MASTER

Modelling the MAC/DLC layer of Bluetooth High Rate in the network simulator

de Graaf, R.E.

Award date:
2003

Link to publication
Modelling the MAC/DLC layer
Of Bluetooth High Rate
in the Network Simulator

by R.E. de Graaf

Master of Science Thesis
December 2003

Supervisors:
Ir. J. van der Meer, Ericsson Telecommunicatie B.V., Rijen
Ir. R. Taori, Ericsson Telecommunicatie B.V., Rijen

Graduation professor:
Prof.ir. J. de Stigter

The Faculty of Electrical Engineering of Eindhoven University of Technology disclaims all responsibility for the contents of traineeship and graduation reports.
Preface

This master thesis is written to finish my study Information Engineering, part of the faculty of Electrical Engineering, at the Eindhoven University of Technology. The graduation project is performed at Ericsson Telecommunicatie B.V., Rijen. The master thesis project gave me the opportunity to increase my skills and knowledge on data communication, ad-hoc networking and simulation related topics. Although it was hard working to get satisfying results for the report, I have worked with lots of pleasure at Ericsson to carry out my master thesis project.

Beside my family I would like to thank some persons in particular that showed interest in and supported me during my project. First of all I would like to thank Jan van der Meer for giving me the opportunity to perform my graduation project at Ericsson Telecommunicatie B.V. Rijen and for general support. Also many thanks go out to my supervisor Rakesh Taori for giving me very good support and advices during the project. I also would like to thank Bas Jansen for providing me the basis, a Bluetooth High Rate simulator framework as the result of his internship at Ericsson, of my project and the help he gave me during the initial stage of the project.

A great thank goes out to my academic supervisor prof.ir. Johannes de Stigter at Eindhoven University of Technology / Swisscom for the feedback and support on my project and report and for his visits to Ericsson Rijen to keep up with ongoing issues and discussing progress.

Furthermore I would like to thank my department colleagues for giving me support and help, especially Tim Ehrhart for his help on UNIX and TCP/IP related questions and Jos den Hartog for his advices and interesting discussions.

Finally thanks to everybody else who showed interest in my project by providing me of valuable comments or suggestions during the project.

Rutger de Graaf

Rijen, 28 November 2003
Summary

Within the Bluetooth Special Interest Group (SIG), research is ongoing into new extensions for Bluetooth. One of these extensions is a High Rate Mode capable of providing raw data rates of up to 12 Mb/s. Goal of this study was to evaluate the TCP/IP behaviour on the performance of the MAC/DLC layer of Bluetooth High Rate. However, only a performance evaluation of the MAC/DLC layer itself could be carried out during the period of the work reported here. On the basis of the current draft specification a model describing the MAC/DLC layer of Bluetooth High Rate has been built.

The Network Simulator [7], open source software freely distributed by Berkeley University, provides a good basis for carrying out performance evaluation using the TCP/IP protocol suite. In a previous project [6], work is initiated on building a model of the MAC/DLC layer of Bluetooth High Rate that can be simulated on the Network Simulator. The work described in this report is a continuation of the work described in [6].

Bluetooth HR characteristics are studied to modify and extend the model with, among others a selective repeat mechanism to provide reliable data transfer on a non-ideal channel, correct scheduling and an error generator for generating random error patterns (modelling independent errors). By use of a script, the model can be configured easily for a particular simulation. The model is tested with respect to throughput as well latency and compared with analytical calculations.

Typical user scenarios for ad-hoc wireless applications are examined and can be categorized into two groups: point-to-point and point-to-multipoint based topologies. The current model is capable of simulating the following topologies:

- nodes with point-to-point links
- nodes that are involved in traffic with more than one node in point-to-point links

Point-to-multipoint topologies cannot be simulated using the current model. Scenarios based on above mentioned topologies are used as input for simulations in NS to measure performance of the MAC/DLC layer. Performance is measured with respect to throughput and packet loss while varying channel conditions. In addition, scenarios are set up in such a way that measurement results can be mutually compared with respect to performance.

