified by the host device. They provide information about the Bluetooth device and the capabilities of the Host Controller, Link Manager, and Baseband.

The **status parameters** are the parameters which provide information about the current state of the Host Controller, Link Manager, and Baseband, and thus cannot be modified but reset by the host device.

The **Testing commands**, are the parameters which are used to provide the ability to arrange conditions for testing, and to test various functionality's of the Bluetooth hardware.

The HCI also handle *flow control, events control and error control.*

1.2.5 Logical Link Control and Adaptation Protocol (L2CAP)

The *L2CAP* resides over the Baseband Protocol while it is the base layer of all other higher layer protocols like RFCOMM, TCS and SDP. It provides connection-oriented and connectionless data services to those upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. The L2CAP Specification is defined for only ACL links but not for SCO links, since real-time voice traffic is transferred directly over reserved bandwidth. Some of the L2CAP services and functions are presented below:

Protocol requirements supporting, including:

Protocol Multiplexing: L2CAP must support the protocol multiplexing since the Baseband Protocol does not do it, and it must be able to distinguish between upper layer protocols such as RFCOMM , SDP, and Telephony Control.

Segmentation & Reassembly (**SAR**): The packets transmitted with the Bluetooth Baseband Protocol are of smaller size than those on higher layer protocols that are designed to use larger packets. Thus large L2CAP packets must be segmented into multiple smaller Baseband packets prior to transmission. Reversely, multiple Baseband packets received should be reassembled into a larger L2CAP packet following a simple integrity check. This SAR functionality is absolutely necessary to support efficient use of bandwidth either above or below the L2CAP layer.

Quality of Service

The L2CAP supports information exchange on the *quality of service* (**QoS**) expected between two Bluetooth units and ensure the QoS contracts are honoured.

L2CAP General Operation

The L2CAP operation is based on the concept of channels and *channel identifier* (**CID**). The CIDs are local names representing a logical channel endpoint on a device.

Operation between Devices

For connection-oriented data channels, a CID identifies each endpoint of the channel, and a connection is setup and kept between the two devices. For the connectionless channels the CID on the source represents one or more remote devices, and such channels restrict data flow to a single direction of a channel "group".

Operation between Layers

It follows a set of general rules as: It must transfer data between higher layer protocols and the lower layer protocol. It must also support a set of signaling commands for use between L2CAP implementations. It should accept certain types of events from lower layers and generate events to upper layers.

To ensure the Segmentation & Reassembly ability, from the source sending side an L2CAP implementation exposes the outgoing large higher layer packets into "**chunks**" that can be passed to the Link Manager via the HCI. On the receiving side, an L2CAP implementation receives "chunks" from the HCI and reassembles those chunks into L2CAP packets using information provided through the HCI and from the packet header.

Other L2CAP features, including Data Packet Format, Signaling, Configuration Parameter Options, and number of Service Primitives, shall not be elaborated here because of the space limitation, see [20].

1.2.6 RFCOMM Protocol

The *RFCOMM* in the Bluetooth is based on and it partially adapts the ETSI standard TS 07.10 [27] to emulate serial ports over the L2CAP protocol. It supports up to 60 simultaneous connections between two Bluetooth devices. Following is the description of the main contents of the RFCOMM protocol.

Device Types and Control Signals: Devices accommodated by the RFCOMM are Basically divided into two:

Type 1 for communication end points such as computers and printers;

Type 2 for those that are part of the communication segment, e.g. modems.

For control Signals, the RFCOMM emulates the nine circuits of an RS-232 interface.

Null Modem Emulation: When transferring non-data circuit modem between two devices of same type connected together, the RFCOMM uses the same way that TS 07.10 [27] transfers the RS-232 control signals to create an implicit null modem.

Multiple Emulated Serial Ports: The number of open emulated ports that RFCOMM can support is up to 60, while it is device implementation-specific. Ongoing connection between a client and a server is identified by a *Data Link Connection Identifier* (**DLC**I) that is unique within one RFCOMM session.

Other main services handled by the FRCOMM protocol include:

- Methods Used for Flow Controls, like L2CAP Flow Control;
- RFCOMM Flow Control;
- Wired Serial Port Flow Control;
- Port Emulation and Port Proxy Entities;
- Service Registration and Discovery;
- Reliability maintenance;

- Low Power Modes, i.e. hold, sniff or park mode, put to idle devices.

1.2.7 Service Discovery Protocol (SDP)

It is expected that finding and making use of services available in a network will become increasingly important. Services such as printing, fax sending, are common to us, but more technical services such as network bridges, e-Commerce facilities, are also becoming common to us and dynamical evolution of services should be expected. Thus, the various aspects of *service discovery*, i.e. how to access the services, advertise the services, choose among competing services, bill for services, and so on are addressed more often than before.

The *SDP* is to provide means to discover which services are available and to determine the characteristics of those available services from a concerned device in the proximity. The Bluetooth needs Service Discovery protocol because the available services may change dynamically, for instance due to a motion of device, new services joining the network or leaving the network. Furthemore, Bluetooth environment has its own unique characteristics.

The SDP design is simple and with minimal requirements on the underlying transport. It uses a *request/response* model where each transaction consists of one request protocol data unit (PDU) and one response PDU. And the SDP can work over a reliable or even unreliable packet. With this protocol, a client must receive a response to each request before issuing another request on the same L2CAP connection.

In Bluetooth SDP, a **service** means a measure that can provide information, perform an action, or control a resource on behalf of another entity. Such measure may be implemented as software, hardware, or a combination of the two. All of the information about a service maintained by an SDP server is contained within a single service record that in turn consists entirely of a list of various service attributes. All the services are defined in number of different classes.

Service discovery is done in various means, e.g. *Searching and Browsing.* The **Search** transaction allows a client to find or to retrieve the services from the particular service records according to the values of attributes of those records. **Browsing** means looking to see what services are actually being offered. In SDP, the mechanism for browsing for services is based on an attribute, the BrowseGroupList, shared by all service classes.

Other point here is that the Bluetooth SDP does not define methods for accessing services after they are discovered, rather they can be accessed in various ways. This might include the use of other service discovery and access mechanisms. In other words, Bluetooth services can be discovered using the Bluetooth SDP or can be accessed using other protocols defined with the Bluetooth technology.

