Routing in SWAN using Bluetooth

Brock, J.W.H.

Award date:
2001

9050

TU/e technische universiteit eindhoven

Section of Information and Communication Systems (ICS/CND)
Faculty of Electrical Engineering
ICS/CND 767

Master's Thesis

Routing in SWAN using Bluetooth

J.W.H. Brock

Coach:	ir. J.R. Schmidt (KPN Research)
Supervisor:	prof.ir. J. de Stigter
Examinator:	dr.ir. P.F.M. Smulders (TU/e, TTE/ECR)
Date:	July 2001

Author: J.W.H. Brock

Date: June 2001

ROUTING IN SWAN using BLUETOOTH

For internal use only at KPN 32740

Documenthistory

Version	Editor	Date	Explanation	Status
0.1	J.W.H. Brock	2000-12-20	First version	
0.2	J.W.H. Brock	2001-06-14	Full version	
0.3	J.W.H. Brock	2001-06-26	Concept version	
1.0	J.W.H. Brock	2001-07-05	Definite version	Finished

KPN Research

Informationsheet issued with Report 32740

Title:	Routing in SWAN using Bluetooth

Abstract:	Bluetooth is a radio technology which can be used to create self-configuring wireless ad-hoc networks. Information can be exchanged is such networks using in-between nodes to relay packets. A network formation algorithm is presented that creates a well connected network, providing network survivability. 40 devices per 400m^2 are needed for such a network. A routing method based on AODV is described for routing in Bluetooth networks. Due to the limited bandwidth, packet types and connections, scheduling is difficult and the throughput is very low. Hence Bluetooth can only be used to exchange small amounts of data between laptops, PDA's, cordless desktops, printers, etc. TCP/IP performs very bad in Bluetooth, because it is not developed for ad hoc networks and transmission control is done twice, both by Bluetooth and by TCP.

Author(s):	Ing. J.W.H. Brock
Reviewers:	Ir. J.R. Schmidt (KPN Research); Prof. Ir. J. de Stigter (TU/e); Dr. Ir. P.F.M. Smulders (TU/e)
Department:	EC
Project:	Eg 4G AIMS
Project manager:	Dr. Ir. A. Mawira
Project number:	16170
Commissioned by:	KPN Research
Date:	June 2001

Key Words:	Bluetooth, Routing, Connectivity, MANET, SWAN, Ad hoc, Mobile, 4G, Network

Mailing List:

KPN Telecom	W. van Blitterswijk (MOPR), R.W. Hendriks (MOPR)
KPN Research	J.R. Ortt, L.J.M. Nieuwenhuis, F.T.H. den Hartog, E.R. Fledderus, O.C. Mantel, P. Huizinga, P.A. Huising, P.A.J. Tilanus, L. Jorguseski, D. Groten, J.R. Schmidt, A.C.J. Verheijen, D. Matic, A. Mawira, C.P.M. van Hoesel, J.W.H. Brock (5x)

Preface

This report was written to finish my study Information Technology Sciences at the Eindhoven University of Technology. Information Technology Sciences is part of the faculty of Electrical Engineering.

The graduation project was executed at the Expertise Centre at KPN Research, located in Leidschendam, the Netherlands, and started in December 2000.

Besides my family and friends, there are three persons I would like to thank. The first two are my supervisors: Prof. J. de Stigter from the Eindhoven University of Technology and Hans Schmidt from KPN Research, who I would like to thank for their support, advise, and of course, the supervision.

The last person who I would like to thank is Stan van Hoesel, for his comments and ideas on network connectivity and survivability.

Jan-Willem Brock, June 2001

Contents

Management Summary

This report answers the question if Bluetooth can be used to provide an office network and how information can be routed through such a network.

Bluetooth is a free technology to connect wireless devices using a free frequency band. The expected chip price is circa five dollar, which will lead to high availability. The disadvantages of Bluetooth are the low speed and the limited range of ten meter.

An opportunity for KPN is the link of Bluetooth with UMTS and GPRS, which leads to increase in traffic, and a threat is the possibility that in the future small groups of mobile phones can communicate directly.

Before information can be routed through a Bluetooth network, first a scatternet must be created. An algorithm is proposed that creates an almost three-connected network and that provides network survivability. Unless mobility and fall-out, three alternative routes are available at any time. Twenty devices per $400m^2$ are needed to create this network. In low density areas, like cafeterias and halls, static devices can be placed to ensure connectivity.

When a scatternet exists any routing method for mobile ad hoc networks (MANET) can be used, however hierarchical and location based methods perform very bad. Four different MANET routing methods are compared and one (AODV) is chosen to be used in Bluetooth. Different route discovery methods are presented for use with Bluetooth. Besides the standard fastest path routing these are shortest path routing, power aware routing and link quality routing.

Routing through Bluetooth has a big disadvantage, due to scheduling demands and connection speeds, the throughput is very bad. Scheduling is this difficult because of the connection and packet types provided by Bluetooth. In some cases the overhead is more than 83 percent. When TCP/IP is to be run over Bluetooth the performance will drop even further. This is because TCP/IP is not developed for mobile ad hoc networks and because TCP tries to control the transmission, what is already done by the Bluetooth link layer. Due to the delay at each device the maximum number of devices in a network is limited to 150 when a maximum delay of 0.5 seconds is wanted.

The security provided by Bluetooth is not suitable for scatternets, this is because a secure connection is created between two neighbouring nodes. To provide a secure connection in a scatternet a method based on SSL is proposed, this is the method used on Internet. Due to the limited bandwidth implementing this method is not recommended.

A possible user profile consists of several computers, PDA's, laptops and printers in an office room, exchanging presentation sheets, business cards, synchronising en polling for e-mail. No large amounts of data, like files and speech, can be transported through this scatternet.

Bluetooth can not (yet) be seen as a threat for mobile telephony, because the throughput is too low and the delays are too long. The development of other wireless technologies like HIPERLAN are interesting to follow, because these can become a threat.

List of Abbreviations

3G	Third Generation Mobile Network (UMTS)
4G	Fourth Generation Mobile Network
ACL	Asynchronous ConnectionLess
AODV	Ad hoc On demand Distance Vector
ARQ	Automatic Retransmission Query
ATM	Asynchronous Transfer Mode
BC	BroadCast
BD	Bluetooth Device
B-ISDN	Broadband-Integrated Services Digital Network
BNEP	Bluetooth Network Encapsulation Protocol
BSIG	Bluetooth Special Interest Group
CAC	Channel Access Code
CSMA	Carrier Sense Multiple Access
DAC	Device Access Code
DES	Data Encryption Standard
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
FEC	Forward Error Correction
FHSS	Frequency Hop Spread Spectrum
GFSK	Gaussian Frequency Shift Keying
GPRS	General Packet Radio Service
GPS	Global Positioning System
HIPERLAN	High Performance Radio Local Area Network
HTTP	HyperText Transfer Protocol
IAC	Inquiry Access Code
IARP	Intrazone Routing Protocol
IERP	Interzone Routing Protocol
IMEP	Internet MANET Encapsulation Protocol
IP	Internet Protocol
IrDA	Infrared Data Association
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LANMAR	Landmark Routing Protocol
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MBS	Mobile Broadband Systems
NP	Non-Polynomial
OFDM	Orthogonal Frequency Digital Multiplexing
OSI	Open Systems Interconnection
PB	Packet Boundary flag
PDA	Personal Digital Assistant
PKI	Public Key Infrastructure
PPM	Pulse Position Modulation
QoS	Quality of Service
RSA	Crypto system named after the creators
RSSI	Received Signal Strength Indicator
RX	Receive
SCO	Synchronous Connection Oriented
SHA	Secure Hash Algorithm
SIG	Special Interest Group

SSL	Secure Sockets Layer
SWAN	Self-configuring Wireless Ad-hoc Network
TDD	Time Division Duplex
TORA	Temporally-Ordered Routing Algorithm
TTL	Time To Live
TTP	Trusted Third Party
TX	Transmit
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Access Protocol
WLAN	Wireless Local Area Network

1 Introduction

As Bluetooth is breaking through, the need for an ad hoc network built with Bluetooth units emerges. Because none of the current profiles supplies a method to exchange information between Bluetooth units by the use of other units, which act as routers, a routing method for Bluetooth networks will be proposed in this report.

The structure of this report is as follows:

In the following paragraph an introduction to Bluetooth is given. It discusses the basics and the possibilities of Bluetooth. In the second paragraph the topics researched are stated.

The second chapter encloses the formation of a network which exists of Bluetooth devices (scatternet). The first paragraph deals with the requirements of the Bluetooth units when ad hoc networking has to be supported. The second paragraph gives an overview of the connection set-up in Bluetooth. Paragraph three handles the formation of a scatternet and the functions that are needed. Also network survivability is dealt with in this paragraph.

In chapter three the routing through a Bluetooth network is discussed. Because no routing methods exist for Bluetooth, the routing methods for Mobile Ad hoc Networks (MANETs) are given in the first paragraph. This paragraph also discusses the differences between Bluetooth networks and standard MANETs. In the second paragraph a routing method for Bluetooth is proposed and an implementation is given. A SWAN (Self-configuring Wireless Ad hoc Network) is a type of MANET.

Chapter four deals with the Bluetooth usage models, including the influence of routing and network properties on the traffic expectations. Also user profiles and Bluetooth applications will be discussed.

The fifth chapter discusses the properties of the proposed routing method. These properties include: scheduling, security, scalability, robustness and billing. The properties are each dealt with in separate paragraphs. TCP/IP is dealt with in the scheduling paragraph.

The last chapter gives the conclusions and recommendations about the use of Bluetooth for building networks and about the routing methods to use it. Also the state of the Bluetooth technology is given.

1.1 Introduction to Bluetooth

Bluetooth is a new technology for short range wireless communication between mobile devices. These devices include laptops, PDA's, mobile phones headsets and many more. Because Bluetooth was originally developed for single hop connections, most current applications involve communication between two devices, like a mobile phone and a headset or a laptop and a PDA. But Bluetooth also supports ad hoc networks of devices. The most simple form of a network is a piconet, a piconet consists of one master and up to seven active slaves. Any Bluetooth device (BD) can be a master or slave. A BD can participate in more than one piconet at any time, but it can be only master in one. A BD that participates in multiple piconets can serve as a bridge, thus allowing the piconets to form a larger network. A set of piconets that are all interconnected by such bridges is called a scatternet. An example of a scatternet is shown in figure1.

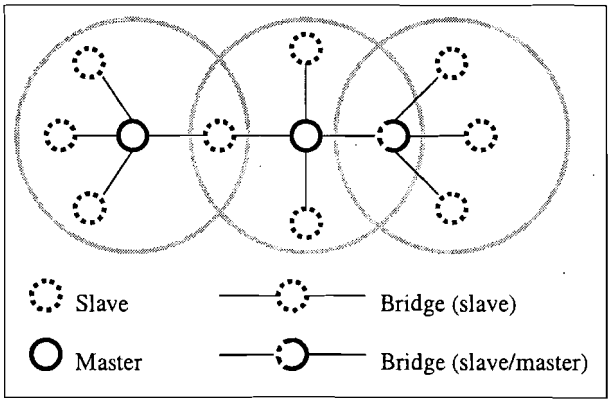

Figure 1: Three piconets in a scatternet.

Since a BD can transmit or receive in only one piconet at a time, the bridging BD's must switch between piconets on a time division basis. A slave synchronises with its masters hopping sequence, which is derived from the masters clock. Time is lost, due to the need to synchronise the radio connection from one piconet to another and due to the signalling which takes place before information can be exchanged.

In Bluetooth a slave is only allowed to send information to a master when it is addressed by the master in the preceding slot, i.e. when it is polled.

The literature on Bluetooth is scarce, because Bluetooth was started in 1998. One of the first articles is [1]. It gives a general overview of Bluetooth. Most of the information is found in the Bluetooth specification [19]. However on many points the specification is not clear and a new version is worked on. This second version is at this stage only available for members of the SIG.

Bluetooth operates in the 2.4 GHz ISM band, this band can be used unlicensed world-wide. The disadvantage is that the transmission power is limited, this however decreases the costs to use and manufacture it. Therefore Bluetooth is very interesting. Because of the growth of the use of mobile computers, mobile phones and PDA's, it is expected that, when Bluetooth modules are integrated in these devices, Bluetooth will be very useful.

Besides the core specification [19], a number of profiles are specified [20]. A profile describes an interface that a manufacturer must comply with, if it is desired that the device can work with devices of other manufacturers for the profile. A profile includes a scope in which the profile can be used, an overview of the dependencies with other profiles and with the Bluetooth protocols, an overview of the applications which can use it and how, and a description of the functions of the profile.

These profiles include, among others, a headset profile, a cordless phone profile and a LAN access profile. The LAN access profile only tunnels the information (IP packets for example), over an emulated serial connection, to a LAN access point which places them on the LAN. No network between multiple BD's can be formed with this profile.

In short the properties of Bluetooth are:
- 2.45 GHz radio band.
- FHSS radio technology.
- Point-to-point and point-to-multipoint connections.
- Range of 10m.
- Speech and data up to 1Mbps.
- A cheap and multipurpose chip.

The current alternatives for Bluetooth are added as a supplement in appendix 1. Also a comparison of Bluetooth and its competitors is given.

1.2 Scenario

In this paragraph a scenario is given where Bluetooth devices in an office environment form an ad hoc network. First an example is given in the figure below.

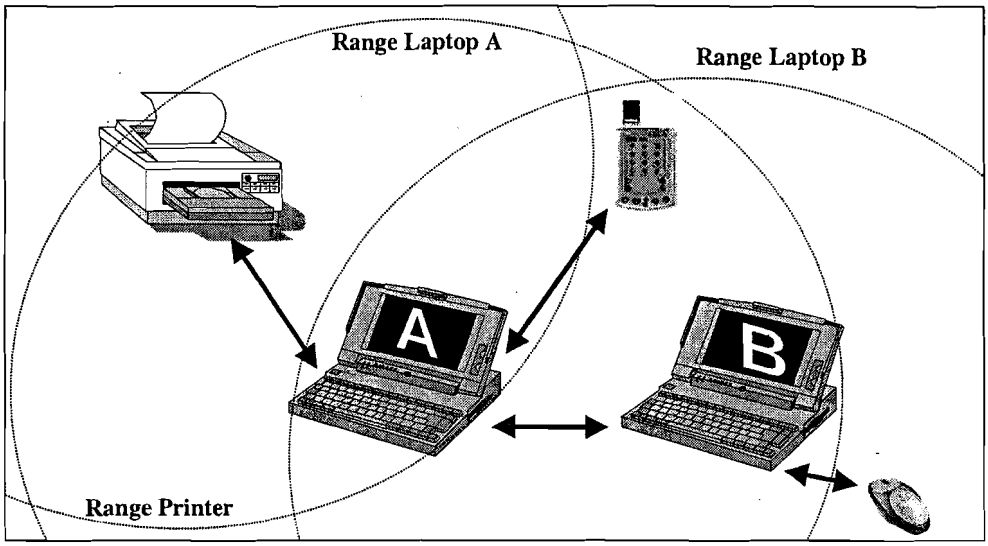

Figure 2: A possible scenario.

All devices in the above figure have a Bluetooth module and their range is given. Laptop B is unable to communicate with the printer, so if it wants to print it is done via laptop A, which acts as relay.

Other scenarios include accessing a LAN via a LAP (LAN Access Point), which is not in range, by the use of other Bluetooth devices as relay, or forming an autonomous network of mobile computers.

More global, Bluetooth can be used as a low speed office network; connecting office equipment like computers, electronic agenda's and printers with each other and with a LAN. In such an environment the mobility is not very high and devices stay at the same place for a considerable period.

1.3 Research

The subject of research is formulated in the following question:

Is it possible to use Bluetooth for exchanging information between mobile devices in an ad hoc network in an office environment and how is this information routed through this network?