Simulation results showed that Bluetooth HR:

- provides robust mechanisms for recovery from errors
- is capable of serving traffic with Quality of Service (QoS) requirements and Best Effort traffic at the same time, without significant loss of performance with respect to aggregate throughput
- protocol overhead is smaller compared to WLAN 802.11b
# Table Of Contents

1 PROBLEM DEFINITION............................................................................................................. 9

   ABSTRACT .......................................................................................................................... 9
   1.1 INTRODUCTION ............................................................................................................ 9
   1.2 RESEARCH SCOPE ....................................................................................................... 9
   1.3 RESEARCH QUESTION ............................................................................................... 10
   1.4 RESEARCH METHOD ................................................................................................. 10
   1.5 ORGANIZATION OF THIS REPORT ........................................................................... 11
   1.6 WHAT HAS BEEN ACHIEVED? .................................................................................. 12

2 AN INTRODUCTION TO BLUETOOTH HIGH RATE ............................................................ 13

   ABSTRACT .......................................................................................................................... 13
   2.1 INTRODUCTION ............................................................................................................ 13
   2.2 THE RADIO INTERFACE .......................................................................................... 13
   2.3 NETWORK TOPOLOGY .............................................................................................. 14
      2.3.1 Addressing and connections .............................................................................. 14
      2.3.2 Casting ............................................................................................................... 15
   2.4 PACKETS .................................................................................................................. 16
      2.4.1 Segmentation ................................................................................................     17
   2.5 MEDIUM ACCESS CONTROL (MAC) ...................................................................... 17
      2.5.1 Token scheduling ............................................................................................. 19
   2.6 ERROR HANDLING MECHANISMS ........................................................................... 22
   2.7 CONCLUSION ............................................................................................................ 22

3 SIMULATION ENVIRONMENT (NETWORK SIMULATOR AND HIGH RATE MODEL) .......... 23

   ABSTRACT .......................................................................................................................... 23
   3.1 INTRODUCTION ............................................................................................................ 23
   3.2 WIRELESS NODE ARCHITECTURE ACCORDING TO [6] ........................................... 24
   3.3 MODIFICATIONS TO THE WIRELESS NODE ARCHITECTURE ............................. 25
   3.4 EXTENSION OF THE WIRELESS NODE ARCHITECTURE ........................................ 29
   3.5 FIRST RESULTS ........................................................................................................ 30
      3.5.1 Throughput calculations ................................................................................. 30
      3.5.2 Measured throughput ..................................................................................... 32
      3.5.3 Latency calculations ....................................................................................... 33
      3.5.4 Measured latency ........................................................................................... 33
   3.6 CONCLUSIONS .......................................................................................................... 35

4 SIMULATION SET-UP ........................................................................................................... 37

   ABSTRACT .......................................................................................................................... 37
   4.1 WHAT IS SIMULATED ................................................................................................ 37
      4.1.1 Channel conditions .......................................................................................... 37
      4.1.2 Traffic sources ............................................................................................... 38
      4.1.3 Differences between the draft specification and the NS model ...................... 39
      4.1.4 Simulation set-up ............................................................................................ 40
   4.2 WHAT IS MEASURED ................................................................................................ 40
   4.3 CONCLUSIONS .......................................................................................................... 41
1 Problem Definition

Abstract
This chapter presents the problem definition for this research project. Section 1.1 of this chapter will give a brief background on Bluetooth and in particular the new High Rate extension of Bluetooth. In section 1.2 the restrictions/limitations to the research project are discussed. Subsequently the research question together with related sub topics are formulated, followed by a step-by-step overview of the research method. Section 1.5 presents the organization of this report. Finally, in section 1.6 the achievements of this project are given.

1.1 Introduction
Bluetooth is a wireless short-range connectivity solution for portable devices as PDA's, mobile phones and laptops [1]. This technology is primarily targeted for replacing the interconnect cables between portable devices. Characteristics of the Bluetooth technology are its universal air interface, low cost and low power consumption. Bluetooth technology also enables ad-hoc networking among portable devices. The operation range of Bluetooth can vary from several meters up to 100 meters. Normal operation however takes place within 10 meters.