1.3 The profiles – how to use the technology in applications?

The Bluetooth core specifications provide us with the principle foundations for the technology. The rest of this chapter is concerned with how to use or follow all of these fundamental parameters into various applications. Since the Bluetooth specifications are open to the public and all the applications may be realized through various Bluetooth products produced by different manufacturers or vendors, there is a call to ensure the *interoperability* of the products. This is where the Bluetooth profiles specifications come into being.

The Bluetooth profiles are developed to describe how implementations of user models – a number of user scenarios using Bluetooth transmission – are to be fulfilled. A profile can be viewed as a *vertical slice* through the protocol stack. A profile consists of number of mandatory terms and parameter ranges from each protocol, thus reaching a risk free objective of interoperability.

The Bluetooth profile *structure* and their *dependencies* can be presented as in Fig.1.2 below. There are dependencies that exist between a profile and other profiles directly or indirectly containing it. For instance, the Intercom profile is dependent on TCS-BIN-Based and Generic Access profiles.

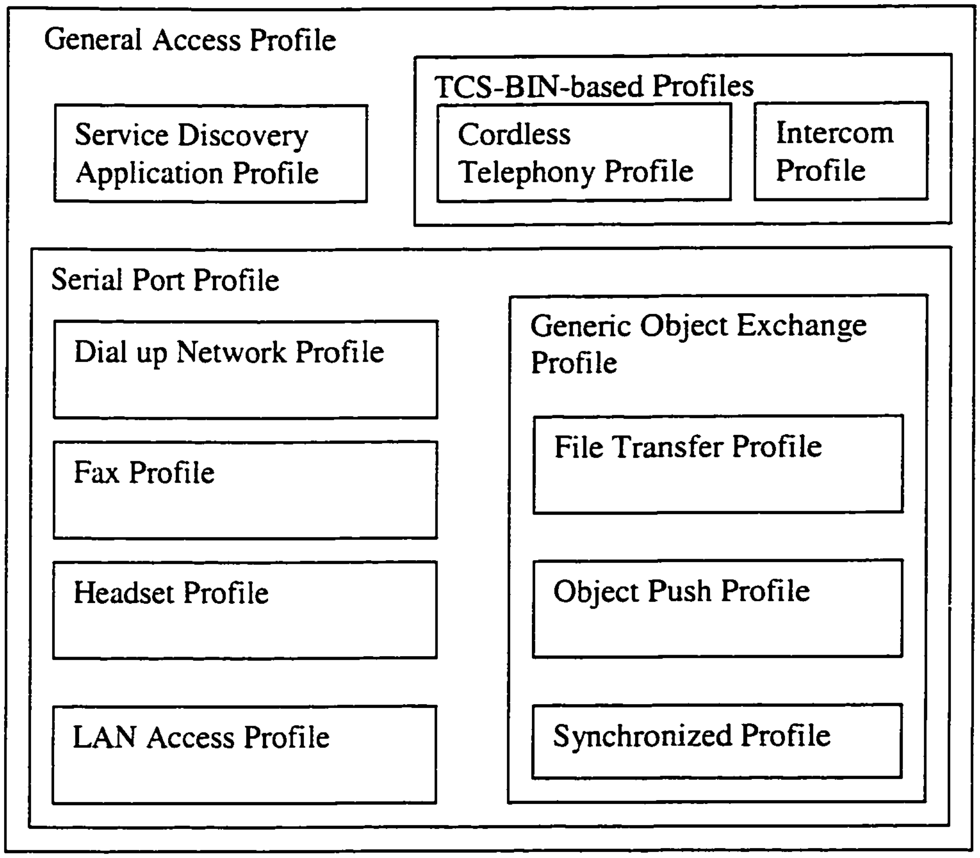

Fig. 1.2 Bluetooth technology profile architecture (Source [21])

1.3.1 Generic Access Profile (GAP)

As the name implies, the *GAP* is the fundamental profile, governing other profiles. It defines the generic procedures related to discovery of Bluetooth devices and related to link management aspects in a connection. This profile describes the use of the lower layers of the core specifications stack (LC and LMP), but also higher layers (e.g. L2CAP, RFCOMM) for security related alternatives. If a Bluetooth device does not conform to any other profile, it shall conform to this profile to ensure basic interoperability.

Configurations/Roles:

In this profile for the descriptions of the roles that two devices involved in a Bluetooth communication can take, the generic notion of the *A-party* (the **paging device** in case of link establishment, or **initiator** in case of another procedure on an established link) and the *B-party* (**paged device** or **acceptor**) is used. The A-party is the one that, for a given procedure, initiates the establishment of the physical link or initiates a transaction on an existing link. The initiator and the acceptor generally operate the generic procedures according to this profile or another profile referring to this profile.

User Requirements/Scenarios:

The Bluetooth user should in principle be able to connect a Bluetooth device to any other Bluetooth device. Even if the two connected devices don't share any common application, it should be possible for the user to find this out using basic Bluetooth capabilities.

Profile Fundamentals:

Profile fundamentals state the requirements on names, values and coding schemes used for names of parameters and procedures on the user interface level. They define modes of operation that are not service or profile-specific, but generic to all profiles. It defines the general procedures that can be used for discovering identities, names and basic capabilities of other Bluetooth devices that are in a mode where they can be discoverable, while only procedures where no channel or connection establishment is used are specified. It defines the general procedure for how to create bonds between Bluetooth devices. It describes the general procedures that can be used for establishing connections to other Bluetooth devices.

1.3.2 Service Discovery Application Profile (SDAP)

The *SDAP* profile defines the protocols and procedures for a service discovery application to discover services registered in other Bluetooth devices and retrieve desired available information using the Bluetooth Service Discovery Protocol (SDP).

Profile Stack: The service discovery user application (SrvDscApp) involves a *local device* (LocDev) that sends service inquiries and receives service inquiry responses from the SDP client servers of *remote devices* (RemDevs).

User Requirements or Scenarios for this profile may be one of the following types:

Search for services by service class, or search for services by service attributes. In other words, searching for known and specific services.

Service browsing, which represents a general service search or search for which services are available. The Bluetooth user should in principle be able to connect a Bluetooth device to any other Bluetooth device. Even if the two connected devices don't share any common application, it should be possible for the user to find this out using basic Bluetooth capabilities.

Profile Fundamentals: Before any two Bluetooth-equipped devices can communicate with each other, the devices need to be powered-on and initialized and a Bluetooth link has to be created.