Before this question can be answered first other questions must be answered:

1. How is an ad hoc network of mobile Bluetooth devices created and maintained, how is network survivability provided and what considerations have to be made?
2. How is routing done in other ad hoc networks and how is the applicability of these routing algorithms for Bluetooth?
3. If routing through a Bluetooth network is possible, what is the best routing method for Bluetooth and what are the properties of this method? These properties concern:
   - Scalability
   - Security
   - Scheduling (congestion control)
   - Robustness
   - Billing methods

2 Creating A Bluetooth Network

This chapter deals with the building of a network (scatternet) with Bluetooth devices. In the first paragraph the requirements for a Bluetooth network are given. The second paragraph describes how two Bluetooth devices set up and maintain a connection. After that the goal of creating a scatternet is given in the third paragraph. This paragraph also describes the chosen method for the formation of a scatternet and design parameters used to achieve the goal. The last paragraph describes the implementation of the network formation method.

2.1 Requirements

There are several requirements when using Bluetooth devices in a network. A device must be able and wanting to be part of a network and, in a later stage, to route packets through this network. Therefore it has to be part of the specification, i.e. there has to be defined a bit that indicates that a device can be used as relay.

In an ideal situation a network is set up in the network layer of the OSI reference model, using the services and primitives of the link layer. The Bluetooth link layer exists of two parts, the Link Manager Protocol (LMP), which creates a connection, and the Logical Link Control Access Protocol (L2CAP), which controls the connection. However the services and primitives of the LMP and the L2CAP are not completely specified and implemented yet, making it impossible to implement a network layer protocol. Therefore the protocol is created in the application layer, using the Host Controller Interface (HCI). The HCI translates functions into LMP and Baseband primitives and is completely specified and implemented by the Bluetooth chip manufacturer. The HCI functions can be called using a library supplied by the manufacturer. The HCI functions needed for creating, maintaining and routing through Bluetooth networks are added as a supplement. The ideal and the actual protocol stack are shown in figure 3.

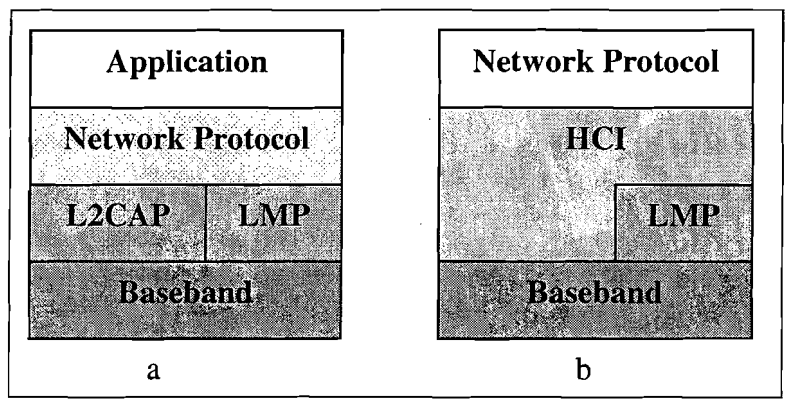

Figure 3: Protocol stacks: (a) ideal and (b) actual

2.2 Connection Set-up

This paragraph discusses the way two Bluetooth devices find each other and create and maintain a connection. In the last subparagraph an overview is given of the connection set-up.

2.2.1 Inquiry

The *inquiry* function enables a unit to discover units in range which are listening for inquiries, i.e. which are in the *inquiry scan* state. The result is the device address and the remote clock. The receiver scans for the inquiry access code (IAC) long enough to completely scan sixteen inquiry frequencies. The scan is performed at a single hop frequency, of which 32 are available. The phase of the frequency is changed every 1.28 seconds. The interval in which a unit scans is at most 2.56 seconds. The *inquiry* function uses two 10ms trains, which divine the 32 hop frequencies into two 16 hop groups. A single train must be repeated for al least 256 times before a new train is used. In order to collect all responses in an error-free environment, at least three train switches must take place. This results in a *Inquiry* which may last for 10.24 seconds in an error free environment. However in most cases an *Inquiry* takes 5.12 seconds [17] and when multiple devices are present, finding any device can be done faster.

2.2.2 Page

To connect to a unit the *page* procedure is used. The remote units must be in the *page scan* state to respond to the *page*. A unit in *page scan* listens for its own device access code (DAC) for the duration of the scan window. Outside this scan window the unit is free to communicate with other units. For two units to connect, only the Bluetooth address is necessary. But when the remote clock is known it speeds up the connection set-up. This is because the paging unit knows in which time interval and on which frequency the paged unit will listen for page messages. If the remote clock is not known, the paging unit has to try different periods and different frequencies. This results in a duration of maximal 7.68 seconds for the connection set-up. However, when the remote clock is known, the typical duration is 0.64 seconds [17].

The slave clock-offset is synchronised each time a packet is received from the master. This is done by comparing the exact RX timing of the received packet with the estimated RX timing. Only the Channel Access Code (CAC) is required for this.

2.2.3 Connection types

The Bluetooth specification defines two connection types over which information can be exchanged. The connection type must be chosen during connection set-up. The types are:

- Synchronous Connection Oriented (SCO)

  This symmetric link supplies two 64kbit/s speech channels (one for each direction), using two successive slots. The SCO link is not interesting for making networks, because all connections must be stopped when a BD enters page scan mode or when page messages are sent.

- Asynchronous ConnectionLess (ACL)

  This link is used for data communication between a master and one or all connected slaves. It is possible to use an error correction method (FEC), but of course this decreases the data rate. For a higher data rate it is possible to use three or five slots for a packet instead of the default one. ARQ is used to request retransmission of a packet.

Bluetooth specifies different packet types for different purposes. The packet types for a ACL connection are given in the table below:

Table 1: ACL Data packets

Packet	Description	Length (slots)	Information (bytes)	(2/3) FEC	CRC
DM1	Data Medium Packet	1	18	Yes	16 bit
DH1	Data High Packet	1	28	No	16 bit
DM3	Data Medium Packet	3	122	Yes	16 bit
DH3	Data High Packet	3	184	No	16 bit
DM5	Data Medium Packet	5	225	Yes	16 bit
DH5	Data High Packet	5	340	No	16 bit
AUX	DH1 packet without CRC	1	30	No	No

With a 2/3 FEC (Forward Error Control) five parity bits are added to every ten-bit slot. A CRC is a Cyclic Redundancy Check which in this case uses sixteen bits to check if the packet was sent successful. When the connection is poor the FEC should be enabled and the DMx packets should be used. If there is no distortion the FEC should be disabled and DHx packets should be used.

For a higher throughput it is more efficient to use DH5 packets. However, when more connections exist at the same time, it is not efficient to use DH5 packets. This is because there is no proportional division possible between the connections. In this case one or three slot length packets should be used.

Authentication and encryption can be used to provide a secure connection. Each Bluetooth device has private keys and a random number generator. Authentication and encryption properties can be set up during the connection set-up. More information about the security are given in chapter five.

2.2.4 Connection Modes

A master-slave connection can be in different modes, in most cases the modes are controlled by the piconet master. The different modes are listed below.

2.2.4.1 Hold Mode

In Hold mode a slave temporarily does not support ACL packets from the piconet master. This mode is used to free capacity on the master device to perform an inquiry or to attend another piconet. Prior to entering Hold mode, master and slave agree on the time duration. This duration may vary from 0.625ms to 40.9s.

2.2.4.2 Park Mode

Park mode is equal to the Hold mode, however the master must exit the park mode. Therefore the park mode duration is unlimited. A master can have more than seven slaves connected, but all other connections must be parked. To exit the park mode parked slaves listen for broadcast messages, after which they are allowed to sent a packet to the master. When a master has parked slaves it establishes a beacon channel for the communication with, and synchronisation of, parked slaves. The interval between the beacon slots can range from 0.625ms to 40.9s.

2.2.4.3 Sniff Mode

In Sniff mode the slaves duty cycle for listen activity is reduced. The master only polls the slave only after every Tsniff slots. Where the duration between two polls ranges from 0.625ms to 40.9s. This mode is used when there is not much information to be sent between the devices and the information exchange occurs regularly.

2.2.5 Master-Slave Switch

When a unit currently is slave in a piconet and wants to become the piconet master (and the current master agrees) a Master-Slave Switch is performed. This switch results in a reversal of the RX and TX timing. When the (old) master has other slaves attached, they are parked before the switch and after the switch they are reconnected to the new master. A Master-Slave Switch is combined with the accept connection command, in this way the unit which performed the inquiry can become slave of the connection.

2.2.6 Connection Overview

The complete set-up is shown in the figure below. The dashed arrows indicate that the procedures are optional and the dashed time-axe indicates that the procedures can take place at any time and in any order. However when a device is in hold or park mode it is not able to perform a role switch or to exchange data.

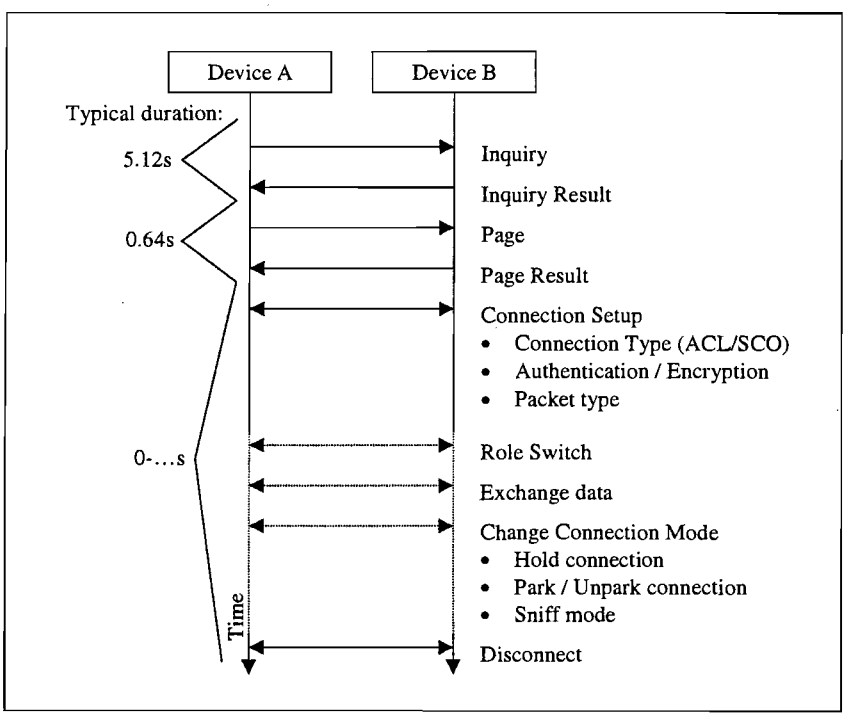

Figure 4: Connection set-up between two Bluetooth devices.

The situation above uses the mandatory paging procedure, which must be supported by all Bluetooth devices. Optional paging procedures are defined in the specification, which allow faster connection set-up. There are three optional paging schemes, but only one is defined yet. The main difference between this optional paging scheme and the mandatory scheme is the construction of the page train sent by the pager. In addition to transmitting in the even master slots, the pager is transmitting in the odd slots as well. This allows the slave to reduce the scan window and results in a factor two faster connection set-up.

Other optional paging methods could include reducing the used frequencies. This however makes the system less robust, because if there is disturbance on the chosen frequency a connection is not possible. In the mandatory paging procedure the system switches to another frequency and therefore eliminates the disturbance.

2.3 Scatternet Formation

In this paragraph the formation of a scatternet is discussed.

The network must exist before routing takes place. If the connection set-up is done when a route is needed, it results in a route discovery which takes the connection time times the number of hops. This is illustrated in figure 5, which is based on a typical inquiry plus connection set-up time of about five seconds. It is clear that this results in unacceptable delays.

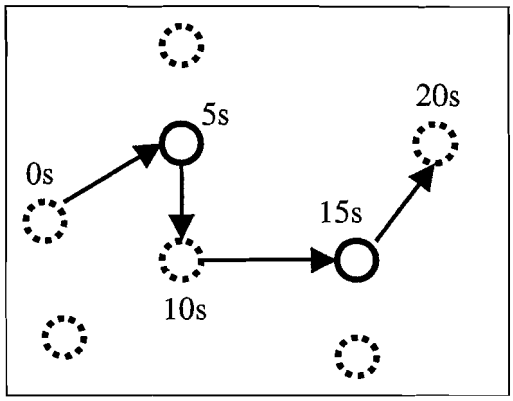

Figure 5: Time indication for route discovery

The goal of the formation is to form a scatternet which:

- Optimises throughput
- Includes the most active units
- Is self organising
- Handles mobility and fall-out
- Is robust

To optimise throughput multiple paths between units are desired. In extreme this is a network where all nodes are connected to all nodes in range. However, the Bluetooth specification limits the number of connections. A strongly connected Bluetooth network is shown in figure 6a. At the other end there is a network where only one path to each BD exists, this is shown in figure 6b.

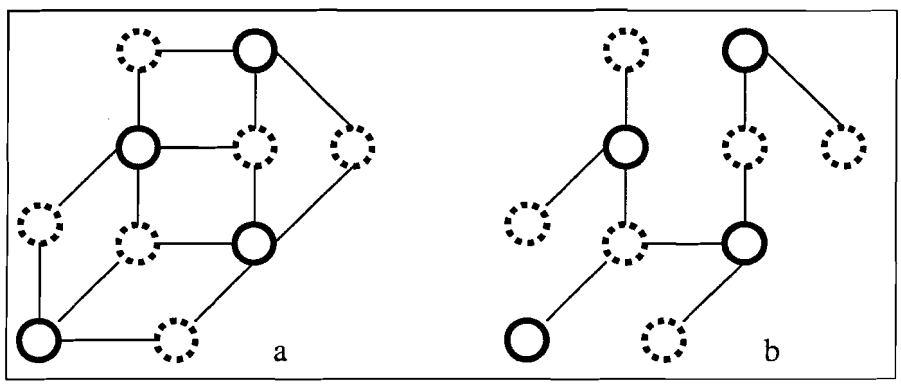

Figure 6: A strongly connected (a) and a single connected (b) scatternet.

When more paths exist the network is more robust. This is because such a network contains less critical units, i.e. units that divide the network when it falls out. The number of units participating in such a network and the area coverage are limited, because many units are in transmission range of each other.

Of course the optimum is somewhere in the middle, providing enough coverage of area, enables enough units to participate and still provides enough robustness to overcome fall-out and mobility.

The design parameters covering this problem are:

- The number of connections of each device. This contains both the number of masters connected to one slave and the number of slaves connected to one master.
- The bridging method. Should bridging devices be slaves connected with several masters, or should bridges be master in one piconet and slave in one or more other piconets? This second case forces the device to switch between a master and a slave role on a TDD basis.
- The number of different paths between devices, i.e. cycles in the network. This relates to the first design parameter and is discussed more thoroughly in paragraph 2.3.1.

Considering this problem, the first limitation is the number of active slaves that can be connected to a master. The maximum number of active slaves is seven. When there are more slaves connected to a master they have to be parked. When one slave wants to communicate, an other slave has to be parked. Therefore in this implementation a master is not allowed to have more than seven slaves, to improve throughput.

When a unit is slave in multiple piconets, it must synchronise with a master when it wants to communicate. This synchronisation is done by remembering the difference between the masters clock and the units own clock. Regular updates are necessary because the clocks can drift.

This synchronisation takes about one slot-time. A slave needs two slots per master, one for synchronisation and one the transfer of information. With this information the maximum number of masters per slave can be determined. This is three masters per slave, because three times two results in six slots, leaving one slot unused. This, however, is based on an ideal situation where all masters poll the slave in turn.

Because Bluetooth has a slotted MAC protocol, where a slave can send if it is addressed by the master. It is possible that, when a unit is slave in multiple piconets, the polls of the different masters, after which the slave is allowed to send, overlap. Resulting in a non-optimal connection. This is because the different piconets are not synchronised. The same problem arises when a unit is slave in one piconet and master in another.