The Bluetooth Special Interest Group (SIG) was founded to gain worldwide support/interest for the Bluetooth technology. The SIG is responsible for carrying out the development of the technology by developing an open industry specification. First specification of the Bluetooth standard was released by the SIG in 1999 [4]. This standard was later refined and released as version 1.1 [5]. Version 1.1 of the specification will in this report be referred to as Bluetooth Basic Rate (Bluetooth BR). Bluetooth Basic Rate is now seeing an increased use in consumer applications. Some very common products are a wireless headset to connect to a mobile phone or a USB plug to connect a laptop PC to a PDA for instance.

As Bluetooth BR gets widely used, and as higher data rate applications (e.g. video streaming) emerge, the need for increased bandwidth is felt. To address this need, research and standardization is ongoing into extensions for Bluetooth, one of these extensions concerns higher data rate Bluetooth. Referred to as Bluetooth HR in this report. Where Bluetooth Basic Rate utilises a gross data rate of 1 Mb/s, Bluetooth HR targets a data rate of up to a maximum of 12 Mb/s.

1.2 Research scope
For Bluetooth HR several protocol layers are defined; A physical layer in which the radio interface is defined. A Medium Access Control/Data Link Control (MAC/DLC) layer where medium access and data link control mechanisms are defined. A Link Manager, which is used for link control and configuration. A Logical Link Control and Adaptation Protocol (L2CAP), which establishes virtual channels between hosts and provides for
segmentation of user data into Bluetooth transmission supported data sizes. Finally, several profiles are defined (e.g. headset for communication between wireless headset and mobile phone and RFCOMM for serial communication).

A basis for this High Rate extension is already made in a draft specification for the physical layer [3] and the MAC/DLC layer [2]. In [2] the MAC/DLC layer is referred to as baseband.

The work presented in this report is limited to the MAC/DLC layer of Bluetooth HR. In particular, the project is focused on measuring the performance of the MAC/DLC layer for TCP/IP traffic. The mechanisms for medium access and data link control, as specified in a draft specification [2] are used as the basis for this work.

1.3 Research question

The formulation of the research question is as follows:

What is the performance of the MAC/DLC layer of the HR mode of Bluetooth for TCP/IP traffic, measured for different scenarios under different radio conditions, and how does it compare to that of Wireless LAN 802.11b?

As a basis for this study it was decided a priori that the Network Simulator be used. Given this input, the task was subdivided as follows:

“Model the behaviour of the High Rate extension of Bluetooth suitable for use in the NS (Network Simulator freely distributed by Berkeley University) software package.

On the basis of this model, perform a well-defined simulation study so as to measure the performance of the MAC/DLC layer of the High Rate mode of Bluetooth.”

Several sub topics are formulated which are related to the research topic and will contribute to the solution of the research topic:

Sub topics:
1. What are the characteristics of the Bluetooth High Rate extension?
2. How should the High Rate extension be modelled?
3. Which simulation methods are used for doing performance evaluation of ad-hoc networks?
4. What are typical metrics used for measuring performance?
5. What are typical user scenarios for short-range ad-hoc networks?
6. Which factors in Bluetooth High Rate cause delay in transmission times?
7. Which typical physical limitations cause interference in radio traffic?
8. How should non-idealities like fading and shadowing be modelled within the framework of NS?
9. Which protocol parameters influence throughput?
10. Which protocol parameters have to be optimised to increase throughput?
11. What are the differences in throughput when comparing Bluetooth High Rate with other ad-hoc networking techniques?

1.4 Research method

The following steps are defined which will contribute to structure carrying out the research project. Subsequently each separate step will be described in more detail.
Chapter 1: Problem definition

Initial research will concentrate on studying the following topics:
1. Bluetooth
2. NS (Network Simulator)

Next the performance evaluation will be executed in a step-by-step process:
3. Research and study of to be modelled behaviour
4. Modelling the behaviour
5. Simulating and measuring throughput and delay
6. Optimising parameters
7. Analysis and observations of simulation

Steps 3 to 7 above can be iterative executed for several scenario's/applications.
8. Conclude and recommend

1) Bluetooth
   This study will focus on the characteristics of Bluetooth High Rate and some
   main differences it has with Bluetooth Basic Rate. Chapter 2 of this report will
   describe the general operation of Bluetooth High Rate.