Service discovery can be initiated by either a master or a slave device at any time within the same piconet. A slave in a piconet may also initiate service discovery in a new piconet while it should notify the master of its original piconet that it will be unavailable (e.g. into a hold mode) for a certain time.

1.3.3 The TCS-BIN-Based protocol

This profile includes two sub-profiles, Cordless Technology Profile (CTP) and Intercom Profiles.

1.3.3.1 Cordless Technology Profile (CTP)

This profile defines the features and procedures required for interoperability between different units actively involved in a "3-in-1 phone" scenario. In this scenario Bluetooth products can be used as a short-range bearer for accessing fixed network telephony services via a base station. For other cordless-only or cordless telephony services in a PC such as used in a residential or small office environment, the 3-in-1-phone use case can also be generally applied. This is the reason why the profile is named "Cordless Telephony".

1.3.3.2 Intercom Profiles (IP)

This *IP* profile is also referred to as *walkie-talkie* usage in the regime of Bluetooth. It is about the requirements for the support of the intercom functionality within the 3-in-1 phone use case. The requirements include the end-user services, the features and procedures required for interoperability between Bluetooth devices in the 3-in-1 phone use case.

1.3.4 Serial Port Profile (SPP)

The Serial Port Profile defines the features and requirements for setting up between two Bluetooth devices the emulated serial cable connections using RFCOMM. And it is essentially about the procedures or protocols used by devices for RS232 (or similar) serial cable emulation. Through a virtual serial port abstraction this profile takes care of the Bluetooth cable replacement of legacy applications.

This profile defines how to support *Point to Point Protocol* **PPP** networking over serial cable emulation in the cases of: LAN Access for a single Bluetooth device,

or for multiple Bluetooth devices, PC to PC.

Protocol Stack include: The Baseband, LMP and L2CAP, which are the OSI layer 1 and layer 2 Bluetooth protocols; and RFCOMM which is the Bluetooth adaptation of GSM TS 07.10, providing a transport protocol for serial port emulation; and SDP which is the Bluetooth Service Discovery Protocol.

Profile Scenario is applicable to setting up virtual *serial ports* (or equivalent) and connections on two devices as if there were a real serial cable connecting them (with RS232 control signalling).

There will be only one connection at a time, and consequently only point-to-point configurations are considered, but there is no limitation on concurrence. Multiple executions of this profile can be run concurrently in the same device.

Profile Fundamentals:

It is *optional* in executing this profile to use security features such as authorization, authentication and encryption, while it is *mandatory* to provide support for authentication and encryption.

This SPP profile includes number of higher level profiles as follows.

1.3.4.1 Headset Profile (HP)

It defines the requirements and features to support the "**Headset**" use case, essentially for devices implementing the usage of "Internet Bridge" model. Typical example devices include headsets, personal computers, and cellular phones, etc.

1.3.4.2 Dial-up Networking Profile (DNP)

It defines the requirements and features to support the "**Dial-up networking**" use case, essentially for devices implementing the usage of "Ultimate Headset" model. Typical example devices include modems and cellular phones.

1.3.4.3 Fax Profile (FP)

The fax profile defines the requirements and features to support the "**Fax**" use case, essentially for devices implementing the fax part of the usage of "Data Access Points, Wide Area Networks". Typical example devices include Bluetooth cellular phone or modem that can be used on a computer as a wireless fax modem to send or receive a fax message.

1.3.4.4 LAN Access profile (LAP)

This LAP profile defines LAN Access using Point to Point Protocol PPP (or others in the future) over RFCOMM. It firstly defines how Bluetooth-enabled devices access the services of a LAN using PPP, then defines how to form a network consisting of two Bluetooth-enabled devices using the same PPP mechanisms.

1.3.4.5 Generic Object Exchange Profile (GOEP)

It defines the requirements and features to support the "**Object Exchange**" usage model such as Synchronization, File Transfer, or Object Push model, and essentially as a generic profile for all application profiles using the **OBEX** (object exchange) protocol.

Profile Scenarios is applicable to usage of a Server by a Client to push data object(s), or usage of a Server by a Client to pull data object(s)

This profile includes other three sub-profiles further, they are Object Push Profile (OPP), File Transfer Profile (FTP) and Synchronization Profile (SP).

1.3.4.5.1 Object Push Profile (OPP)

It defines the requirements and features to support the "**Object Push**" usage model which uses the underlying *Generic Object Exchange Profile* (**GOEP**) to define the interoperability requirements for the protocols needed by applications. Typical example scenarios include: Object Push, Business Card Pull & Business Card Exchange, and the like that involve the pushing/pulling of data objects between Bluetooth devices.

1.3.4.5.2 File Transfer Profile (FTP)

It defines the requirements and features to support the "**File Transfer**" usage model which again uses the underlying Generic Object Exchange Profile (GOEP) to define the interoperability requirements for the protocols needed by applications. Typical example scenarios may involve a Bluetooth device to browse, transfer and manipulate objects on or with another peer device.

1.3.4.5.3 Synchronization Profile (SP)

Synchronization profile defines the requirements and features to support the "**Synchronization**" usage model which again uses the underlying Generic Object Exchange Profile (GOEP) to define the interoperability requirements for the protocols needed by applications. Typical example scenarios may involve a computer to instruct a mobile phone or PDA to exchange PIM data, or vice versa, or automatically to start synchronization when two Bluetooth devices are getting into proximity.

Part 2. Research Literatures Review

Soon after the Bluetooth technology was proposed, many researchers became interested in the possibilities it offered for networking and a number of research papers has been written on the Bluetooth. A study of these research articles is important for becoming aware of the research frontiers in the area, and gaining some further insights into the Bluetooth technology development, its possibilities and limitations. Following is a review on some selected research papers dealing with the Bluetooth. They represent the current research concerns on the evaluation, exploration, or improvements for the Bluetooth technology.

2.1 Characterization of Bluetooth Scatternet Topologies

Bhagwat and Rao [2] studied some mathematical properties of *scatternet topologies*, and propose a way to enumerate such topologies. Their methodology is based on *graph theory*, and use *binary matrices* as an alternative.

The authors firstly demonstrated that without proper understanding of the mathematical fundamentals of the scatternet topologies, it is hard to reach a correct enumeration of the topologies.