Investigations have been done concerning the performance aspects of scatternet formation. In [5] a statistical experiment is described in which the optimal number of links is determined. This is measured in relation to the maximal throughput. The conclusions are that the number of units bridging between piconets has to be kept to a minimum and that the optimal link coverage is about one third.

2.3.1 Network Survivability

Because the time to set up a connection is large and because it is a network of mobile, ad hoc hosts a robust network is wanted. Such a network is not split when a device is removed. Creating and maintaining a network is called network survivability.

First the definition of a robust network must be determined. Several definitions are given below:

1. The most strong definition of a robust Bluetooth network is a network in which all devices can reach each other by more than one, node disjunct, routes. This is called a k-connected network (with k>1), i.e. when all devices can reach any other device by two node disjunct routes this is called a two-connected network. Node disjunct means that no two routes may include the same device. This is stronger than edge disjunct, where no two routes may include the same link. Node disjunction is chosen because in Bluetooth it is likely that when one link is broken another one will follow, because the device switches off or because it moves.

2. A less strong definition of a robust Bluetooth network, based on the same principle, is a network in which all devices (masters and slaves) that are connected to the network with more than one connection can reach each other in at least two ways.

This subset includes all masters that have more than one slave and all slaves that have more than one master, i.e. the bridging units. With this definition devices at the edge of the network are allowed to be single connected to the network.

3. Another definition can be that a given percentage of the network must remain connected when a certain device falls out or when a certain link is lost. With this method the goal can to maintain 95 percent of the network when any device fails.

4. A combination of above methods results in a definition of a robust Bluetooth network as a network where a percentage of the network is k-connected. Again, the goal for this percentage can be 95 or more percent.

For all above methods counts that the network itself does not check for connectivity. The design parameters like the number of connections and the master slave proportion are determined and when applied in all devices, a self organising robust ad hoc Bluetooth network will exist.

There are two ways in which the design parameters can be determined: by Random Graph Theory and by simulation. Both methods will be described below.

In Random Graph theory, where ad hoc networks are a practical part of, this is an already known problem, described as follows. Given a random set of points (nodes) in a geometric area, how many randomly chosen edges (connections to other nodes) should each node have to ensure k-connectivity.

Testing for two-connectivity can be a difficult process, as is shown in figure 7.

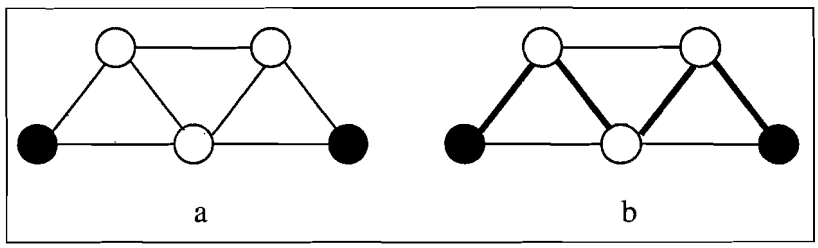

Figure 7: Two-connected nodes in a graph (a) with single connected path (b)

The two black nodes in figure 7a are two-connected, i.e. there exist two node-disjunct paths between the nodes. However an algorithm may find the path shown in figure 7b and conclude that the nodes are not two-connected.

Another difficulty is the occurrence of articulation points. An articulation point is shown in figure 8.

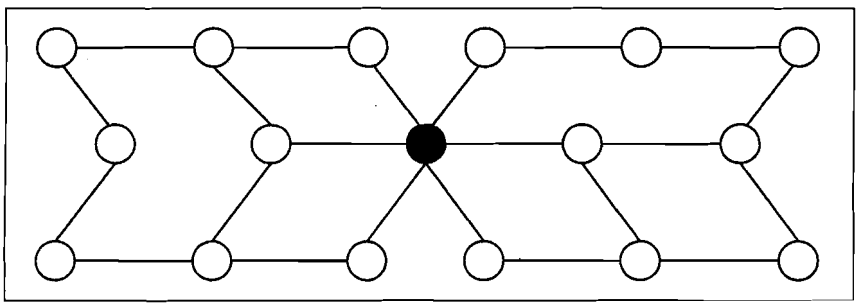

Figure 8: An articulation point in a graph

Unless all nodes have multiple connection with the graph, it is not two-connected. This situation must therefore be prevented. Which is difficult in an ad hoc network because there is no node with a network overview.

It is not possible to project the Bluetooth network survivability problem directly on this theory, because in Bluetooth exist two different types of nodes, namely masters and slaves. Both with an other maximum of possible connections. This increases the variance in the number of connections, what makes the problem more difficult.

Therefore the simulation is the best way to determine the design parameters to create a robust Bluetooth network.

The design parameters to be determined are the master slave proportion and the timing aspects, like the time after which a device must switch role. The master slave proportion depends on the number of connections masters should have to slaves which act as bridge, this is limited to seven, and the number of connections a slave should have to masters, which is limited to three.

To provide more certainty for connectivity, static nodes can be added in places where too little devices are available. Such places can include corridors, stairs, elevators and large open places like cafeterias. Devices will become cheap enough to use abundant.

This abundant use of devices is also done in army environments, where survivability is a main issue.

2.3.2 Scatternet Formation Algorithm

In consideration with the items discussed in the previous paragraph, a scatternet formation algorithm is given in this paragraph.

A scatternet is created in an ad hoc way. All devices contain the same functions and start as slave. Only slaves perform an inquiry, so the device that wants to connect to the network sacrifices its own power and time.

When no master is present or when no master is currently listening, the slave becomes a master itself. Now it enables the inquiry scan so other slaves are able to connect. When seven slaves are connected it disables the inquiry scan until a slave disconnects. If the devices moves itself and all connections are lost it will become slave again.

When the slave is able to find a master, it will stay slave and continue the search for other masters. When three masters are connected it will not perform inquiries anymore. To improve performance the number of inquiries can be decreased when the slave is part of a scatternet.

This method uses the ability to 'connect as'. Normally when a unit performs an inquiry it will become master of the connection, however it is possible for the requested unit to accept the connection as master, leaving the other unit slave.

When a slave performs an inquiry it connects with the first unit that responds and stops the inquiry. As described earlier, it becomes a master when no devices respond to the inquiry.

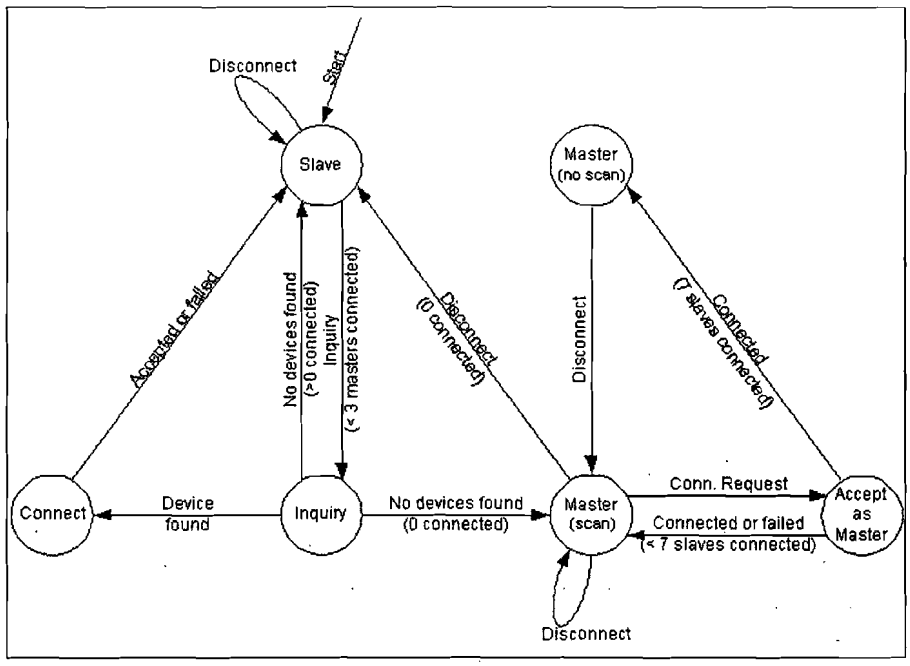

Figure 9: Scatternet formation Flow Diagram

Starting a network in an ad hoc way can take a lot of time when both devices start at the same time. When two units start as slave, searching for a master, they can not find each other. To prevent that both units switch at the same time (which is not likely), and still can not find each other, a random factor is included.

The *inquiry scan* period must be long enough for a device to be found, the inquiry to be completed and to cover the connection time. When the search period of one device overlaps the search period of another device, they can not find or connect.

With this proposed method, the properties of the network can be adjusted by timing the *inquiry* and the *inquiry scan*. When a unit is part of a network and it will often perform an *inquiry* (and connects to the device found) many cycles will be created.

The design parameters depend on the following Baseband characteristics:

- The time necessary to find and connect.
- The time necessary to synchronise.

Because this are Baseband properties, these are not part of the research and are therefore treated as given facts. Through technological development done by the manufactures of Bluetooth chips, these values are still decreasing.

The implemented values are:

- Max. seven slaves connected to one master.
- Max. 3 masters connected to one slave.
- Masters do not perform inquiries.
- A slave performs an inquiry one tenth of time, until it is filled up or until the networks stops changing. In this way there is a good possibility that two devices are able to connect. This means that the inquiries do not overlap.
- When no master is found within the inquiry time, the slave becomes master.

2.3.3 Simulating Scatternet formation

To test the proposed scatternet formation method, a simulation tool is written in which Bluetooth devices form an ad hoc network. All devices perform the same functions and have the same design parameters. They connect in an ad hoc way confirming the Bluetooth specification and the proposed algorithm.

The simulation is done for various densities, based on an estimated default value. This value is determined by estimating the number of Bluetooth devices in an office. The area of an office room is ca. $20m^2$ and in it are on average five Bluetooth devices (two laptops, two electronic agenda's and one printer or LAP (LAN Access Point)). The existence of a LAP does not exclude the need for a network with Bluetooth devices, it adds the possibility to connect to other networks like internet.

In the simulations the percentage of loose nodes were measured. A loose node is a node which is connected to the network with one connection or a node of which only one of its neighbours is connected to the network. This is illustrated in figure 10.

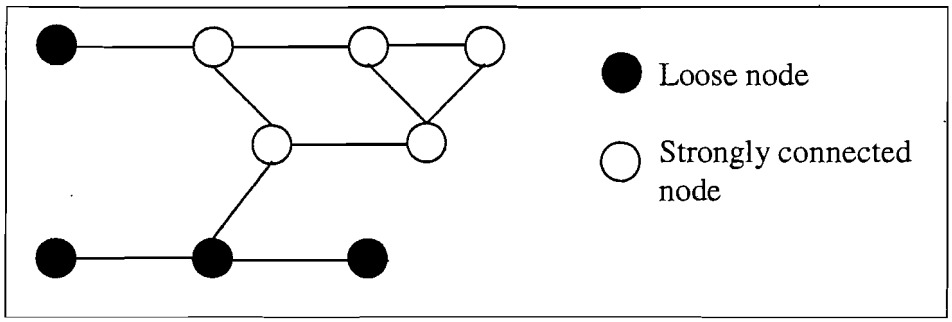

Figure 10: Loose nodes.

Through the simulations the following questions must be answered:

- What is the minimum number of devices necessary to form a robust network?
- What is the best algorithm for scatternet formation?

The simulations are done for different number of slaves per master and for both two and three masters per slave, because this can be interesting for developing scheduling algorithms. The Bluetooth properties are implemented in the simulation.

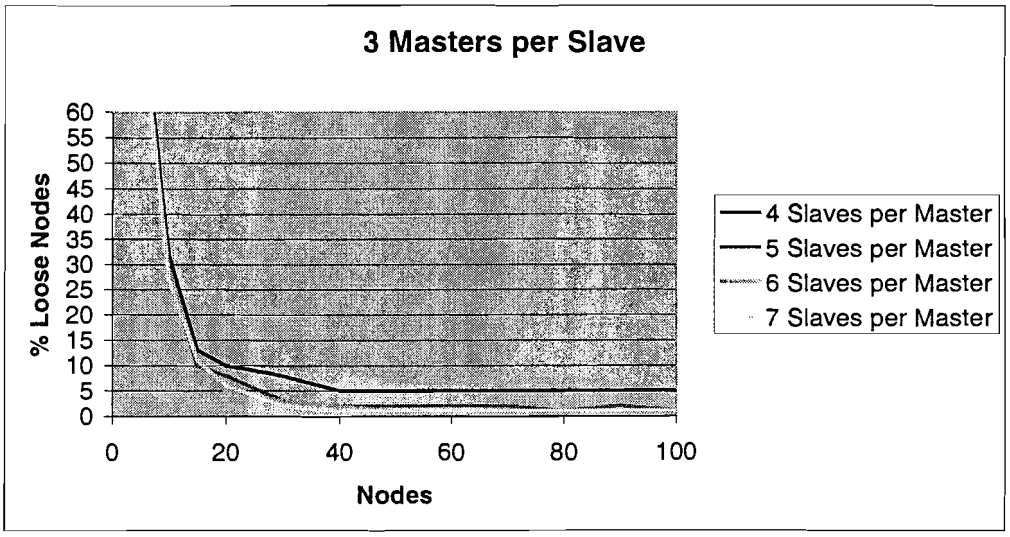

Figure 11: Scatternet connectivity for different densities

From this simulation the minimum number of active nodes can be found. Given the requirement that a network is well connected if only five percent of the nodes are loose, a

minimum of twenty active nodes is necessary. This is equal to fifteen nodes in range of the Bluetooth device.

A remarkable point is the limited influence of the maximum number of slaves a master can have. The simulations show that the master-slave ratio is limited to 5.6, when the maximum number of masters per slave is six or seven. Otherwise the master-slave ratio is inversely proportional to the ratio between the number of slaves per master and the number of masters per slave. This however, is in a static environment. When nodes are added or removed randomly the master slave ratio changes but the percentage of loose nodes remains very low. Again, this is not influenced by the maximum number of slaves per master.

To make the simulation more realistic nodes are added and deleted in a random Gaussian distributed manner. This adding and deleting nodes is more harmful than moving nodes and had no notable influence on the connectivity. The results of this simulation are shown in figure 12.

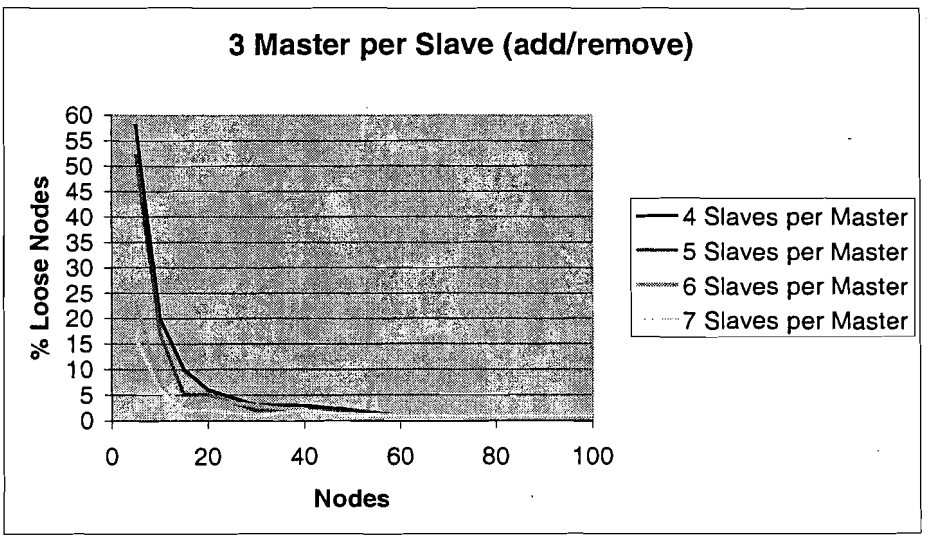

Figure 12: Connectivity with adding and deleting nodes

The above results are for the case where a slave can have three masters, however when the number of masters per slave is limited to two there is only a very small increase in loose nodes.