2) NS
   In order to make extensions to the simulator a good understanding of the NS
   architecture is necessary. This will also contribute to topics as interpreting
   simulation results.

3) Research and study of to be modelled behaviour
   In the first stage of the actual performance evaluation, in-depth research of a
   certain non-ideal character has to be carried out. The results of this study will
   serve as an input for the next step: modelling of the behaviour.

4) Modelling the behaviour
   When detailed character of a certain non-ideal behaviour is in place, the next
   step will be to model the behaviour in separate modules.

5) Simulating and measuring throughput and delay
   Output of the previous stage will be used to measure throughput and delay on a
   Bluetooth High Rate channel under the non-ideal conditions that is modelled.

6) Optimising parameters
   Parameters concerning the MAC/DLC layer protocol and higher layer protocols
   may have to be adapted to optimise the throughput under certain channel
   conditions.

7) Analysis and observations of simulation
   The last stage of the iterative performance evaluation consists of analysing the
   simulation results and report observations made during the analysis.

8) Conclude and recommend
   Writing the conclusion and recommendations in this report will be the last step of
   this research project.

1.5 Organization of this report

Next chapter presents an outline of the Bluetooth HR system. In chapter 3 modelling of
Bluetooth HR is discussed. Chapter 4 addresses topics about simulation set-up. Typical
user scenarios used for simulations can be found in chapter 5. In chapter 6
measurement results, obtained by simulations, are presented and analysed. Finally,
conclusions and recommendations are given in chapter 7.
1.6 *What has been achieved?*

1. A model according to the Bluetooth HR draft specification has been built. However some procedures (e.g. contention resolution) have been simplified and therefore they differ from the draft specification. More details about the differences can be found in chapter 3.

2. The model is capable of simulating the following topologies.
   - Nodes with point-to-point links
   - Nodes that are involved in traffic with more than one node in point-to-point links

3. The performance of the MAC/DLC layer is measured for typical user scenarios based on topologies as stated in 2.

4. Multicast for point-to-multipoint links could not be implemented

5. Contrary to original planning, the influence of the TCP/IP behaviour itself (e.g. TCP back off behaviour) could not be examined. However the overhead due to TCP/IP protocol has been taken into account for simulations.

6. A comparison with WLAN 802.11b as planned could not be made. Only preliminary conclusions have been drawn from existing literature (see 6.3).
2 An introduction to Bluetooth High Rate

Abstract
In this chapter an outline of the Bluetooth High Rate system is provided. A more detailed explanation of Bluetooth Basic Rate can be found in [5]. The overview that is given in this chapter is not a full specification of Bluetooth High Rate. A draft specification of the MAC/DLC layer of Bluetooth HR is given in [2].

2.1 Introduction
In the following section the radio interface used in Bluetooth is discussed in short. Followed by an outline of the network topology architecture that is proposed for Bluetooth High Rate. Section 2.4 discusses the packet format that is used in Bluetooth High Rate. Channel access is explained in section 2.5. Finally in section 2.6 two error-handling mechanisms are explained that are used in Bluetooth HR.

2.2 The radio interface
Communications in Bluetooth take place in the unlicensed 2.4 GHz ISM (Industrial, Scientific and Medical) band. Bluetooth BR deploys frequency hopping (FH). Using the frequency hopping technique the 2.4GHz band is divided into 79 1-MHz subchannels. Radio transceivers send and receive data over a sequence of the subchannels using a hopping pattern. The subchannels are centred at \((2.402 + k)\) MHz, \(k = 0,1,\ldots,78\). During a connection the radio transceivers hop from carrier to carrier in a pseudo-random fashion, this results in a FH channel. Units participating in the FH channel can negotiate a High Rate channel. In the Bluetooth High Rate extension 70 stationary RF channels are defined. A channel in Bluetooth HR occupies 4 MHz bandwidth. The channels are centred at \((2.4075 + k)\) MHz, \(k = 0,1,\ldots,69\). The channel positioning of FH and HR channels is shown in Figure 2.1. For allowing multiple participants on the High Rate channel Time Division Duplexing (TDD) is applied.