An intuitive approach to enumerate the scatternet topologies is to consider the space of general topologies of a graph, then identifying the number of possible scatternet topologies with its various constraints from that space. This is simple in concept but it may not be feasible when the number of nodes is getting large. According to graph theory, the total number of possible topologies of *n* nodes is $2^{n(n-1)/2}$. If n is equal to or larger than 7, enumerating all the topologies is not feasible even by computer.

To solve this problem and to be a bit more focused, the authors studied basically on *tree shaped scatternet*. Their fist step is to apply **labeled graphs** to the scatternets with reasonable constraints, e.g. the number of degrees of a master or slave node, or the maximum edges of a scatternet. In general, scatternet topologies can be represented as a **partition** of an (even) integer, each part of the partition represents the number of *edges* incident on a node of the scatternet. Thus the enumeration of the topologies can be obtained from the **partition result** that is reduced from the primitive enumeration. Such partition can be constructed by an existing algorithm (Havel and Hakimi) [28], [29].

Further, as Bluetooth nodes are exactly classified as masters and slaves, the enumeration of scatternet topologies can be studied with the theory of

labeled bipartite graphs. The enumeration of the topologies of the scatternets thus can be farther reduced by this labeled bipartite approach.

As mentioned before, the scatternet topologies can be represented with matrices. Corresponding to the basic *labeled bipartite graph*, the scatternet can be represented with *adjacency matrices* of N*N, where each element of the matrix will have value 1 if the corresponding column and row variables (nodes) have edge between them (thus adjacent to each other), otherwise zero (and all the diagonal elements are zero too). The enumeration of the scatternet will then identical to the enumeration of this kind of binary matrix.

Corresponding to the bipartite graph representation, the scatternet can be represented with M (masters) * N (slaves) binary matrix with row sum vector (a row sum is equal to the degrees of a master node) and column sum vector (a column sum is equal to the degrees of a slave node). To enumerate topologies of a specific scattrernet, the authors included other natural constraints, e.g. the slave or master degree sequence, into the consideration. The enumerating polynomial can be represented as a product of elementary symmetric functions. But such product function is hard to handle when the number of nodes in a scattenet getting large. One solution suggested is to substitute the product of elementary symmetric functions with proper *power sum* expressions, such solution is presented in another paper of the authors.

The above are the theoretic studies of the mathematical properties of scatternet topology. Numerical examples of results obtained in this paper are given below. Suppose there be 10 nodes in total and the number of master or slavers is shown on the first and the second column, the third column is the total number of edges among these nodes, and the fourth column is the total number of topologies (the number of slaves acting as bridges has also been taken into account.) This table fairly gives us impression about the magnitude of the scatternet topology space.

Table 2.1 Numerical examples of scatternet topologies

Masters	slavers	edges	No. of topologies
2	8	9	1024
2	8	11	1792
2	8	≤ 12	6848
3	7	9	45927
3	7	11	76545
3	7	≤ 12	244944
4	6	9	276480
4	6	11	186624
4	6	≤ 12	820800

2.2 A Bluetooth Scatternet Formation Algorithm

The authors of this paper, Law and Siu [6], declared that compared with other existing suggested scatternet formation approaches they constructed an algorithm with the lowest **time complexity** O(log n) in the scatternet formation process, and **message complexity** O(n) in sending message between devices during that process, where n is the number of total nodes in a scatternet.

The paper starts with the observations of the importance of these two complexities, time and message complexity, in Bluetooth scatternet formation performance. *Time complexity* measures how efficiency an ad hoc scatternet can be formed. Obviously large time complexity shall not be acceptable to the user. *Message complexity* measures the number of messages sent between the nodes during the formation. An important factor is that the Bluetooth is low power mounted, therefore low message complexity is definitely expected to achieve the saving of *power consumption.* Other measures related to Bluetooth network quality are:

the number of piconets,

the maximum degrees of the devices, and

the network diameter.

Based on these measures, the authors thus devised an algorithm for scatternet formation leading to a *quasi-optimal* number of piconets and limiting the degree of single shared slave to 2.

Their algorithm is based on a concept of a **component**. A component can be a single disconnected device or a set of connected devices, like a piconet or a scatternet itself. Each component has one and only one **leader**. In a piconet the leader is its master. In a scatternet the leader is one of the masters of all the piconets. The role of a leader is to know and to connect the possible slaves. A leader can be retired and can be a leader later again if needed.

For easier understanding, their algorithm is presented beside.

In the beginning of each round in the scatternet formation, all leaders call the procedure MAIN, where a leader has a probability p (1/3 <p < 2/3) to call the SEEK procedure to acquire a slave, or call SCAN procedure to become a scan device otherwise.

Eventually after each round there are some n devices being connected between those executing SEEK and SCAN procedures through CONNECTED procedure by those

The algorithm:

```
Main(leader u)
x a random number in [0, 1)
if x < p (1/3 < p < 2/3)
  then Seek(u)
else if S(u) = ∅ ;
  then Scan(u)
   else v ←an unshared slave of u
Scan(v)

Seek(u)
u performs INQUIRY
if a slave v is found
  then u connects to slave v by PAGE
//S(u) ← S(u) ∪ {v}
Connected(u; v)

Scan(v)
v performs INQUIRY SCAN
if v is contacted by a master u
  then v waits for u in PAGE SCAN

Connected(leader u; slave v)
if v is a leader
then // v was an isolated leader
if jS(u)j < k
  then v retires
  else u retires
  y ← an unshared slave of u
  Move(fyg ; u; v)
else w ← the other master of v
  w retires
```

leaders. In the CONNECTED procedure, if two devices belong to two piconets, then these two piconets will try to merge into a single piconet through MERGE procedure if the number of slaves in those two piconets is not bigger then the maximum predefined. If the number of slaves in those two piconets is equal or bigger than the maximum predefined, some of the slaves in one piconet will move into other piconet through MIGRATE and/or MOVE procedures. These procedures will be repeated until all the nodes are connected.

```
switch
  case |S(u) ∪ S(w)| + 1 < k :
     Merge(u, v, w); return
  case |S(u)| = 1 :
     Move({u}, nil, w)
     v disconnects from w ; return
 case |S(u) ∪ S(w)| + 1 = k :
    u retires
    y ← an unshared slave of u
    Merge(u, v, w)
    Move({y}, u, v)
    v becomes a leader ; return
 case default :
    Migrate(u, v, w); return

Merge(master u, slave v, master w)
 v disconnects from w
 Move(S(w) \ v, w, u)
 Move({w}, nil, u)

Migrate(master u, slave v, master w)
 i ← min(k - |S(w)|, |S(u)|- 2)
 // i is the number of slaves to migrate

Move(set Z, master u, master w)
devices in Z disconnect from u
devices in Z wait for w in PAGE SCAN
w connects to devices in Z by PAGE
```