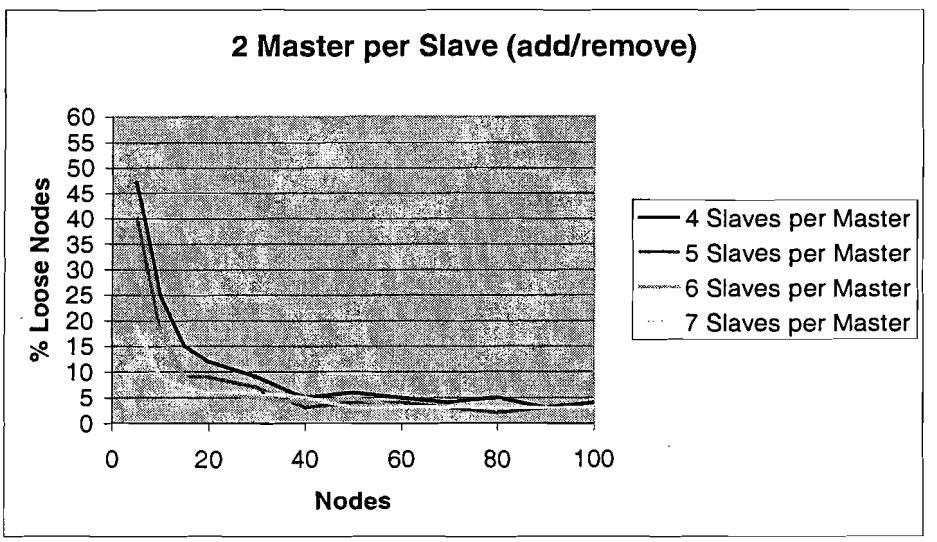

Figure 13: Scatternet connectivity with two masters per slave

In the simulation, the nodes that are near the borders cover less area for finding other nodes. This influences the simulation. However, in the used scenario, these borders also exist and therefore they make the simulation more realistic.

When enough nodes were available no network splits or situations as shown in figure 8 were detected.

The known limitations of the simulation method are:

- The simulation did not include any user profiles.
- No area scenarios, like walls and other objects that influence the transmission power (and thus the range), were included in the simulation. However, while using the devices it was noticed that the range was larger than specified and walls and other obstacles did not influence the connections.
- In real situations the nodes are not placed randomly. There will be nodes that are more important and therefore other nodes will gather around such nodes. Examples of such nodes are printers or access points.

2.3.4 Proving Scatternet Formation

Using article [26] it can be proven that a random graph is k-connected when all nodes have at least k connections.

A simulation has been done to determine the average number of connections a Bluetooth device has over time. The average has been taken because it is not important some devices have only one connection, because the goal is the main part to be strongly connected. The simulation was done using the optimal inquiry and connection time to find and connect to a random device in range. This resulted in the following figure:

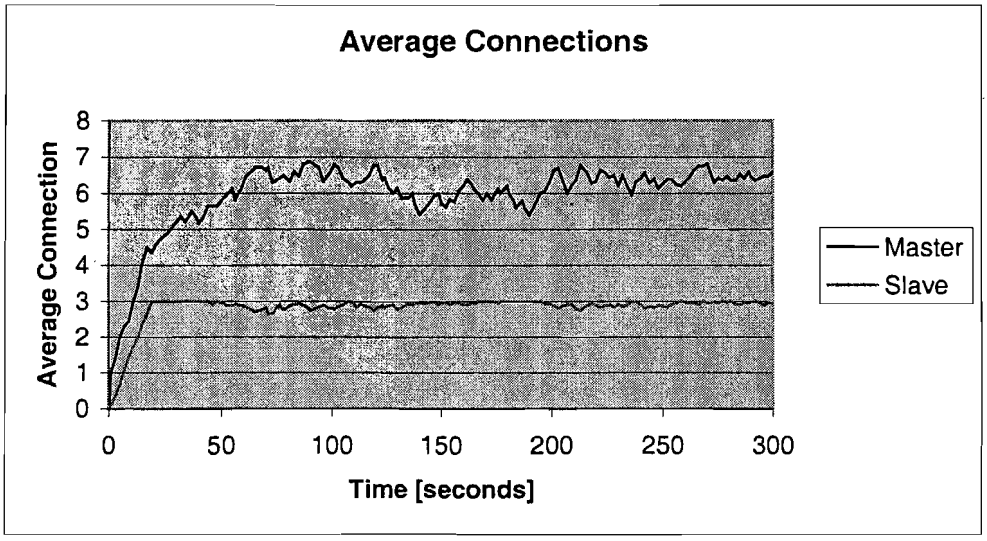

Figure 14: Average connections per Bluetooth device

As can be seen in the figure, theoretically a 3-connected network exists starting at 21 seconds.

2.3.5 Conclusions Scatternet Formation

At this point research question one has been answered.

Given the characteristics of Bluetooth a scatternet formation algorithm has been introduced that forms a scatternet which:

- Optimises throughput, by providing multiple possible paths between two devices and by letting only slaves perform inquiries.
- Includes the most active units, by not putting units in park mode and not performing master slave switches.
- Is self organising, because the goal of each unit is to form a network.
- Handles mobility and fall-out in a good way.
- Is robust, because the network is almost three-connected.

The advantages of the implemented scatternet formation method are:

- No nodes are placed in standby mode.
- Masters do not need to inquire.
- A strongly connected network is formed.
- No master-slave switch is needed.

The main disadvantage is:

- The dependence on the Bluetooth connection set-up. The maximum connection set-up is still about ten seconds in an error free environment, however the discovery of any device can be very fast.

3 Routing Through A Bluetooth Network

This chapter describes a routing method for Bluetooth networks. Before the routing can be done the connections between devices must be already formed. This is due to the long connection set up time and is explained in the previous chapter. The task of the router is to find the best path, if one exists, to another device. The best path can be chosen from different viewpoints, like:

- Shortest path (fewest number of devices)
- Fastest path (least traffic)
- Cheapest path (power or billing)
- QoS

A number of interesting routing protocols types are described below:

- **Source routing vs. hop-by-hop routing.** Basically there are two methods of routing through a network. The first uses datagrams (hop-by-hop routing) and the second uses a virtual circuit (source routing). For each datagram a new best route to the destination is determined, where on a virtual circuit all packets travel the same route. In unstable networks datagrams are preferred, because they overcome fall out and mobility. With a virtual circuit it is possible that halfway a session the traffic on one part of the route increases and therefore the whole connection has become slow.

Table 2: Comparison source- and hop-by-hop routing

	Source routing	Hop-by-hop
Advantages	• fast route discovery	• robust • small routing table
Disadvantages	• large overhead • large routing tables	• long initial delay

- **Hierarchical vs. non-hierarchical routing.** When different hierarchies of networks or routers exist, hierarchical routing can be applied. This is especially interesting when the locations are known and different types (speeds) of connections exist, i.e. the higher in hierarchy the faster the connection.

Table 3: Comparison hierarchical and non-hierarchical routing

	Hierarchical	Non-hierarchical
Advantages	• fast route discovery • location information	• more robust
Disadvantages	• hierarchical network needed	

- **Reactive vs. proactive routing.** In reactive routing protocols the route is only discovered when it is needed. In proactive routing protocols all routes are continuously search and kept up-to-date, so it is already available when it is needed. When delays are the bottleneck a proactive routing protocol must be used, but when overhead is a bottleneck reactive routing protocols must be used.

Table 4: Comparison reactive and proactive

	Proactive	Reactive
Advantages	• fast route discovery	• robust
Disadvantages	• large overhead	• long initial delay

- **Stochastic vs. deterministic routing.** Deterministic routing protocols use only one path and keep using it while it is available. Stochastic routing protocols uses stochastic channel variables like traffic and delay to choose a path, i.e. they predict the chance that a certain route is the best based on earlier measurements.

Table 5: Comparison stochastic and deterministic routing

	Stochastic	Deterministic
Advantages	• best throughput	• simple
Disadvantages	• difficult • database needed • multiple routes needed	• the best route may not be used

- **Multi-path vs. single path routing.** A multi-path routing protocol uses multiple parallel paths for routing. In most cases these path are completely disjunct, they do not use the same nodes and connections (excluding the source and destination). Of course multiple paths must be available for this method.

Table 6: Comparison multi-path and single path routing

	Multi-path	Single path
Advantages	• high throughput • robust	• easy • small routing table
Disadvantages	• difficult • large routing tables • multiple paths needed in the network	

3.1 Routing in MANETs

There are several routing protocols for routing in conventional ad-hoc networks. All of these protocols can be used for routing in Bluetooth, however many of them have disadvantages like they don't deal with the mobility or they use fixed routers (instead of mobile host acting as routers) or access points (base stations) or they have too much overhead for the limited capacity (memory, processing power) available in Bluetooth.

As described in [2] scatternets differ from classical ad-hoc networks, in terms of traffic characteristics, mobility patterns and scaling requirements. And it is not enough for two Bluetooth devices to be in each others transmission range to communicate, like in conventional mobile ad hoc networks.

Also location based routing methods exist, however in real ad hoc networks the position is not known and GPS or other location determination systems must be used. This however is to expensive for Bluetooth applications.

Zone routing protocols, like IERP and IARP, and landmark protocols, like LANMAR, are meant for very large networks and do not exploit their benefits in the scenario described for Bluetooth. Therefore they are not discussed in this paper, more information about these protocols can be found on the IETF MANET Charter [27].

3.1.1 Destination Sequenced Distance Vector (DSDV)

DSDV [8] is a hop-by-hop distance vector routing protocol. It requires each node to periodically broadcast updates of the routes known by the node. DSDV is based on distance vector routing, with the advantage that is guarantees loop-free routes. Another advantage is that the number of route updates is low. Disadvantages are that is does not support multi-path routing and that is has excessive overhead. The control overhead grows as $O(n^2)$, with n the number of nodes.

Each DSDV node maintains a routing table listing "the next hop" for each reachable destination. When used with Bluetooth this "next hop " can be the connection over which the packets have to be forwarded.

More information about DSDV can be found in [8],[14] and a comparison with other ad hoc routing methods can be found in [6].

3.1.2 Temporally-Ordered Routing Algorithm (TORA)

TORA [23] is a distributed routing algorithm based on a "link reversal" algorithm. It is designed to discover routes on demand, provide multiple routes to a destination, establishes routes quickly, and minimising communication overhead.

Route optimality, like shortest path, is considered of secondary importance and therefore longer routes are often used to avoid the discovery of newer routes. For each destination a separate copy of TORA is run in the node. When a route is needed, a node broadcasts a Query packet. When the destination, or a device which knows the destination, receives this Query it broadcasts an Update packet, containing its place with respect to the destination. All nodes that forward this Update packet adjust the route to the destination when necessary.

The advantages of TORA are the quick route discovery, the existence of multiple routes and the minimisation of overhead. The disadvantages are the secondary importance of route optimality and the dependence on IMEP, which provides a reliable, in-order delivery of all routing control packets, plus the notification of link status changes.

More information about TORA is given in [14] and a comparison between TORA and other ad hoc routing methods is given in [6].

3.1.3 Dynamic Source Routing (DSR)

DSR [7],[21] uses source routing, rather than hop-by-hop routing. Each packet carries the complete ordered list of nodes which it must pass.

DSR uses two mechanism, namely route discovery and route maintenance. Route discovery obtains the route to the destination by broadcasting Route Request through the network. To speed up the route discovery, each node maintains a cache of source routes.

The advantages are that intermediate nodes do not need to maintain up-to-date routing information and with that eliminating periodic route updates. However this can be a disadvantage when it wants to send packets itself, because now it has to perform a route discovery.

More information can be found in [14] and a comparison between DSR and other ad hoc routing methods is given in [6] and a comparison between DSR and AODV is given in [15].

3.1.4 Ad Hoc On-Demand Distance Vector (AODV)

AODV [22] is a combination of DSR and DSDV. It uses the route discovery and route maintenance from DSR and the hop-by-hop routing method from DSDV.

When a route is needed a Route Request is broadcasted through the network. Each node that forwards the request creates a reverse route for itself to the source. When the destination, or a node with a path to the destination, is found it replies the number of hops and a sequence to the destination. Each node that forwards this Reply, stores a forward route to this destination.

A comparison between AODV and other ad hoc routing methods are given in [6] and a comparison between AODV and DSR is given in [15].

3.1.5 Overview of MANET routing algorithms

Table 7: MANET routing algorithms

	Type	Type in relation with Bluetooth	Properties
DSDV	• Non-hierarchical • Hop-by-hop • Proactive • Deterministic • Single path	+ + + - - - -	• Excessive overhead
TORA	• Non-hierarchical • Hop-by-hop • Reactive • Deterministic • Multi path	+ + + + + - +	• Bad MANET performance • In-order delivery needed • Dependence on IMEP
DSR	• Non-hierarchical • Source routing • Reactive • Deterministic • Single path	+ + - + + - -	• Good performance in MANETs
AODV	• Non-hierarchical • Hop-by-hop • Reactive • Deterministic • Single path	+ + + + + - -	• Good performance in MANETs • Little overhead

3.1.6 Choosing a MANET routing algorithm

The differences between the above algorithms depend on the used models, like mobility and transmission rate. In [6] a simulation comparing these algorithms is described. This article is written by the inventors of DSR.

DSDV performs bad when the mobility is high. High mobility is when the nodes never pause and move with an average speed of 10 meters per second. The overhead of TORA and DSR is high in this case.

Their conclusion is that DSR is the best routing algorithm and that AODV is close.

Another article [15], written by the inventors of AODV, has a different conclusion. In simulations done with the same tools and environments as [6] the AODV algorithm performs better in situations with high load and many nodes. DSR performs better in their simulation when the density is small and the load is low.

Because the authors understand the design parameters of their own algorithms the best, it is a logical result that in the articles they write their own algorithm will perform best.

No independent comparison between the different MANET protocols is published yet. And the choice has to be made using these two comparing articles.

Because AODV is more suitable for Bluetooth than DSR, the choice for the routing algorithm for Bluetooth is AODV.

3.2 Routing in Bluetooth

After a network is formed all MANET routing protocols are applicable to Bluetooth for managing multi-hop routes through the scatternet. As seen before the main difficulty is the formation of the scatternet. Another difficulty is the service discovery handling in MANETs, this however is not part of this paper.

The question is however to determine the best routing protocol for Bluetooth scatternets. Such a routing protocol must deal with the following restraints:

- Bluetooth Devices are typically low power devices.
- A Bluetooth device has little processing power. This implies that the route calculations must be minimised.
- Little memory available. Therefore the size of the routing tables and caches is limited.
- Low transfer rate. Making the minimisation of routing overhead a primary concern.
- Bluetooth is a mobile ad hoc network. Therefore the discovery of a new route must be fast.

These restraints determine the type of routing protocol to be used:

Unless Bluetooth has a hierarchy (masters and slaves) a hierarchical routing protocol is not preferred, because there is only one type (one transfer rate) of connection and there is no location information available.

A reactive routing protocol is desired because bandwidth is a limiting factor and the longer delay is no problem.

A single path routing protocol is taken because the table size and processing power is limited and a multi-path routing protocol increases the routing table size and the number of calculations.

If possible a stochastic routing protocol is desired, because this improves the throughput and reduces the delay or power consumption. Because batteries are still very large and do not contain much energy, power consumption is a main issue for routing in scatternets

Signal power is a link property, like traffic-load or the BER (Bit Error Rate), and can be added to any route discovery method. This therefore does not influence the routing method choice.

The chosen routing method is based on the AODV routing protocol. This choice is based on the following properties:

- AODV is a reactive routing method, which reduces the number of transmission, because a route is discovered only when needed.
- AODV is hop-by-hop routing protocol which is better adapted to mobile ad hoc networks than source routing protocols.
- Based on the comparisons in [6] and [15], AODV has little overhead and good performance in MANETs.
- AODV can deal with the unreliability caused by the mobile ad hoc character of scatternets.