The following modulation formats are provided in Bluetooth HR:
- DBPSK (4 Mb/s)
- DQPSK (8 Mb/s)
- 8-DPSK (12 Mb/s)

Figure 2.1: Channel positioning
To allow the coexistence of a High Rate channel and the FH channel the FH sequence is adapted from 79 to 75 carriers to open up a window of 4 MHz in the 80 MHz FH spectrum. Adaptive Frequency Hopping enables communications to take place between BR units on the FH channel and units on the HR channel without mutual interference.

2.3 **Network Topology**

In the previous section the FH channel is discussed. This is the basis for an ad-hoc Bluetooth network. A collection of units that participate in a FH channel and thus are able to communicate with each other forms a piconet. Access to the channel is regulated by one device called the piconet master. The piconet contains also at most 7 slave devices with which the piconet master is actively involved in communications. A device can be allocated as slave or master. Multiple piconets can coexist at the same time.

The Bluetooth High Rate extension makes it possible for devices in a piconet, that have high data rate capabilities, to negotiate a High Rate channel. One device in this channel is assigned a function as HR supervisor. Unlike the master in the FH channel, the HR supervisor does not regulate access to the channel. The role the supervisor undertakes is explained in section 2.4. Several High Rate channels can be set up within a piconet. In Figure 2.2 a FH channel, which contains two High Rate channels, is represented.

![Figure 2.2: FH channel with two HR subnets](image)

Units in control by different supervisors, hence participating on different HR subnets, are not able to communicate with each other. In addition, the piconet master does not control units in a HR subnet. In case of poor channel conditions, units in the HR channel can fall back into the FH channel. Because of the fact that the timing of the FH channel is aligned with the HR channel.

2.3.1 **Addressing and connections**

*Logical points*, point addresses coupled to a physical unit (or units in case of multicast and broadcast) are used in Bluetooth HR to address a specific source or destination.
Chapter 2: An introduction to Bluetooth High Rate

Logical Points

Logical Control Points (LCP)

Device Control Point (DCP)

Broadcast Control Point (BCP)

Logical Traffic Points (LTP)

Unicast Traffic Point (UTP)

Multicast Traffic Point (MTP)

Figure 2.3: Logical points in Bluetooth HR mode

The logical points are divided into two groups, i.e. control and traffic points as are represented in Figure 2.3. This way control traffic is segregated from data traffic to avoid the situation that over-full data buffers block control traffic.

When a unit is injected into the Bluetooth HR channel, a Device Control Point is assigned to it. The DCP is used for all device related control information and is allocated by the piconet master. In addition, each unit has a Broadcast Control Point that is also used for network related control information. Besides the DCP and BCP, the units active in transmissions can have one (or multiple) unicast traffic points and multicast traffic points. The HR supervisor carries out allocating the traffic point addresses.

Connections between logical points are called logical links. A distinction can be made between logical control links and logical traffic links. Logical control links can be set up between two DCP’s or between a DCP and a BCP. On the other hand logical traffic links can be set up between two UTP’s or between a UTP and a MTP. The principle of unicast and multicast will be explained in the next sub-section.

2.3.2 Casting

Bluetooth HR supports unicast, multicast and broadcast. Unicast means data communication between a single sending node and a single receiving node (point-to-point), whereas multicast means data communication between a single sending node and multiple receiving nodes (point-to-multipoint). In the last mentioned case each receiving node gets assigned the same multicast point address. Finally, broadcast is intended for transmitting network related control information (not data) to all nodes participating on the HR channel.

Multicast and broadcast can both be used as acknowledged or unacknowledged service. In Figure 2.4 one of the HR subnets from Figure 2.2 is enlarged to illustrate the logical points and logical links.

Figure 2.4 also shows the unicast and multicast principle. A logical traffic link can be established between as well two unicast traffic points (UTP) as one UTP and multiple Multicast Traffic Points (MTP).