Several mathematical properties can be derived from the algorithm and they lead to the results of low time and message complexities. The message complexity O(n) can be seen as approaching the lowest bound, as each device in the scatternet formation process will send at least one message. The time complexity O(log n) (which physically means the number of rounds of the execution of the "MAIN") can be intuitively understood, that it is lower than linear complexity O(n) is just because there may be more than one node to be connected to a piconet through the MERGE or MOVE procedures of each round. Other two advantages of this algorithm are:

(1). The number of the piconets formed in a scatternet is at most (n-2)/(k-1) +1, which is asymptotically approaching the optimal number (n-1)/k, where k is the maximum number of the slaves allowed in a piconet. E.g., when n = 100 and k = 7, the concerned algorithm ends up with 17 piconets while the lower bound is 100/7 = 15.

(2). The highest degree of any slave node is 2, which certainly reduces or avoids the possible bottleneck problem in message transmission through an individual device (that normally will be a bridge node).

2.3 Performance Aspects of Bluetooth Scatternet Formation

This article [7] is about the *performance implications* of forming scatternets from piconets. As the Bluetooth specifications give a network manager quite flexible choices in forming a scatternet, the number of possible formations can be huge within a physical proximity. Obviously, different formations may bring different performance of the corresponding scatternets after being formed. The impacts of the formations or the topologies on the performance of a scatternt are thus fundamental or even critical to the effectiveness or efficiency of that scatternet. However, as stated in [7], it is hard to optimize the formation and thus the performance through analytical approach due to the complexity implied by the number of the possible formations. It is not easy either to use numerical optimization through simulation.

The authors thus instead used in [7] statistical approach to study the relationship between the scatternet design rules and its performance parameters. In their approach they firstly developed some simple heuristic algorithms to generate scatternets with desired parameters such as the number of piconets and links. Then they perturbed those algorithms and run them for large number of times to get statistically representative samples of all possible and legal scatternets. Then they used their analytical Bluetooth link model and a simple traffic model to simulate the resulting scatenets and evaluate their performance in terms of throughput, and finally determined the relationship between scatternet formation rules and performance.

With their link model, their statistic results show, e.g. in the case of the forward package length being five slots, that the performance in terms of the amount of *Forward Allocation* and *Forward Available User Rates* decline with the increase of the package length in the reverse direction, but the amount of *Reverse Allocation and Reverse Available User Rates* increase. The bridging overhead declines with the increase of the package length in the reverse direction (refer to Fig. 2.1 below).

Their experiments also support an intuitive expectation that the system performance is closely affected by the number of links and the overall bridging overheads. Their experimental results can be seen from Fig. 2.2 below.

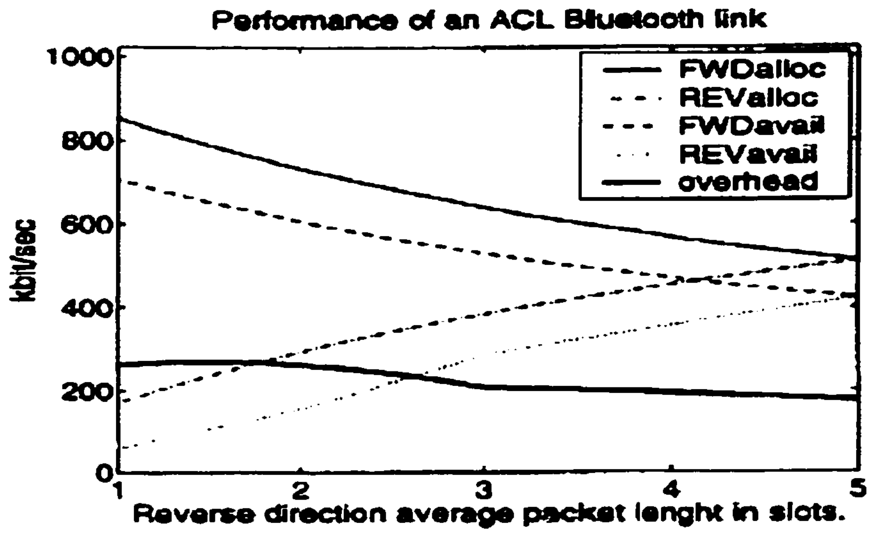

Fig. 2.1 Capacity of an asymmetric Bluetooth link based on the authors link model: amount of allocations, available user rates in the forward and reverse direction, and the overhead.

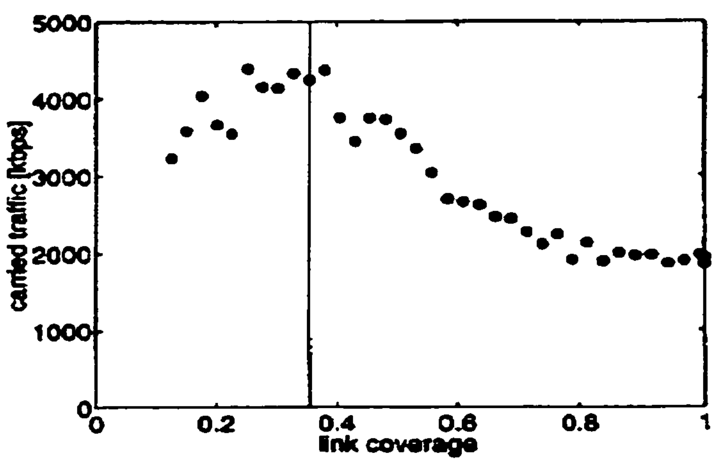

Fig. 2.2 Effects of changing number of links

My comment on this paper is that the authors mainly gave their experimental results rather than their models that were not elaborated in the paper. This makes it a bit hard for readers to grasp the concepts or the notations used and rationale behind their models and algorithms.

2.4 Handoff Support for Mobility with IP over Bluetooth

The previous papers are dealing with issues related to scatternets with static Bluetooth nodes, i.e. scatternets where the nodes do not change their position in the network. This paper [1] is focused on the handoff support issue when Bluetooth nodes are roaming among different access points, called as base stations. Those base stations and other suggested facilities like Bluetooth Gateways, Agents, Cellular IP Routers are organized into a proposed reference network -- the *Bluetooth Public Access* (**BLUEPAC**) network as conceived by the authors. Each base station may have more than one Bluetooth transceiver (BT) that connects to the individual Bluetooth nodes wirelessly.