3.2.1 Route discovery

Route discovery is done by broadcasting a RouteRequest packet through the Bluetooth network. A node only forwards the first RouteRequest packet from a specific node it receives, this is checked by remembering the sequence number of the RouteRequest packet for each node. Then it adds an entry in its own table for the source of the

RouteRequest packet with the hop count as distance if it has no shorter route to the source. Before forwarding the packet it increases the hop count.

When the destination or a route that knows the destination is reached a RouteReply packet is generated and sent back to the source, via the shortest route. Because Bluetooth uses a master-slave relation based on polling, the reply path is as good as the just discovered path. This is because the transmission in both directions of each connection is controlled by the master and only when the link is defined symmetric. For performance aspects an asymmetric link could be better. This congestion control aspect is considered in chapter four.

To limit the broadcast a Time To Live (TTL) field can used, which has a maximum of the diameter of the network. A sequence field is used to ensure that no cyclic routes are created. This however is only needed when corrupted nodes are to be expected.

With this method two route types can be found: the shortest and the fastest.

Bluetooth however has the possibility to measure the incoming signal power. With this value the route with the lowest power consumption could be computed. In Bluetooth this value is called RSSI (Received Signal Strength Indicator).

When static nodes are used for providing network survivability, they can be used for adding location information to the route discovery method. However in a small office network this can be considered as overhead.

3.2.2 Route Maintenance

When a Bluetooth device moves or falls out, this has to be discovered by the surrounding nodes. When a data packet arrives and the link to the next device is not available, a number of options are possible for the device to do:

1. It deletes the packet and relies on the originator to discover a new route after a certain time-out, i.e. it depends on upper layers.
2. It deletes the packet and broadcasts a RouteError packet.
3. It deletes the packet and sends a RouteError packet to the originator.
4. It tries to discover a new route to the destination.
5. It tries to discover a new route to the destination and informs the originator that a link was broken and that there might be a better route to the destination.

In Bluetooth a device gets an event when a link is disconnected. This provides the possibility for a new route to be found before any packets have to be send. However a device does not know if the remote device has fallen out or moved or that it has moved itself. And power is used for the discovery of a new route, despite the route may not be necessary anymore.

As a result of the choice for a reactive routing protocol the third option is chosen. So when a connection is lost the neighbouring nodes delete all nodes through that link from their tables. When a packet is received for an unavailable node a RouteError packet is transmitted to the originator and all nodes on the route update their table.

3.2.3 Implementation Bluetooth Routing Method

With Bluetooth there is no need for Hello packets, because an event is generated when a connection is lost.

The HCI functions needed for Routing are SendHCIDataPacket and GetHCIDataPacket, to keep the implementation for masters and slaves the same, the broadcast is performed manually. The link layer broadcast available by Bluetooth is not used, because a slave can not perform this.

The HCI packet is shown in the figure below:

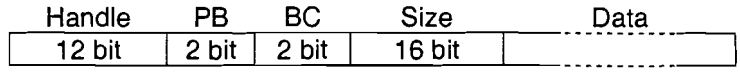

Figure 15: HCI packet

The Handle field holds the connection handle of the connection on which the packet has to be send.

In the PB (Packet Boundary) field must be set to '01' for a new L2CAP packet and '10' for continuing a fragmented packet. Other values are for future use.

When broadcasting, i.e. route requests, the BC field must be set to '01'. This indicates an active broadcast, which is suitable because only active devices are part of the network. When the BC field is set to '00' a point-to-point packet is created. The other possibilities include a piconet broadcast ('10'), which also sends the packet to parked slaves, and '11' is used for future use.

The size field holds the size of the data field in bytes.

In the data field the network packet can be encapsulated. This packet holds the routing header and a data field.

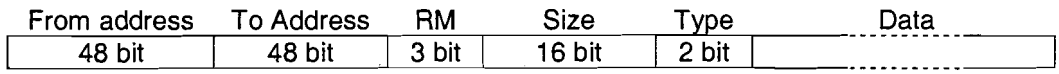

Figure 16: Router packet

In the RM field the routing method can be chosen:

Table 8: Routing method field options

RM field	Routing method
000	Fastest path routing (default)
001	Shortest path routing (fewest hops)
010	Power aware routing
011	Link quality routing
100	Cheapest path
...	Not used

When the default routing method is not used an extra field must be added to store the routing property. This can be the summed RSSI value or the summed link quality of all passed connections.

The type field holds the packet type which are displayed in table 9.

Table 9: Packet types

Options	Packet type
00	RouteRequest (RReq)
01	RouteReply (RRep)
10	RouteError (RErr)
11	Data

In the data field the higher layer data can be encapsulated.

The actual overhead depends on the packet type and the chosen routing method. The overhead per packet type is shown in table 10.

Table 10: Overhead per packet type

Packet	Data field size [bytes]	Transfer Rate (sym) [kb/s]	HCI header [bytes]	Routing header [bytes]	Data field Remaining [bytes]	Overhead [%]	Eff. transfer rate (sym) [kb/s]
DM1	18	108.8	4	15	3	83	18.5
DH1	28	172.8	4	15	13	54	79.5
DM3	123	258.1	4	15	108	12	227.1
DH3	185	390.4	4	15	170	8	359.2
DM5	226	286.7	4	15	211	6	269.5
DH5	341	433.9	4	15	326	4	416.5

ACL packets have the possibility to continue a packet. In this way a routing packet can be split over multiple ACL packets and performance can be improved.

The fields of the routing table are shown in table 11:

Table 11: Routing table composition

Field	Description
Destination	48 bit Bluetooth device address
Handle	Connection handle for the next hop
Timestamp	Lifetime of the current entry
Sequence	Sequence number of control packet from this destination
Routing method specific fields	
Hops	The number of hops to the destination
Power	Summed RSSI values of all hops on the current route
Link quality	Summed link quality for all connection on the route
Costs	Summed costs for this connection

The table must be updated when a connection change takes place or when an entry is timed out.

3.2.3.1 Multi-path Routing

Multi-path routing is difficult to combine with AODV. Because it is not wanted that the whole path is logged in the RouteRequest packet. In that case it would be a source routing algorithm. Another disadvantage of logging the route is that one unit must determine if the routes are disjunct, which requires processing power.

When multiple routes are wanted a new RouteRequest must be broadcasted for each new route. A node that already is part of a route between the source and the destination must delete this packet. In this way a new disjunct route is found if available.

Of course there are not more disjunct routes then the number of available connections on the current device. Given the 'almost' three connected network generated by the scatternet formation algorithm described in chapter two, in most cases there are three disjunct routes possible between a random source and destination pair.

When multiple paths are preferred a routing table entry has to exist for each route.

The choice if multiple routes are preferred unless the increased overhead due to the extra broadcasts, has to be made by higher layers. It is possible that packets do not arrive in order, this also has to be solved by higher layers.

3.2.3.2 Link Quality Routing

Bluetooth provides a function to determine the link quality. This function can be used to determine the best path. Where best is the least errors.

For each link the function Get_Link_Quality can be called and these values can be added in a field in the RouteRequest packet. The destination can chose the path with the highest overall quality. Because the link quality is equal for each direction of the link, the best return route is equal to the best forward route. The Get_Link_Quality function is described in appendix 3.

3.2.3.3 Power Aware Routing

Equal to link Quality Routing, however in this case the power is determined for each link and added in the RouteRequest packet. This can only be done with the 1mW Bluetooth version, because the 100mW version adjusts its own transmitting power to create an optimal link. There is no function defined to read this transmitting power level. With the 1mW version the power cab be determined by reading the RSSI (Received Signal Strength Indication). This can be done with the Read_RSSI function, which is explained in the appendix 3.

3.3 Conclusions

At this point research question two is answered. There are many routing methods available for routing in ad hoc networks (MANETs). When a scatternet is created all MANET routing methods can be used with Bluetooth, however the performance is the issue. Routing methods that use hierarchies or location information are not suitable for Bluetooth.

Four different MANET routing methods are described and compared. A routing method for Bluetooth is proposed that is based on AODV. This choice was made because AODV has the desired properties for Bluetooth and performs good in MANET simulations.

In addition to the fastest path route discovery, several other route discovery techniques were described. These techniques include multi-path routing, link quality routing and power aware routing. However, due to the limited bandwidth, implementing these methods is not recommended.

Because the size of the header of a routing packet is 15 bytes, in one slot packets, which have a pay load of 18 bytes, the overhead per packet is 83 percent. When five slot packets are used, this overhead drops to four percent. This overhead does not cover the routing overhead caused by the routing method, i.e. the route requests and route reply packets.

4 Bluetooth Usage Models

Before the routing method can be optimised, first the information flow in the network must be considered. The information flow between Bluetooth devices is shown in figure 17.

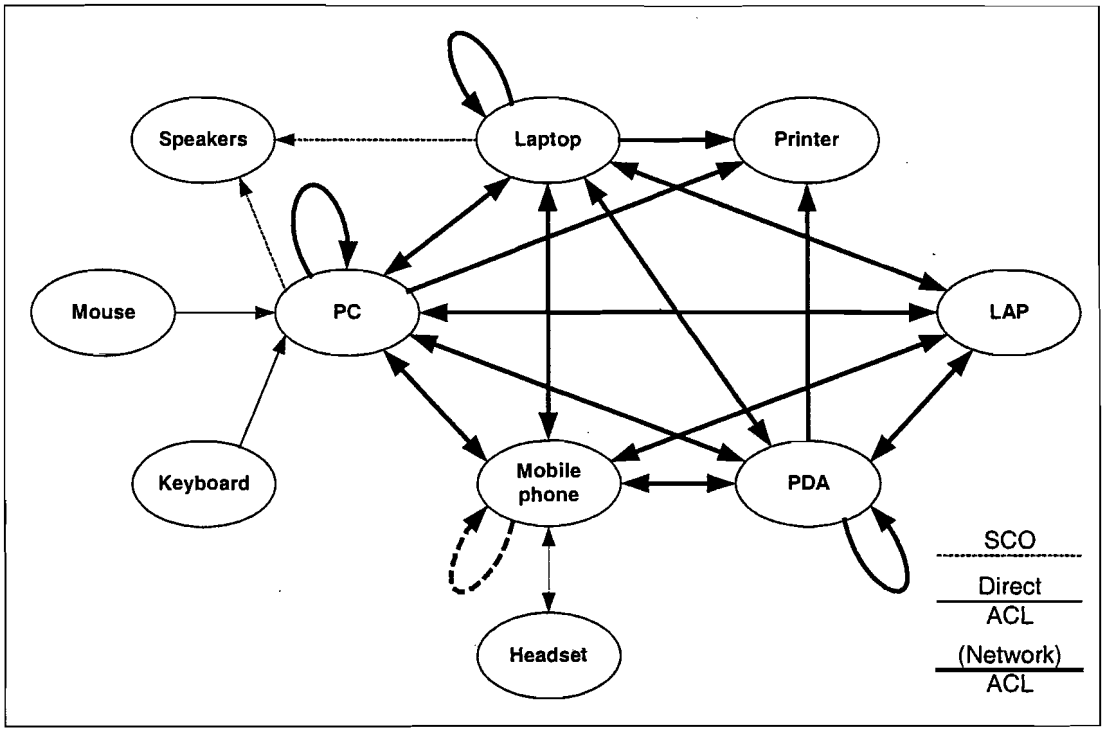

Figure 17: Information flow Bluetooth devices

The thick lines in the figure imply that the information can be routed through a scatternet. The user profile, network, and routing method have impact on this flow. They will be described in the following paragraphs.

4.1 User Profiles

The information flow depends on the applications users run and the traffic generated by these applications.

Table 12 shows the dependencies between Bluetooth devices and applications.

Table 12: Usage Bluetooth Devices

	Office applications	Internet (WAP)	E-mail	Agenda	File / document exchange	Printing	Dial Up Networking	Calling	SMS	Interworking	Interactive Conferencing	Cordless Desktop	Synchronising
PC	✓	✓	✓	✓	✓	✓	✓					✓	✓
Laptop	✓	✓	✓	✓	✓	✓	✓				✓		✓
PDA		✓	✓	✓		✓			✓		✓		✓
LAP										✓			
Printer						✓						✓	
Mobile phone		✓					✓	✓	✓	✓			
Keyboard												✓	
Mouse												✓	
Headset								✓					
Speakers												✓	

Unless devices like headsets and mice are mentioned in this table, they are not part of the network, i.e. they are probably not able and wanting to route and thus have the 'routing bit' disabled. Of course these devices interact with the network and therefore they are placed in this table.

The different traffic types are summed in table 13:

Table 13: Data traffic types

Type	Properties	Example
Constant flow	A constant symmetric flow of data with little variation from one node to another.	Online meeting; chat
Burst	A short data transmission, usually in one direction, that uses the full bandwidth.	Printing or exchanging a file; internet (HTTP)
Polling	A periodic short data burst.	Polling for new e-mail on a mailserver.
Real-time	A constant data flow that is transmitted near real-time. Quality is adjusted to available bandwidth.	Online meeting (real audio, real video)

4.1.1 Possible user profile

With these properties a possible user profile can be described. This profile describes one room in an office environment. In one room two desks are placed, each with a computer on it that is connected with an Ethernet network. Both of the persons seated in the room have a PDA that synchronises via Bluetooth. A Bluetooth enabled printer is located in the room for printing from the computers, PDA's and laptops.

In the room a meeting table is located where presentation sheets and business cards can be exchanged between laptops and PDA's respectively.

At any place e-mail will be available on computers, laptops and PDA's. Mainly text based internet pages will also be available on these mobile devices.

The computer can be part of a cordless desktop via which a Bluetooth enabled mouse and keyboard are connected. And a Bluetooth enabled mobile phone can be used for dial-up networking.

4.2 Network Specific Usage Models

In ad hoc networks there exist special places:

- **High density areas**: Around printers, servers and access points a higher density can be expected, i.e. a person walks with his PDA to a printer and prints, so he can see the paper coming out. In a room were multiple people are seated more Bluetooth devices can be expected.
- **Low density areas**: Several low density areas can exist in an ad hoc Bluetooth network. Such areas include corridors and stairs. As stated before static nodes can be added to provide connectivity. When the devices in such a low density area connect two or more high density areas, traffic-load problems can be expected.
- **Bluetooth devices with non-routers attached**: The non-routers are not included in the network, however they occupy the routing-enabled Bluetooth device. Two examples are:
    - PC with keyboard and mouse: The PC must maintain a real-time connection with the keyboard and mouse. The remaining slots can be used for networking. There must be a difference in priority for ACL packets.
    - Mobile phone with headset: When in use the mobile phone has a SCO connection with the headset, there are only four slots left for ACL packets. The SCO packets already have higher priority than the ACL packets.

All manufacturers are currently only thinking about themselves, the Bluetooth-enabled keyboard manufacturer assumes that it can completely use the BT chip in a PC. However the manufacturers of mice, speakers, mobile phones, and other PC add-ons assume the same. One BT chip can not support multiple SCO connections (to the speakers and mobile phone) and real-time connections (to the keyboard and mouse) at the same time. A solution to this problem can be multiple Bluetooth chips in one device, this problem however is not part of this research.

4.3 Routing Method Specific Usage Models

The routing method has great impact on the distribution of data on the network. Because all connections are equal, the best routing algorithm provides an equal distribution of the data over all connections.

To determine the maximum network load in an ad hoc Bluetooth network, where all nodes are sending a constant stream of data, a scatternet was created. This scatternet was created by the simulator tool after running for a while and is shown in figure 18.