The dotted arrow in the figure below illustrates a reverse link. QoS in the forward and reverse link can defer from each other.
2.4 Packets

In Bluetooth BR various packet types are defined whereas in Bluetooth HR only one packet type is defined. Packets used on the HR channel have the format as shown in Figure 2.5. The HR packet format consists of 5 fields. The preamble and sync field are used for synchronisation purposes. The high rate packet header contains destination and the source address, a number of fields regarding packet segmentation, flow control, modulation type of payload, length of payload and a CRC field for error detection. User information is packed within the payload and if sufficiently large divided into segments. More about segmenting of the payload is provided in the following sub-section. The HR packet is terminated with a trailer.

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Sync</th>
<th>Header</th>
<th>Payload</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 B</td>
<td>2 B</td>
<td>11 B</td>
<td>0-4095 B</td>
<td>2/4/6 bits</td>
</tr>
</tbody>
</table>

**Figure 2.5: High Rate packet format**

DBPSK modulation is used for the first three fields of the packet because of its robustness. The payload can be modulated with DBPSK as well DQPSK and 8-DPSK. The trailer uses the same modulation as the payload. When no payload is provided in the packet the DBPSK method is used for the trailer.
2.4.1 Segmentation

As is shown in Figure 2.5 the payload of a HR packet can have a size of up to 4095 bytes. To prevent from retransmitting a complete packet in case of partial errors in the payload, segmentation is applied in Bluetooth HR to make selective repeat of erroneous received segments possible.

When a higher layer packet provides more than 128 bytes of user information to the baseband, the payload is segmented in 128 byte chunks. Each segment is provided with an additional sequence number field and a CRC field. In Figure 2.6 is shown how segmenting takes place and how baseband HR packets are derived of the higher layer datagrams. Segments in a HR packet must belong to the same higher layer datagram.

As can be seen from Figure 2.6 all payload segments, except the last segment, contain 128 B of user information.

![Figure 2.6: Payload segmenting](image)

2.5 Medium Access Control (MAC)

In this chapter the methods that are used for Medium Access Control in Bluetooth HR are presented. In Bluetooth BR traffic is regulated by the piconet master, which means that the master is actively involved in each transmission. Bluetooth HR however, enables units to communicate directly with each other without involving the master. Therefore a distributed control protocol based on a ping-pong protocol is introduced for Bluetooth HR. The ping-pong protocol is based on the following principle: When a node receives a packet it also acquires the token. The token allows only this unit to communicate directly with other nodes on the channel on peer-to-peer basis without involving the supervisor. A supervisor device is assigned in each High Rate channel. The supervisor is required to be in radio range of all devices participating on the High Rate channel. In this way it can broadcast changes in the channel configuration to all units on the HR channel. A priority scheme is added to the protocol to allow for services other than best effort. Using the priority scheme units can get periodically access to the channel by means of so called priority slots (PS). Priority slots are HR timeslots that are assigned to a traffic
point of a node that has requested a certain level of priority. At least the supervisor has been assigned supervisor priority slots (SPS) whereas more nodes may be assigned priority slots. The SPS provides the supervisor node unconditional access to the channel. The supervisor uses the supervisor priority slots primarily for token distribution to provide all nodes on the HR channel fair access to the channel and prevent starvation. The latter is provided by applying Round Robin to the token distribution, so each node will get periodic access to the channel. Additionally an algorithm referred to as sticky-token is introduced. This algorithm causes the token to stick between a unit pair until all traffic between the pair is exhausted or is interrupted by a PS. In the figure below the principle of the ping pong protocol is illustrated.

![Ping-pong protocol](image)

**Figure 2.7: Ping-pong protocol**

Node A sends a packet to node B, which will send because of the sticky token protocol and if it has any data pending for node A, a packet back to node A. At a certain time the data between node A and node B is exhausted and Node A will send data to node C. Subsequently node C issues a packet for node A. This ping-pong traffic will go on till nodes have no more data to serve.