Under the regime of the reference network, a Bluetooth node can get into or leave the network or roam among different base stations of the network arbitrarily. Accordingly there must be some solutions to support such mobility, as the existing Bluetooth LAN profile did not take mobility into consideration. One previous proposal that had been done by some authors of this paper by adapting *mobile IP* and *Cellular IP* on the OSI layer 3 [24]. This paper is dealing with the handoff issue on the OSI layer 2.

For any handoff problem, there are two main aspects: the mechanism or how the handoff shall be carried out, and the performance or efficiency of such handoff approach.

The mechanism of the handoff is based on a suggested *IP Adaptation Layer* that is to be inserted between the IP and the L2CAP layers of the Bluetooth stack. Such adaptation layer shall hide the information detail from the IP layer. The adaptation layer has two work versions corresponding to incoming node and to base station respectively.

The main idea of the version for incoming node is that, when a (new) Bluetooth device is entering the BLUEPAC network, the adaptation layer gets into a "**Discovery**" state and acts as a master to **INQUIRY** and **PAGE** the possible Bluetooth units of a base station to establish a connection to that base station. After that, the adaptation layer enters the "Configuration" state and then initiates a master/slave status swap, the base station changes to be a Master while the incoming node becoming its Slave. This swap is necessary because the performance of the base station will be greatly degraded later on if it remains to act as a slave to number of connected mobile nodes as its masters. This is because, as we know, if a slave wants to transmit data, it must be waiting until its master informs it to do so. Now, after finishing the configuration, the adaptation layer enters the "Connected" state and the incoming node becomes a slave of the base station and starts to work normally.

The version to base station is basically to maintain a reliable connection to the mobile Bluetooth devices, and the main states are "Configuration" and "Connected", its work mechanism and state transfer situations are understandably corresponding to those in the version for the incoming nodes as above.

To verify the proposed protocol with no real Bluetooth hardware available at that time, the authors used an **emulation** system to perform their tests. The emulation system was implemented in Specification and Description Language SDL/PR 88 and run on Linux PCs, and data communication was done via an Ethernet connection in lieu of radio media. They claim that their tests over such emulation system show that all goals set up in their conceived protocol have been achieved.

For the performance analysis of the proposed handoff protocol, the authors used a statistical approach.

Within the IP adaptation layer, the analysis was carried out principally through a splitting of the handoff process into three phases, *line loss* detection phase that is done by the link supervision timer, and *inquiring phase* and *paging phase* to next base station. The *link supervision timer* has a default value of 20 seconds, which is quite long from a user point of view. However, the authors did not discuss it further and left it open. They ignored at this time the *paging phase* because the page action can be assumed to follow the success of the inquiring phase immediately. Thus their analysis was focused on the inquiring phase. They also looked into the inquiring duration with the *scan interval* "T_{SI}" and the backoff time "T_{maxRB}" of the base station in responding to the mobile node's inquiring. The response time was assumed to be uniformly distributed on T_{SI} and T_{maxRB}.

Their analysis results show that the *inquiring time* could be within 4 seconds for the case of a sole base station in the region with probability of 90%, and 2 seconds for the case of three base station in region with the same probability. In general, more base stations in region will certainly increase the possibility to get reconnected in an acceptable time range (e.g. 2 seconds).

In reality, the duration of a *link breakage* due to a handoff may be far longer than that encountered during the inquiring procedure within the adaptation layer as studied. In addition, more time loss is happening on the upper TCP layer. TCP has an exponential **backoff** algorithm for link reconnection and data retransmission. This backoff time could be up to 1 minute if the TCP attempted a link reconnection number of times and finally succeeded a link reconnection and data retransmission. Such kind of a long time link breakage is intolerable. To overcome this problem, the authors discussed two approaches proposed by other researches: (a) semi-soft handoff scheme and (b) smooth handoff scheme. However, neither one of them can be adopted directly to solve the problem. The performance improvement and optimization of the handoff issue needs to be further studied, as well as the ideal link supervision time.

2.5 Bluetooth As A Personal Area Networking Enabler

As noted before, the main intended application of the Bluetooth technology is for the *personal area network* (PAN). Such PAN in Bluetooth terminology can be a piconet or a scatternet, while communications among number of PANs will be definitely in a scatternet form. A critical issue in forming such PAN system will be the determination of the gateway nodes between different piconets to form the scatternet, and the mechanisms for such gateways to provide robust, reliable and efficient information routing and transmission tasks. However, the current version of the Bluetooth technology did not specify the scheduling mechanism strictly for such gateway nodes. In the paper *An inabler for personal area networking* [4], the authors intended to fill this gap by giving an overview on their proposed scheduling algorithms and mechanisms and by presenting further a group of algorithms named as "Rendezvous point" for such purposes. .

The proposed mechanisms are based on the observation of the characteristics of the Bluetooth ad-hoc network PAN, namely, its distributed operation, dynamic topology due to its node mobility, its fluctuating link capacity, and its low powered devices. Among all of the concerned, the *interpiconet* nodes, also called *bridges* in other papers, will play a crucial rule in routing or forwarding packets to achieve highest possible throughput, or minimum delay, or least power consumption. The focus of this paper is on the design of proper algorithms to approach such ultimate **interpiconet scheduling** (**IPS**) goals as well as **intrapiconet scheduling** (**IRPS**) goals.

Two main insights were then gained: from protocol stack point of view, the routing/scheduling should be performed on the Bluetooth network layer below the IP layer, rather then the IP layer itself as could be intuitively supposed. Indeed within the SIG PAN work group, a "*Bluetooth Network Encapsulation Protocol*" (**BNEP**) had been proposed that shall reside between the IP layer and the L2CAP layer. The purpose is to enhance the packet forwarding efficiency and flexibility. Other insight is that the scheduling algorithm and the scatternet formation algorithm should be considered and implemented together so as to enhance the throughput further. This is because the routing/scheduling efficiency will be largely depending on the scatternet topology. A solution shall be then to assume an investigation of the possible alternation or even an optimal reformation of the current topology before a scheduling to be performed, due to the mobility of each node. An optimal topology at one time may be worst in another instant. It follows that the algorithm for determining the scatternet formation should principally focus on the proper presence of switches of an interpiconet node (the gateway node) in each of the piconets belonging to, and the duration of packet (data) exchange between the gateway and the Master of each piconet. Such kind of presence time point is the so called "*Rendezvous Point* " (RP), and the duration of packet (data) exchange being called a "*Rendezvous Window*" (RW).