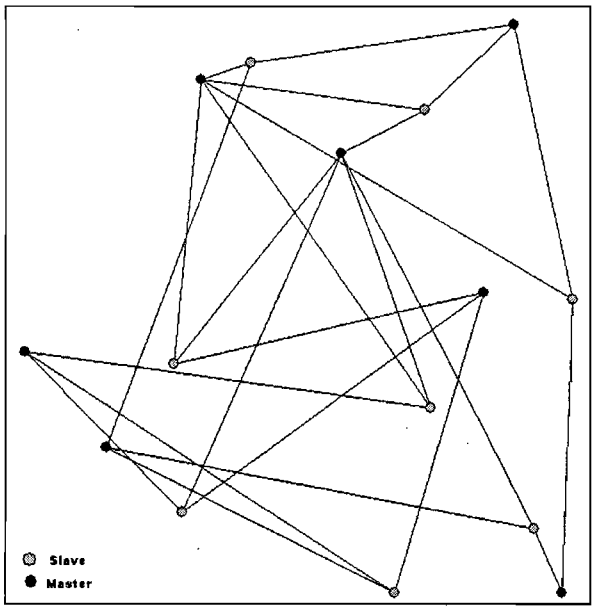

Figure 18: Scatternet with 15 devices

The routes between all nodes are determined using a fastest path routing algorithm, i.e. with the routing method described in chapter three. For each connection is calculated by how many routes it is used and this is displayed in figure 19. The thicker the line, the more routes use the connection.

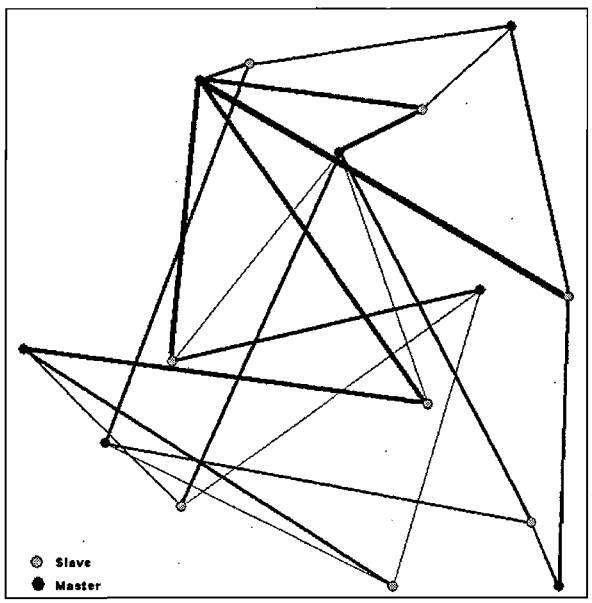

Figure 19: Traffic load per connection

As can be seen in the figure the load is not well spread. The number of routes per connection vary from three to twenty one.

To show the impact of this problem it is assumed that all routes are used at the same time, that the transfer over all routes is the same and all connections are assumed symmetric.

Given that the maximum capacity of a symmetric Bluetooth connection is 433.9 kb/s, this results in connections with 21kb/s per route. This example uses DH5 packets, which are not always usable in scatternet implementations. In the more practical situation when DM1 or DM3 packets are used the data rate drops to 5kb/s, 12kb/s per route respectively.

When, using an optimal routing method, the load is equally spread over the connections, there are still about ten routes per connection, this results in data rates from 10kb/p,

24kb/s for DM1 and DM3 packets respectively. Considering that a route is as slow as the slowest link, this has major impact on the information exchange in a Bluetooth network.

Of course, when a more burst traffic is generated this situation gets better, however it makes clear that there is a problem.

A non equally spread traffic load situation as shown above has great influence on the scheduling demands. And problems can be expected in devices that are part of many routes.

4.4 Conclusions Bluetooth Usage Models

With the current properties a Bluetooth network can be used to exchange small amounts of data like business cards, presentation sheets, mainly text based internet pages, and incidentally a small printer file. All these applications have a burst traffic pattern.

When the demands are higher, like online meeting, file exchanges, and internet downloads, a high speed backbone is desirable which connects scatternets. This is because when more routes use the same connection, the throughput per route becomes very low, using a backbone solves this problem.

The main impact on the routing method is the interworking between Bluetooth and this backbone network.

A simple implementation of this backbone are LAPs connected to a fast Ethernet network. In the future this backbone can be a GPRS or UMTS network, which increases the mobility.

5 Properties Of The Bluetooth Routing Method

5.1 Scheduling and Congestion Control

Two types of scheduling are involved in Bluetooth. These are inter-piconet scheduling and intra-piconet Scheduling. The inter-piconet scheduling is done by the piconet master and is already enclosed by the specification. The intra-piconet scheduling is not enclosed by the specification yet. This paragraph discusses scheduling in scatternets, which encapsulates both inter- and intra-piconet scheduling.

The difficulties concerning scheduling in Bluetooth are:

- The SCO connections have higher priority than ACL connections, i.e. there are reserved slots.
- Because the master can only send packets to a slave in even slots and a slave can only send in odd slots, scheduling occurs in pairs of slots.
- The ACL data packets can have three sizes: one, three or five slots and the cannot span across different voice slots.
- Scheduling depends on the usage profile of the scatternet.

These difficulties will be discussed below.

As stated before, Bluetooth has SCO connections for transferring voice between two neighbouring devices. Because the voice packets have higher priority than data packets, the scheduling algorithm must deal with this restriction. Therefore it is wanted that routing devices do not maintain SCO connections. Because a device can not perform an inquiry when is has a SCO connection, it is not likely that it will become part of the network. The demand that no SCO connections are wanted is in line with the scope, because the summed devices (laptops, PDA's, printers and LAPs) are not expected to maintain SCO connections in the given scenario.

Because scheduling occurs in pairs of slots, it is more efficient to send information in the direction that the most data is coming from. In most information exchanges this is already the case. However, while surfing the Internet more information is received than sent and the opposite counts for printing.

With one slot packets scheduling is easy, but the transfer rate is slow, and with five slot packets the transfer rate is good, but scheduling is difficult.

5.1.1 Optimising Connections

In an office network there is a big chance that there is more traffic towards a LAP or printer and therefore it can be interesting to use asymmetric connection to enlarge the throughput towards these devices. The decision to use a symmetric or asymmetric link is made by the Bluetooth link layer, but can only be made when DM3 – DH5 packets are used.

In order to obtain a higher throughput another packet type has to be chosen. The symmetric and asymmetric data rates for the different ACL packets are shown in table 14. The composition of ACL packets is shown in table 1 in chapter two.

Table 14: Asymmetric Connection Data Rates

Type	Max Data rate [kb/s]		
	Symmetric	Asymmetric	
		Forward	Reverse
DM1	108.8	108.8	108.8
DH1	172.8	172.8	172.8
DM3	258.1	387.2	54.4
DH3	390.4	585.6	86.4
DM5	286.7	477.8	36.3
DH5	433.9	723.2	57.6

From table 14 the difference in traffic load for the two directions can be calculated, at which the connection type must be changed.

This method has impact on the route discovery algorithm, because the route back is not equal to the forward route. When this is implemented in standard AODV, the found route is the fastest forward route. This is good for printing, but with Internet the reverse route has probably a higher traffic load and a fast reverse route is desired.

When one connection has a much higher traffic load than the other connections more optimisation can be achieved by changing the packet type.

These optimisations are illustrated in the following example:

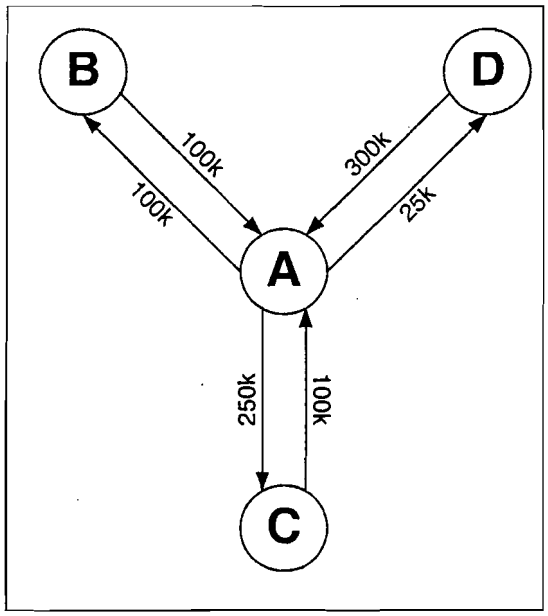

Figure 20: Example traffic load per connection

Given the situation in figure 20, the optimal connection set up is displayed in table 15.

Table 15: Asymmetric connection set up example

Connection	Packet Type	Data rate [kb/s]	
		Up	Down
A - B	DM1	108.8	108.8
A - C	DM3	258.1	258.1
A - D	DM3	387.2	54.4

In order to perform the scheduling with two DM3 and one DM1 connection, currently the device has to be master. When the device has more connections, using DM3 – DH5 packets becomes more difficult. Another difficulty is when to change the connection set up.

5.1.2 TCP/IP

Because of the common use, it is desired to run TCP/IP over the Bluetooth network. This is already possible, however the used method is very inefficient. The current and the desired method are shown in the following figure.

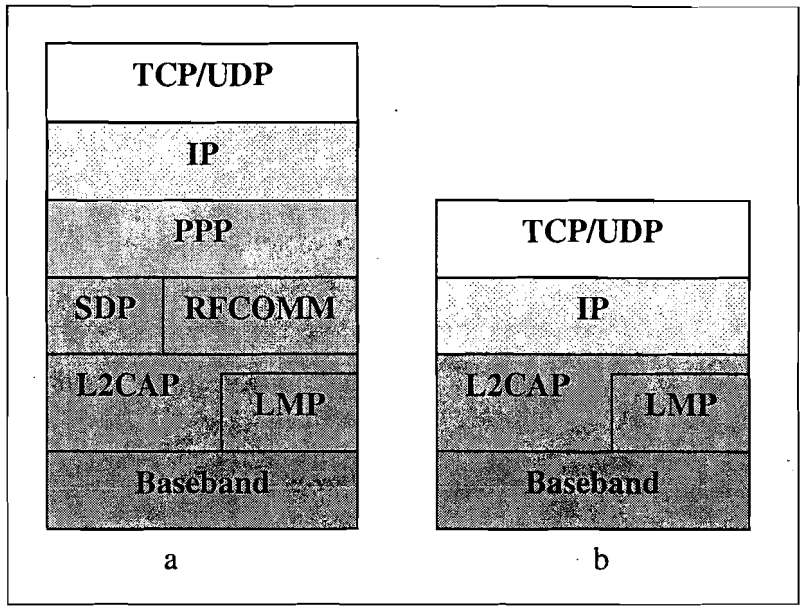

Figure 21: TCP/IP over Bluetooth: a)current stack; b) desired stack

TCP is a transport protocol that provides sequencing and flow control, were UDP relies on other layers for these properties. Because in Bluetooth the flow does not have to be controlled (all devices send at the same rate and polling is used), TCP can be considered as overhead and is therefore not wanted.

The main difficulties with IP over Bluetooth are:

- IP was not designed for mobility and it assumes the device to stay in the same IP subnet. An IP router routes the information based on the location of a device. This location can be determined from the IP address. IP runs over a hierarchical network.
- Header compression must be done, because the bandwidth of Bluetooth is limited.
- Multicast support is desired when streaming real time applications are used.
- DNS name and address resolution must be handled. This is a big challenge in an ad hoc network like Bluetooth.
- Mobile IP and Cellular IP are solutions for IP for mobile devices. Also for Bluetooth there is an IP solution available: BluePAC. BluePAC [31] uses fixed routers instead of mobile routers and therefore no self-organising ad hoc network is created.

Currently, to use TCP/IP applications, IP packets have to be tunnelled through Bluetooth. This results in a situation in which one network protocol is above another.

The aspects and performance of different TCP implementations and different MAC and SAR scheduling algorithms for piconets are discussed in [18]. From this article it becomes clear that the maximum TCP throughput is between 365 kb/s and 380 kb/s for different TCP versions. This is half of the specified 720 kb/s connection. In an error free environment this could be increased to ca. 450 kb/s. Because of the free frequency band, where also other radio communication systems are transmitting, such an error free environment is an exception.

TCP/IP throughput and delay measurements in piconets are added in appendix 5.

5.2 Security

Security in Bluetooth can be compared with security issues dealt with on Internet. It consists of three factors:

- Security (secrecy): information must be protected from third-parties, who are not allowed to obtain the information.
- Integrity: information can not be changed along the route.
- Authentication: Is the other entity who it says that it is?

In Bluetooth security is only provided for the link between two devices. However, this security has no use when the two devices don not have a direct link. This is because the intermediate users can not be trusted. Therefore the security must be provided on application level, depending on the demands and services.

To provide security, DES or any other encryption algorithm can be used. Symmetric encryption algorithms are preferred, because they are faster than asymmetric. Integrity can be provided with a hash function, which is for example provided by SHA. Authentication is a more difficult problem, because this can only be provided by the use of public and private keys. The RSA algorithm can be used with these keys. These keys have to be managed by a trusted authority, like the BSIG.

The current Bluetooth specification already contains private keys, and the Bluetooth device address is used as a public key. These keys can be used to provide security in a scatternet.

To secure web pages the SSL protocol is used, a variant of this protocol can be used in Bluetooth. Because it is an application layer protocol it is not affected by changes in the path between two entities. More information about the SSL protocol can be found in [28]

The use of public and private keys in encryption is illustrated in appendix 2.

5.3 Scalability

Scalability is limited because the delay in each node is too high to create a large network. This delay is especially important in real-time and speech applications. Within the given office scenario a delay within 0.5s is desired. This can be the delay experienced over the characteristic path length or over the maximum path length. The characteristic path length is the path length between two nodes, averaged over all pairs of nodes.

To determine the scalability of the network two things are needed. These are the delay per hop and following from that the average maximum path length at which the maximum delay occurs.

First the delay per hop is determined. It depends on the following values:

- The chosen MAC scheduling algorithm, for both masters and slaves.
- The processing power and memory of the Bluetooth device.
- The routing method, i.e. the search speed in routing tables and the handling and interpretation of the packet.

The delay in each hop is therefore device specific.

The total delay depends on the transmission rate and the delay at the hops. It can be calculated as follows:

$$D_{total} = \sum_{Hops} D_{hop} + \sum_{Connections} (TransferRate^{-1} \cdot PacketSize)$$

With D_{total} the delay between the source and the destination and D_{hop} the delay in a Bluetooth device, i.e. the time between receiving a packet and forwarding it.

D_{hop} is determined by using Bluetooth PCMCIA devices from Digianswer and a Microsoft Windows application using the HCI interface. The results for the different packet types is displayed in table 16. Also the transport time for the different packages is calculated and displayed.

Table 16: Delays

Packet type	Average D_{hop} [ms]	Standard deviation [ms]	Transport time [ms]
DM1	55.0311	14.0682	1.292509
DH1	62.2566	10.9995	1.265914
DM3	50.3372	15.4682	2.461583
DH3	61.7665	15.0613	2.454747
DM5	52.2184	9.58349	3.678971
DH5	56.1472	11.8672	3.672912
Average	56.2928		2.471106

No packets were buffered during this measurement.

Now the average maximum path length can be calculated:

The number of hops can be calculated when the number of connections is set equal to the number of hops. Which is true when D_{hop} at the source and destination is half of the average D_{hop}.

$$D_{total} = hops \cdot (D_{hop} + T_{transport})$$

$$500 = hops \cdot (56.29 + 2.47)$$

$$hops \approx 8$$

The average maximum path has a length of eight hops. Now the number of nodes in the network must be determined. This maximum number of nodes depends on the shape of the network: with a stretched network the number of nodes will be smaller than with a round network.

In literature this problem belongs to the random (geometric) graph theory. The graph is geometric because each node can only connect with other nodes in range. However an estimation of the maximum scatternet size is done by simulating several scatternets and using trend lines to estimate by which numbers of nodes the maximum path is 8 hops long.