Based on these studies, the overall functional architecture of the scatternet scheduling is on how an *interpiconet scheduling function* (IPSF) would interact with other functions in a Bluetooth unit. Such interactive functional relationship includes:

The coordination between peer IPSFs;

the response and possibly negotiations of IPSF to an application's requirements or constraints;

the interaction between the IPSF and the IRPS; and

the coordination between the IPSF and the Link Manager.

Such functional architecture gives various ways to do the IPS, while the main criteria in implementing the IPS are the RP and RW as mentioned before. Depending on how these criteria are to be approached, the algorithms to realize the IPS can be in different categories, and the paper suggested a family of the already mentioned "*Rendezvous Point* " algorithms. This family include:

Honouring *Periodic Static Window* (HPSW), where units always honour RPs, the RP being periodical and the RW being static;

Honouring Periodic Dynamic Window (HPDW), where units always honour RPs, the RP being periodical while the RW being dynamic to adapt to both topology and traffic changes;

Honouring Random Static Window (HRSW), where units always honour RPs, the RP being spread out randomly, and the RW being static;

Master-Honouring Dynamic Window (MHDW), where only Master unit always honour RPs, the RP being either periodically or randomly spread, while the RW being dynamic so that they can adapt to both topology and traffic changes.

The authors finally presented a perspective that the Bluetooth scattenet based PAN would be the first realistic ad-hoc wireless network and it would be one of the most promising technologies in changing our daily life and information transfer together with the 3G mobile phones.

Conclusions

In summary, Bluetooth technology is intended to provide a short range wireless connectivity with low power consumptions, especially applicable to ad-hoc formation of personal area networks. Such a wireless personal area network (WPAN) can connect various instruments like laptop PCs, PDAs, phones, appliance controllers, and similar devices together and transfer data among them wirelessly, thus improving data transfer efficiency, convenience, and reliability.

From the evolution point of view, Bluetooth technology is based on and is developed from radio and silicon technologies and computer data communication protocol principles. This report reviews the fundamental concepts in the Bluetooth technology, its relation to other technologies, and additionally gives broader background of evolution of networks, from earlier networks to later wireless and further ad hoc networks and their development. Through this way we can see logically the birth of the motivation for the Bluetooth on one hand, and also the technical possibilities for its realization on the other hand.

This systematic and sophisticated Bluetooth proposal, which allows creation and solution to the WPAN is no doubt prominent. The final realization of the targeted goals of the Bluetooth can be viewed to lead to another leap in office automation, household furnishing, information transferring and searching, multimedia communication, and ultimately can affect people's daily life styles with its various applications.

From a market point of view, due to many predictions of wide inclusion of the Bluetooth chips in many devices and to the technological possibilities it offers, it has created great expectations or even speculations in industrial and business domains. Various predictions have been done about the delivering of the Bluetooth units in the near future.

For instance, according to a study of a high-tech market research firm "In-Stat Group" (www.instat.com) in April 2001, Bluetooth-enabled equipment shipments will soar to 955 million units in the year of 2005, a 360% growth rate annually, and the revenue for Bluetooth radio and baseband silicon will rise to $4.4 billion in 2005 too, in spite of the economic slowdown. Another study was done by Allied Business Intelligence (ABI) who projected the Bluetooth chipset shipment will increase from 11.2 million in 2001 to 33.8 million in 2002. And by 2007 the Bluetooth chips revenue will reach $2.54 billion (http://www.alliedworld.com/servlets/CatalogView?category=CA03).

The above studies have big divergence though, and as we can see today those predictions were somewhat over-optimistic especially for these recent two years. However, they still give us an impression about the market perspectives of the Bluetooth technology.

From a long term point of view, people expect the known "market pull and technology push" phenomenon to happen to the Bluetooth technology again. This phenomenon says that market demand induces development of new technology or solutions, and this development will in turn further stimulate the market demand because of its price reductions and technical maturing, this will then pull the technology development further and cyclically. This phenomenon has been proved on the PC or Internet utilizations. In this view, the above predictions may not necessarily be too optimistic, perhaps it even shall turn out to be less than optimistic. Considered only 10% of the population worldwide -- let along a big number of institutional users -- and each user uses one piece of the Bluetooth chip/card, the chips demanded shall be up to a number of billion. Sooner or later, individuals will experience the use of Bluetooth or similarly enabled devices to be as common as is the use of wrist watches today.

Nonetheless, technical and business development in the utilization of the Bluetooth in the recent two years slowed down to some degree. The reasons are various, e.g. the overall economical sluggishness, and the challenge to reach the targeted unit price (of $5) for a Bluetooth chip. Another main factor probably shall be the competition from the IEEE 802.11 standard as mentioned in the introduction part. Though IEEE802.11 and Bluetooth have some major differences, they do have some overlaps in either technology or applications.

The situation has changed and improved significantly recently. In the end of March this year (2002), the IEEE Standards Association (IEEE-SA) announced the approval of the **IEEE 802.15** standard. This standard is based on, or adapted and copied from a portion of the Bluetooth specifications and thus fully compatible with the Bluetooth technology. There are some major changes and improvements in the IEEE 802.15 standard in comparison to the Bluetooth as well. For instance, a high-level behavioral ITU-T Z.100 *specification and description language* (**SDL**) model for an integrated Bluetooth MAC Sub layer. This SDL model offers an extensive overview of a significant portion of the Bluetooth protocols e.g., the LMP, L2CAP, and the HCI (refer to http://standards.ieee.org/announcements/802151app.html).

The confluence of Bluetooth and IEEE standards does not mean the disappearance of the Bluetooth. Reversely, it opens a bigger door to the application of the Bluetooth technology, as addressed by a leader of the IEEE 802.15 Working Group, quote: "The new standard gives the Bluetooth spec greater validity and support in the market and is an additional resource for those who implement Bluetooth devices."

The perspectives of the Bluetooth technology and its applications stated above fairly justify the study and the review on this technology as a project for a part of the Master program requirements.

From a technological or knowledge domain point of view, Bluetooth is a complex system across number of knowledge domains, from hardware to software, from radio to digital techniques, from wired to wireless networks, from protocol design to component organizations, from logic considerations to application implementations, etc. The Bluetooth Technology (Core) Specifications are more than one thousand pages long, and the Profile contains about 500 pages, plus a great number of other references. It demanded a lot of time to read them, digest them and understand them, and then combine them and organize them together to produce this report.

Through the preparation of this report, I have got a further understanding and review on those individual courses or topics learnt previously and thus gained a better understanding of their mutual relationships or supports, plus a new knowledge domain to me specifically related to Bluetooth technology.

Through the writing of this report, I have got further exercise in literature and information searching, knowledge absorptions and excerptions, not only general understandings but also frequently comprehensions of some aspects of the related topics, and technical writings.

Thus, from study point of view, the preparation and writing of this report is a comprehensive knowledge command stage. It exposed me to a broad and extensive knowledge field, and exercised me greatly in knowledge absorption and technical writings. It is a hard work stage to a student, but more importantly it is beneficial and fruitful to a student, and that exactly is the goal and purpose of the major report stage for a student to fulfill his Master program.

References

[1] Baatz, Simon; Frank, Matthias; Goepffarth, Rolf; Kassatkine, Dmitri; Martini, Peter; Schetelig, Markus; Vilavaara, Asko, Handoff support for mobility with IP over Bluetooth, Proceedings of the 25th Annual Conference On Local Computers Network (LCN'00), Tampa, Florida, US, November 2000, IEEE

[2] P. Bhagwat and S. Rao, On the characterization of Bluetooth Scatternet Topologies. Tech. Report, Dept. of CS, Univ. of Maryland, USA. 2001.

[3] P. Bhagwat, Bluetooth: Technology for short-range wireless applications IEEE Internet Computing, May/June 2001

[4] P. Johansson, M, Kazantzidis, R, Kapoor, M, Gerla, Bluetooth: An enabler for personal area networking, IEEE NETWORK September/October 2001

[5] K. Fleming, R. Hunter, J. Inouye, and J. Schiffer, Enabling Always On, Always Connected (AOAC) Computing with Bluetooth Technology. Intel Technology Journal, Number Q2, 7-14, 2000.

[6] C. Law and K.-Y. Siu, A Bluetooth Scatternet Formation Algorithm, The 1st IEEE Symp. on Ad-Hoc Wireless Networks (SAWN'01), San Antonio, Texas, USA, Nov. 25-29, 2001.

[7] G. Miklos, A. Racz, Z. Turanyi, A. Valko, and P. Johansson, Performance Aspects of Bluetooth Scatternet Formation. The 1st

IEEE-ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC '00), Boston, USA, August 11, 2000.

[8] B. Miller and C. Bisdikian, Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications, Prentice Hall, 2000.

[9] V. Park and M. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, IEEE INFOCOM'97, 1997.

[10] L. Ramachandran, M. Kapoor, A. Sarkar and A. Aggarwal, Clustering algorithms for wireless ad hoc networks, The 4th ACM Int. workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '00), pp54--63, 2000.

[11] J. Bray, C. Sturman: Bluetooth 1.1: Connect without cables, Publisher: Prentice Hall PTR. 2001

[12] Redl, Siegmund M, GSM and personal communications handbook, *Publisher* Boston: Artech House, c1998

[12] Roberts, James; Mocci, Ugo ; Virtamo, Jorma; Broadband network traffic : performance evaluation and design of broadband multiservice networks, *Publisher:* Berlin; New York : Springer, c1996

[13] Lee, William C. Y; Mobile communications design fundamentals, 2nd ed. *Publisher* New York: Wiley, c1993

[14] Christos J.P. Moschovitis ... et al. History of the Internet: a chronology, 1843 to the present, *Publisher* Santa Barbara, Calif. ABC-CLIO, c1999

[15] Black, Uyless D; Advanced Internet technologies, *Publisher* Upper Saddle River, NJ: Prentice Hall, 1999

[16] Groe, John B CDMA mobile radio design, *Publisher* Boston: Artech House, 2000

[17] Held, Gilbert; Data over wireless networks: Bluetooth, WAP, and wireless LANS,
Publisher New York, NY: McGraw-Hill, c2001

[18] Ya Xu, John Heidemann, Deborah Estrin, Geography-informed Energy Conservation for Ad Hoc routing, Proceedings of the seventh annual international conference on Mobile computing and networking, 2001, Rome, Italy

[19] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, Robert Morris, Capacity of Ad Hoc wireless networks, Proceedings of the seventh annual international conference on Mobile computing and networking, 2001, Rome, Italy

[20] Specification of The Bluetooth System, Core, Version 1.1, Bluetooth SIG, February 22, 2001 http://www.bluetooth.com/

[21] Specification of The Bluetooth System, Profiles, Version 1.1, Bluetooth SIG, February 22, 2001 http://www.bluetooth.com/

[22] Andy Dornan, The Essential Guide to Wireless Communications Applications, Publisher, Prentice Hall PTR, 2001

[23] Frank P. Coyle, Wireless Web, A manager's Guide, Publisher, Addison-Wesley, 2001.

[24] M. Albrecht, M. Frank, P. Martini, M. Schetelig, A. Vilavaara. A. Wenzel, "IP Services over Bluetooth: Leading the Way to a New Mobility", Proceedings of the *24th Conference on Local Computer Networks*, Lowell, MA, October 1999, pp. 2-11.

[25] IEEE 802.11 Specifications, IEEE 802.11 Work Group for Wireless Local Area Networks, 1997, 1999. http://grouper.ieee.org/groups/802/11/index.html

[26] HIPERLAN/2 Specifications, ETSI/BRAN (Broadband Radio Access Networks within ETSI), 1999 http://www.hiperlan2.com/

[27] TS 101 369 V7.1.0 (1999-11) Technical Specification, GSM 07.10 version 7.1.0 (1998), Terminal Equipment to Mobile Station, (TE-MS) multiplexer protocol, ETSI (European Telecommunications Standards Institute) SMG4, http://www.etsi.org

[28] Havel, V., A Remark on the Existence of Finite Graphs (Czech.), Casopis Pest. Mat., 80(1955), 477 - 480.

[29] Hakimi, S. L., On the Realization of a Set of Integers as Degrees of the Vertices of a Graph. J. SIAM Appl. Math., 10(1962), 496 - 506.

<<<<>>>>>