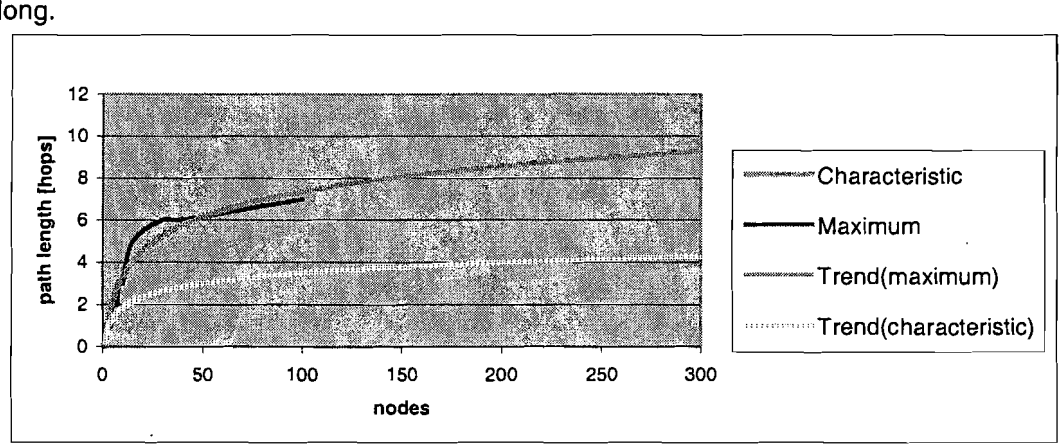

Figure 22: Scatternet size and the maximum and characteristic path

From this figure a maximum scatternet size of about 150 nodes can be read, because at that point the maximum path length trend line crosses the eight hops grid line.

When an average delay of 0.5 seconds is wanted, the characteristic path length trend line makes clear that large networks are possible. However very large delays may occur occasionally and that probably is not desired.

Also the use of the frequency band increases with the number of devices and thus limits the scalability. No performance measurements are known with a large number of Bluetooth devices in the same area, therefore this influence is a subject for further study.

5.4 Robustness

Two types of robustness are present. The first, network survivability, is obtained with the scatternet formation algorithm proposed in chapter two. By creating redundant connections there exits an alternative route between all pairs of nodes at any time. In this way mobility and fall-out do not result in a network split.

The second type of robustness is provided by the routing method. The chosen routing method adapts to mobility and fall-out, without generating too much overhead.

With these two properties a robust routing method for Bluetooth networks is created.

5.5 Billing

Because there is no central entity, billing is a very difficult issue. Questions to be answered are when a user must pay and for what. And does a user earn money for relaying information.

Transporting the information is not a service provided by the network operator anymore. Therefore other services have to be developed in order to earn money. In this situation people pay for the service they request and the question has to be asked if they must pay each time they use the service or should they be billed once a month. Of course this depends on the service provided. These services can be managed by Jini [29] or Salutation [30], which are methods for automatic service configuration.

When a user pays each time a service is requested the most simple solution is a prepaid card which its value is decreased each time a service is requested. The value might be increased when the device relays information for other devices.

An area where services can be provided is in public buildings, where users can connect their PDA's or Laptops to the internet or other services, i.e. a train or bus station where tickets can be ordered, newspapers can be downloaded and delays are available by the use of Bluetooth.

However, in an office environment this is not an applicable solution and the money must be made at the edge of Bluetooth networks, like the connection with a telephone network. In this case Bluetooth increases the use of the mobile and fixed networks.

Billing in ad hoc networks is an open issue and not in the scope of this research.

5.6 Conclusions Routing Properties

At this point research question three is answered. The different properties of the routing method have been dealt with.

Scheduling and congestion control are very difficult due to the chosen technology for Bluetooth. Most problems are link layer problems like MAC scheduling. The different connection and packet types have great influence on these properties. Therefore some optimisations concerning connections have been proposed.

The security provided by Bluetooth can not be used for transferring secure information through scatternets. A SSL solution using the provided keys can be used for security in scatternets.

Scalability depends mainly on two factors, namely delay and the frequency band usage. The delay limits the scatternet to circa 150 devices, the influence on the frequency usage with many devices in the same area is an open issue.

For the used scenario the profit must be made at the edge of the Bluetooth network, i.e. the connections with the fixed core network or with UMTS or GPRS. Billing in ad hoc networks is still an open issue.

6 Conclusions & Recommendations

6.1 Conclusions:

Before routing can be done in Bluetooth a well connected scatternet must be created that provides network survivability. The presented scatternet formation algorithm forms a network that:

- Optimises throughput
- Includes the most active units
- Is self organising
- Is robust

This is done without placing nodes in standby mode and without performing master-slaves switches. Also masters do not need to inquire. At least 20 active devices per $400m^2$ are needed to create a well connected network.

When a scatternet is created all MANET routing protocols can be used with Bluetooth. However there are large differences in performance. Hierarchical and location based routing methods are not suitable and therefore not researched.

Four different MANET routing methods are described and compared. A routing method for Bluetooth is proposed that is based on AODV. This choice was made because AODV has the desired properties for Bluetooth and performs good in MANET simulations.

Due to the low connection speed and predefined packet types it is not possible to get good performance in a Bluetooth network. The overhead per data packet can be up to 83%. When routing control overhead is added, the throughput will be very bad.

Bluetooth has the possibility to determine the power consumption and quality for each link. With these values alternative routes can be discovered, limiting the power usage or maximising the quality. However the extra overhead does not make it profitable.

With the current properties a Bluetooth network can be used to exchange small amounts of data like business cards, presentation sheets, mainly text based internet pages, and incidentally a small printer file. All these applications have a burst traffic pattern.

When the demands are higher, like online meeting, file exchanges, and internet downloads, a high speed backbone is desirable which connects scatternets. Such backbone can be a fast Ethernet network or a mobile network like UMTS or GPRS. In this case the main impact on the routing method is the interworking between Bluetooth and this backbone network.

Scheduling and congestion control are very difficult due to the chosen technology for Bluetooth. Most problems are link layer problems like MAC scheduling. The different connection and packet types have great influence on these properties. Therefore some optimisations concerning connections have been proposed.

The influence of a TCP/IP connection over Bluetooth is very bad for the performance. Because TCP and IP were not developed for mobile ad hoc networks and because TCP and Bluetooth have overlap in functions it is not efficient to use TCP/IP over Bluetooth.

The security provided by Bluetooth can not be used for transferring secure information through scatternets. A SSL solution using the provided keys is described for security in scatternets.

Scalability depends mainly on two factors, namely delay and frequency band usage. The delay limits the scatternet to circa 150 devices. Delays will become a big problem in all mobile ad hoc networks which use in-between nodes to relay information.

Billing in ad hoc networks is still an open issue.

6:1.1 State of Bluetooth Technology

Many problems were experienced because the Bluetooth technology is still under development. These problems will be described below.

The specification is not completed yet, many items are not clear and are solved by different manufacturers in their own way. Also the L2CAP and LMP service primitives are not specified, so a real OSI layer three network protocol could not be specified.

Some of the needed functions are not implemented by manufacturers, unless they claim to be completely conform specification. An example is the 'connect as' option in the AcceptConnection function.

Unless the specification provides networking for Bluetooth, no manufacturer has implemented scatternets yet. This probably is because it is not a primary goal. Bluetooth was developed for connecting two devices over a radio interface, in a later stage the networking option was added. Most current profiles describe how to connect two devices in a specific situation, in the latest update an networking profile was added. This profile describes how to emulate Ethernet in a piconet.

The manufacturers support regarding HCI applications is minimal, their primary goal is to implement profiles which developers can use directly, like emulating serial connections over Bluetooth. With the used hardware, relaying information in a piconet was not possible using HCI functions. Also many bugs were discovered in drivers and libraries and as many updates of libraries were obtained. This makes clear that the manufacturer has too little time to completely test their products and that support to other developers is of secondary interest.

6.2 Recommendations:

With the current state of the Bluetooth technology, it is recommended that KPN should wait with ad hoc networking in Bluetooth until the technology is matured.

6.3 Further study

The first topics for further study are Bluetooth specific and should not be done by KPN:

- Performance measurements scatternet / multiple Bluetooth devices.
- Changes in connection set-up, like symmetric and asymmetric links, and packet types to increase performance.

Research topics interesting for KPN:

- Interworking between Bluetooth and UMTS, GPRS and other fast networks.
- Ad hoc networks with larger bandwidth, like HIPERLAN and WLAN.
- QoS in mobile ad hoc networks.

References

[1] Haartsen, J
BLUETOOTH-THE UNIVERSAL RADIO INTERFACE FOR AD HOC, WIRELESS CONNECTIVITY
Ericsson Review, Vol.75, no.3; 1998, pages 110-117.

[2] Bhagwat P. and A. Segall
A ROUTING VECTOR METHOD (RVM) FOR ROUTING IN BLUETOOTH SCATTERNETS.
1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99), 1999, pages 375-379.

[3] Hu, Y and D.B. Johnson
CACHING STRATEGIES IN ON-DEMAND ROUTING PROTOCOLS FOR WIRELESS AD HOC NETWORKS
In: Proceedings of the sixth annual ACM/IEEE international conference on Mobile computing and networking (MobiCom'00), August 6 - 11, 2000, Boston, MA USA, pages 231 – 242.

[4] Johnson, D.B.
ROUTING IN AD HOC NETWORKS OF MOBILE HOSTS
In: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, 1994, pages 158-163.

[5] Miklos, G and A. Racz, Z. Turanyi, A. Valko, P. Johansson
PERFORMANCE ASPECTS OF BLUETOOTH SCATTERNET FORMATION
In: Proceedings of the first annual workshop on Mobile Ad Hoc Networking Computing (MobiHOC'00), 2000, pages 147-148.

[6] Broch, J. and D.A. Maltz, D.B. Johnson, Y. Hu, J. Jetcheva
A PERFORMANCE COMPARISON OF MULTI-HOP WIRELESS AD HOC NETWORK ROUTING PROTOCOLS
In: Proceedings of the fourth annual ACM/IEEE International Conference on Mobile Computing and Networking (MoMuC'98), 1998, pages 85-97.

[7] Johnson, D.B. and D.A. Maltz
DYNAMIC SOURCE ROUTING IN AD HOC WIRELESS NETWORKS
Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181. Kluwer Academic Publishers, 1996.

[8] Perkins, C and P. Baghwat
HIGHLY DYNAMIC DESTINATION-SEQUENCED DISTANCE-VECTOR ROUTING (DSDV) FOR MOBILE COMPUTERS
In: Proceedings of the SIGCOMM'94 Conference on Communications, Architectures, Protocols and Applications, 1994, pages 234-244.

[9] Kalia, M and D. Bansal, R. Shorey
MAC SCHEDULING AND SAR POLICIES FOR BLUETOOTH: A MASTER DRIVEN TDD PICO-CELLULAR WIRELESS SYSTEM
IEEE international workshop on Mobile Multimedia Communications (MoMuC'99), 1999, pages 384-388.

[10] Wu, S.F. and C. Perkins, P. Bhagwa
CACHING LOCATION DATA IN MOBILE NETWORKING
In: Proceedings of the IEEE workshop on Advances in Parallel and Distributed Systems, IEEE Comput. Soc. Press, Los Alamitos, USA, 1993, pages 71-76.

[11] Zhou, J. and K. Lam
UNDENIABLE BILLING IN MOBILE COMMUNICATION
In: Proceedings of the fourth annual ACM/IEEE international conference on Mobile computing and networking (MobiCom'98), October 25 - 30, 1998, Pages 284 - 290

[12] Correia, L.M. and R. Prasad
AN OVERVIEW OF WIRELESS BROADBAND COMMUNICATIONS
IEEE Communications Magazine, vol 35 no. 1, January 1997, pages 28 – 33.

[13] Fernandes, L.
MBS – MOBILE BROADBAND SYSTEM
In: Proceedings ICUPC'93 Canada, 1993

[14] Groten, D
SELF-CONFIGURING WIRELESS AD HOC NETWORKS (SWAN) – ROUTING STRATEGIES, OPPERTUNITIES AND LIMITATIONS
KPN Research report 00-32635, February 2001.

[15] Das, S.R. and C.E. Perkins, E.M. Royer
PERFORMANCE COMPARISON OF TWO ON-DEMAND ROUTING PROTOCOLS FOR AD HOC NETWORKS.
In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Tel Aviv, Israel, March 2000, p. 3-12.

[16] Penrose, M.D.
ON K-CONNECTIVITY FOR A GEOMETRIC RANDOM GRAPH
Random Structures and Algorithms, *Volume: 15, Issue: 2, Date: September 1999, Pages: 145-164*

[17] Haartsen, J.C. and S. Mattison
BLUETOOTH – A NEW LOW-POWER RADIO INTERFACE PROVIDING SHORT-RANGE CONNECTIVITY
IEEE volume 88 No. 10 October 2000

[18] Das, A and A. Ghose, A. Razdan, H. Saran, R. Shorey
ENHANCING PERFORMANCE OF ASYNCHRONOUS DATA TRAFFIC OVER THE BLUETOOTH WIRELESS AD-HOC NETWORK
In: IEEE proceedings on INFOCOM 2001, Volume: 1 , 2001
Page(s): 591 –600

Electronic literature

[19] Specification of the Bluetooth System, Volume 1: Core, Version 1.0b,
The Bluetooth Special Interest Group (BSIG), 2000
URL: http://www.bluetooth.com
Date last check: 2001-01-11

[20] Specification of the Bluetooth System, Volume 2: Profiles, Version 1.0b,
The Bluetooth Special Interest Group (BSIG), 2000
URL: http://www.bluetooth.com
Date last check: 2001-01-11

[21] Johnson, D.B. and D.A. Maltz, Y. Hu, J.G. Jetcheva
The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks
IETF MANET Charter, Under work
URL: http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-04.txt
Date last check: 2001-06-11

[22] Perkins, C.E.
Ad Hoc On-DEMAND DISTANCE VECTOR (AODV) ROUTING
IETF MANET Charter, 1997
URL: http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-07.txt
Date last check: 2001-06-11

[23] Park, V.D. and M.S. Corson
TEMPORARELY ORDERED ROUTING ALGORITHM (TORA)
IETF MANET Charter, 1997
URL: http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-03.txt
Date last check: 2001-06-11

[24] Wireless Local Area Networks (WLAN)
IEEE Working group for WLAN standards, 2000
URL: www.ieee802.org/11/
Date last check: 2001-06-11

[25] HIPERLAN Standard
European Telecommunications Standards Institute (ETSI), 2000
URL: http://www.etsi.org/technicalactiv/
Date last check: 2001-06-11

[26] IrDA website
The Infrared Data Association
URL: http://www.irda.org
Date last check: 2001-06-11

[27] IETF MANET Charter
Mobile ad hoc network charter
URL: http://www.ietf.org/html.charters/manet-charter.html
Date last check: 2001-06-11

[28] How SSL works
Secure Sockets Layer description
URL: http://developer.netscape.com/tech/security/ssl/howitworks.html
Date last check: 2001-06-11

[29] The community resource for Jini™ technology
URL: http://www.jini.org/
Date last check: 2001-06-11

[30] Salutation Consortium website
URL: http://www.salutation.org
Date last check: 2001-06-11

[31] Albrecht M. and M. Frank, P. Martini, M. Schetelig, A. Vilavaara, A. Wenzel
IP SERVICES OVER BLUETOOTH: LEADING THE WAY TO A NEW MOBILITY
URL: www.cs.uno.edu/~golden/6990MC/MobilePapers/ipoverbluetooth.pdf
Date last check: 2001-06-11

[32] Public Key Infrastructure – The Verisign Difference
URL: http://www.verisign.com
Date last check: 2001-07-04

[33] Building Corporate Public Key Infrastructure
URL: http://www.infoseceng.com/corppki.htm
Date last check: 2001-07-04

List Of Figures

List Of Tables

Appendix 1 Alternatives for Bluetooth

There are several alternatives for Bluetooth available, they are given here only for information about the working space of Bluetooth. At the end a comparison is given with different radio networks.

1.1 IrDA

IrDA [26] is an alternative for Bluetooth, which uses infrared light for communication between devices. IrDA uses a narrow beam of light (30°) and works with distances between zero and one meter. Like Bluetooth, IrDA is used for communication between laptops, printers and PDA's. The advantages for Bluetooth above IrDA are the larger possible distance, no direct optical path (line-of-sight) is needed and the networking possibilities. The disadvantages are the more expensive chips and the lower data rate.

When the Bluetooth devices become mass produced, the chip price will decline and the greatest disadvantage will disappear.

More information about IrDA can be found at the IrDA web-site [26]

1.2 WLAN (IEEE 802.11)

The IEEE WLAN (Wireless LAN) standard [24] provides two physical layer radio specifications, operating in the 2.4-2.4835GHz band, and one infrared specification. The two radio specifications are a 1Mb/s version and a 2Mb/s version. The infrared specification provides a 1Mb/s connection (and an optional 2Mb/s connection).

The WLAN standard continuously being updated to current needs, right now there already is a 11Mb/s connection (IEEE 802.11b) and a high speed version is worked on.

The main disadvantage of WLAN is that it is developed for LAN access of computers, where Bluetooth is a multipurpose system. As a result WLAN will not be as largely implemented as Bluetooth. An advantage for networking is that it is compatible with Ethernet. More information can be found at the IEEE 802 web-site [24].

1.3 HIPERLAN

HIPERLAN [25] is a high speed local area network which uses the 5GHz radio band for communication. HIPERLAN is developed for the wireless communication between computers and a (wired) core network. There are two types defined, type 1 is the long range variant (800m at 1Mb/s) and type 2 is the short range, indoor, variant (50m at 20Mb/s). Type 2 is meant for office use and thus the alternative for Bluetooth. At this time HIPERLAN/1 is becoming obsolescence and HIPERLAN/2 is being upgraded for a larger range and higher speeds.

The application area's of HIPERLAN type 2 and Bluetooth partially overlap, however Bluetooth is more widely applicable. And HIPERLAN chips are too expensive to build into PDA's and mobile phones. More information about HIPERLAN can be found in [25].

1.4 Future Wireless Systems (MBS)

Numerous wireless systems are still being developed. These systems provide constantly higher transmission speeds. One of these systems is MBS (Mobile Broadband Systems) [13] , which defines data speeds up to 155Mb/s. However, before these chips become cheap enough to be implemented in devices, instead of Bluetooth, will take many years. And therefore it is not really an alternative.

More information about MBS can be found in [13].

1.5 Comparison of Radio Networks

Table 17: Comparison of Radio Networks

	Bluetooth	WLAN	HIPERLAN 1	HIPERLAN 2	MBS
Radio band	2.45 GHz	2.45 GHz and 5 GHz	5 GHz	5 GHz	40 and/or 60 GHz
Technology	FHSS	FHSS, DSSS	OFDM	OFDM	-
Access type	Point-to-point and point-to-multipoint	LAN Access point; CSMA; Point-to-point	LAN Access point; CSMA	LAN Access point; point-to-point	Terrestrial
Data rate (Mb/s)	1Mb/s	1,2,11,... Mb/s	1 Mb/s	20 - 50 Mb/s	Up to 155Mb/s
Range	10 m (opt 100m)	30m	800m	50m - ...	1000m
Application	Speech; data	LAN; data	LAN; data	LAN; data	LAN, HDTV, Mobile office
Remarks	Cheap; Multi-purpose	Compatible with Ethernet	Becoming obsolescence	Compatible with 3G (UMTS), ATM and IP networks	To be developed; compatible with B-ISDN

Appendix 2 The Use of Public and Private Keys in Encryption

In this appendix the basics of the usage of public and private keys are given. But first a few assumptions must be made.

- An entity its public key is known by all other entities and its private key is private.
- Information encrypted with the private key can only be decrypted with the corresponding public key and visa versa.
- One entity is the client and another is a server. This has nothing to do with the network relations, only with the offering of services.

Notations:

- $Information_{key}$ means that 'information' is encrypted with 'key'.
- TS stands for time-stamp.
- $Hash._{-1}$ is the hash of the last received message.

First a (unsecured) connection must be set up:

Client	→ request →	Server

Now the server replies with a certificate. Part of this certificate is encrypted using the servers private key. Is the client able to decrypt the certificate using the public key, then the certificate is valid and the client is sure about the server.

Client	← $certificate_{private\text{-}key}$ ←	Server

Because everybody can decrypt information encrypted using the servers private key, a new key must be created.

Client	→ $new\text{-}key_{public\text{-}key}$ →	Server

Now secure information exchange can take place, using the symmetric encryption algorithms.

Client	→ $information_{new\text{-}key}$ →	Server
Client	← $information_{new\text{-}key}$ ←	Server

To provide integrity a time-stamp is added to the information and a hash is calculated. When the information is changed, the hash will be invalid and re-transmission is impossible due to the time-stamp.

Client	→ $(info,ts,hash(info,ts))_{new\text{-}key}$ →	Server
Client	← $(info,ts,hash._{-1},hash(info,ts,hash._{-1}))_{new\text{-}key}$ ←	Server

This method is based on a Public Key Infrastructure model that uses certificates from Trusted Third Parties (TTP). A TTP distributes public keys, controls the certificates and makes sure that the owner of the certificate can be trusted. An example of a TTP which controls certificates for Internet commerce is Verisign [32]. The BSIG could act as TTP for Bluetooth security. More information about PKI architectures and certificates can be found in [33] and [32].

Appendix 3 Used HCI Functions

The used HCI functions are described in this appendix. For each HCI function there is a LMP / L2CAP counterpart.

3.1 Functions

3.1.1 Link Control Commands

Inquiry The inquiry function discovers nearby Bluetooth devices. One of the parameters is the inquiry duration.

Inquiry_Cancel When enough devices have been discovered the inquiry can be stopped with Inquiry_Cancel.

Create_Connection When the address of a nearby Bluetooth device is known, an ACL connection can be created. The connection packet type can be set with this function.

Disconnect This function is used to stop an ACL connection.

Accept_Connection_Request This function has two parameters, the first is the address from which the request came and the second is the role which the requested device wants in the relation, i.e. master or slave.

3.1.1 Link Policy Commands

Unless these functions are not needed in the scatternet formation algorithm, the may be used by other Bluetooth devices that are connected with devices in the network.

Hold_Mode With this command a connection can be put in hold mode for maximal 40.9 seconds. When a connection is in hold mode devices are free to perform other tasks.

Park_Mode With this command a connection can be parked. It has to be unparked by the master. When a connection is parked the devices are free to perform other tasks.

Exit_Park_Mode Function used by the master to exit the park mode for a certain connection.

3.1.1 Host Controller & Baseband Commands

Read_Scan_Enable Reads the current scan mode:

- No scans enabled: used by slave devices
- Page scan enabled: scan for page messages (connection request)
- Inquiry scan enabled: scan for inquiries
- Both scans enabled: used by masters

Write_Scan_Enable Writes the scan modes described above.

Read_Page_Timeout This values describes how long a device will wait for a response from a remote Bluetooth device to a connection request.

Write_Page_Timeout This function writes the page time-out.

Read_Connection_Accept_Timeout This parameter allows a device to automatically deny a connection request if not accepted within the specified period.

Write_Connection_Accept_Timeout This function writes the connection accept time-out.

3.1.1 Status Parameters

Read_RSSI Reads the received signal strength indication, which is between –128dB and 127dB.

Get_Link_Quality This function returns the link quality ranging from 0 to 255 (the higher the value, the better the link).

3.2 Events

Inquiry_Complete Occurs when the inquiry is completed and returns the number of inquiry results.

Inquiry_Result Occurs before the inquiry is completed when a nearby Bluetooth device is found. It returns the remote address and the clock offset.

Connection_Complete Occurs when a connection is created successfully, the connection handle and remote address are returned.

Connection_Request Requests a connection with a remote device. This is speeded up when the clock offset is known.

Disconnection_Complete Occurs when the disconnection was successful.

Command_Complete Occurs when a command was executed successfully. Depending on the command return parameters are present.

Command_Status Returns the status of an executed command.

Mode_Change Occurs when the mode has been changed and returns the connection handle and new mode.

Appendix 4 The Simulation Tool

In this appendix the simulation tool is described. The tool is written in Borland Delphi, because that is the available language. The program runs on both Microsoft Windows NT 4.0 and Microsoft Windows 2000 machines.

A screendump of the program is shown in figure 23

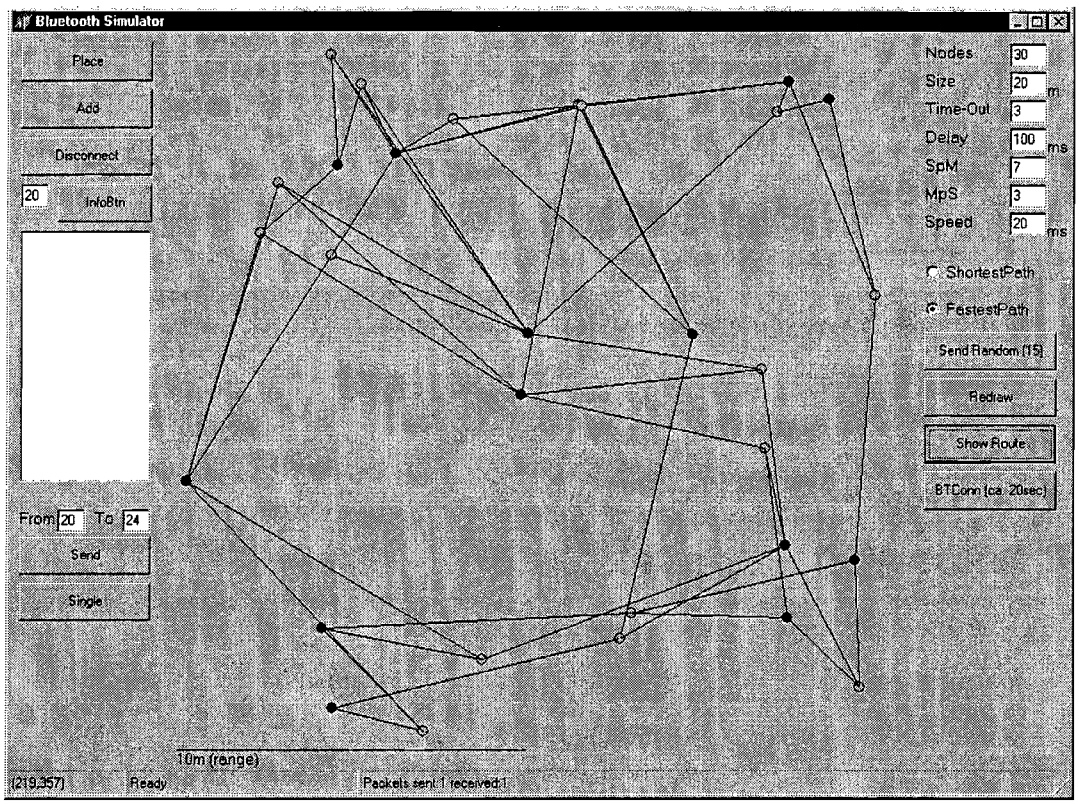

Figure 23: Screen dump simulator tool

The different measurements are stored in files. The current screen shows 30 nodes on 400m^2. The dark nodes are the masters and the lighter are the slaves. For testing the maximum number of masters per slave and slaves per master can be adjusted.

4.1 Simulation Model

The simulation tool was primarily written for scatternet formation measurements, i.e. the connectivity. The used functions are all available in the Bluetooth specification and will be implemented in future devices.

The routing method implementation is merely a functional implementation: timing and delay are difficult because the unsynchronised piconets are difficult to simulate. In real Bluetooth devices the timing specific values differ, because the implementation depends on the manufacturer.

The adding and removing of nodes is done in a Gaussian way where the mean is the number of nodes for which the experiment is done. A Gaussian distribution is shown in figure 24, with n the number of nodes.

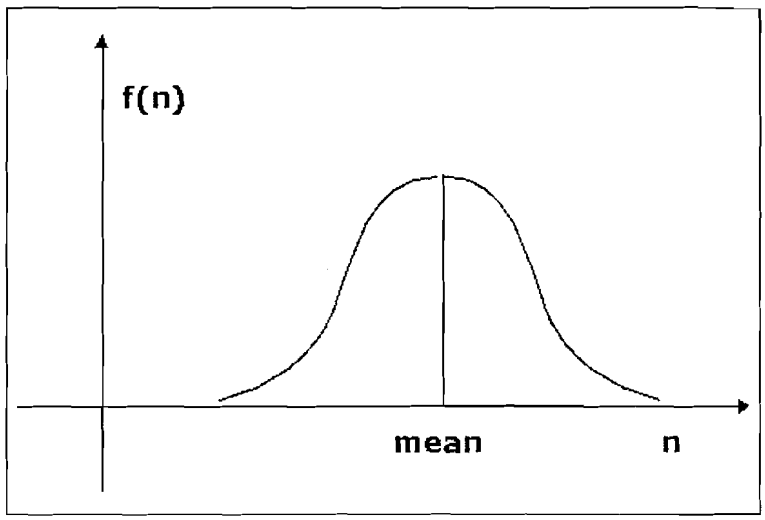

Figure 24: Gaussian distribution

4.2 Simulations

Simulations can be run for a certain amount of time, calculating or measuring the following values:

- connectivity
- master/slave ratio
- average connection per master and slave
- routes between pairs of nodes
- table entries

Throughput and delays are not simulated because the specification does not describe a good scatternet model. No MAC scheduling and delays are specified for scatternets.

Appendix 5 Piconet throughput measurements

Measurements have been done using the Digianswer Bluetooth Environment and the Digianswer Ethernet emulator. In the Bluetooth Neighbourhood TCP/IP runs over BNEP, the Bluetooth Network Encapsulation Protocol, developed by Digianswer.

Six Bluetooth PC cards were used to create a piconet (scatternets are not supported). In this piconet data was exchanged and the throughput between the devices was measured using the Digianswer Static Viewer. All traffic between two slaves, like S1→S2, goes through the master. The measurements are shown in the table below:

Table 18: Piconet Throughput

Measure-ment	Routes	Transfer rate [kB/s]
1	M1 → S1	40
2	M1 → S1 S1 → M1	17 17
3	S1 → S2	17
4	S1 → S2 S2 → S1	10 10
5	S1 → S2 S3 → S4	12 12
6	M1 → S1 S2 → S3	15 10

The connection set-up is shown is the following figure:

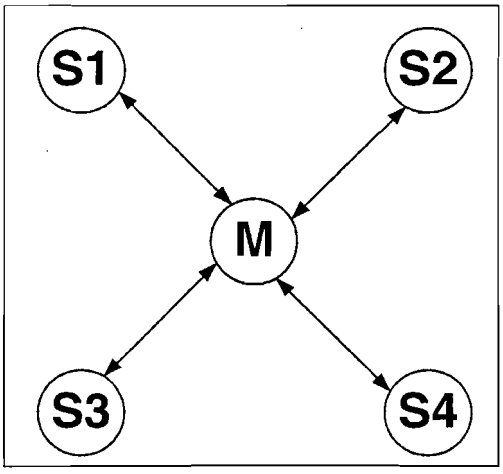

Figure 25: Piconet used in throughput measurement

The reason for the difference between measurements four and five is that in four two symmetric connections are used, where in measurement five, four asymmetric connections are used. The results show this is easier to schedule by the master and therefore the throughput is a little bit higher.

The ping delay times have been determined by measuring the delay for 50 ping messages. The measurement has been done for two cases, in the first the piconet is idle, i.e. no data is exchanged. In the second case Information was exchanged over the piconet: two slaves exchange data and one slave sends data to the master. The slave to slave ping was done from one of the sending slaves. The results are displayed in the figure below.

Table 19: Ping delay times

Route	Average ping idle [ms]	Max ping idle [ms]	Average ping busy [ms]	Max ping busy [ms]
S → M, M → S	21	67	108	268
S → S	31	75	392	1060

It was remarkable that the used packet types are DM3 and DH3 packets. Almost no DM1, DH1 and DM5 packets are used. When only one connection exists DH5 packets were used. The packet type choice is made by the Digianswer emulator.