![Ping-Pong protocol with priority scheme](image)

**Figure 2.8: Ping-Pong protocol with priority scheme**

Nodes with QoS requirements on one of their outgoing traffic links can reserve bandwidth on the HR channel by allocating priority slots to one of their traffic points. The priority slot provides a node (or to be precise a traffic point) with unconditional access to the channel irrespective of where the token resided before the current slot. Note that an approaching priority slot can interrupt the sticky-token algorithm. In Figure 2.8 the ping-pong protocol is shown in the case priority slots are applied for unit C. As can be seen
from this figure at some moment a packet got lost. This results in *dead time* (or idle time) on the channel since none of the nodes on the channel acquired the token. Priority slots provide token recovery on the channel in case a packet, which implies a token, got lost. The priority slots also provide timing synchronization. Since the location of the priority slots are known to all units on the channel, this can be used for synchronization.

### 2.5.1 Token scheduling

In this sub-section the token passing scheme will be explained in detail. There are two ways in particular for a node to acquire the token. The first is provided by means of priority slots. Secondly the token can be acquired in between any priority slot by receiving a packet on a traffic point or a control point.

Before going into detail two terms have to be defined. In the following will be referred to the term *waiting data* when a node has data, which is waiting to be transmitted, in one of its traffic point buffers. Second term used is *broadcast inquiry*. A broadcast inquiry is a header only packet that is broadcasted to all nodes on the HR channel. The nodes that are synchronized to the HR channel will respond this inquiry with an acknowledge. Additionally, nodes can indicate in the acknowledge if they have data waiting for any other node. From the received acknowledges the node that issued the inquiry can derive which nodes have waiting data and are synchronized to the channel.

Initially, the token can be acquired via a supervisor priority slot or a priority slot that is allocated for a particular node. What will happen after a priority slot took place? To answer this question the token passing schedule is shown in Figure 2.9. When a node gets access to the channel via a priority slot this node will look for waiting data on the traffic point belonging to the priority slot. If data is present, data transmission can start. When there is no data present, other traffic points on this node will be examined for waiting data as sticky token is not applied at a priority slot. If these have data, a random traffic point will be selected where traffic points, which serve priority traffic, have precedence above others. After the traffic point selection the transmission starts. In case no traffic point has data, broadcast inquiry will be applied. After broadcast inquiry the token will be distributed to a node, which is selected in a random fashion out of the nodes that requested the token. When no requests for the token were received the token is killed, introducing dead time on the channel.

The token scheduling for supervisor priority slots differs at one point from the priority slot schedule, token distribution is carried out instead of serving the traffic point that belongs to the PS. If the supervisor node is in turn in the Round Robin scheme the priority slot scheme will be resumed at the point that other traffic points on this node will be examined for waiting data. In case the supervisor node is not in turn, the token is passed to the node that is in turn in the round robin scheme. After token passing the token is received on the device control point (DCP) of the receiving node.

The schedule of receiving a token on a DCP is shown on the right side in Figure 2.10. In this case sticky token is applied, which means that data waiting on this node for the node from which the token is received has to be served first. If no data is waiting to be served to that node Broadcast Inquiry is applied. After the broadcast inquiry will first be checked if a randomly selected traffic point on the inquiring node has data to serve to a synchronized node. Where traffic points serving priority traffic, have precedence above other traffic points. Transmission starts if data could be served to a synchronized node. If that is not the case the token will be distributed to a random selected node having data to serve. In case there are no nodes having data to serve the token is killed.
Finally when data transmission is started, HR packets with or without data (baseband acknowledges) will arrive at a node on a traffic point. In this case also sticky token is applied, so the receiving traffic point will examine if it has data for the reverse link. If no data is waiting to be served on the current traffic point, other traffic points on the current node with a link to the sender of the data packet will be examined for data. The token is passed to a random traffic point that has data to serve, where traffic points serving

![Diagram](image-url)

Figure 2.9: Token passing after priority slot
priority traffic, have precedence above other traffic points. If no data is present for the node from which the last packet (and thus the token) is received sticky token will be maintained so the token is sent back. Sticky token stops in case no data is present for the node from which the last packet is received and the received packet contained no data, the sender has thus indicated that its buffers are exhausted for the links between the sticky token node pair. Broadcast inquiry will now be applied to check if other nodes need channel access. The procedure for token handling after broadcast inquiry is identical to what is just described for the schedule of receiving a token on a DCP.

Figure 2.10: Token pass to traffic point or device control point
Chapter 2: An introduction to Bluetooth High Rate

2.6 Error handling mechanisms

In Bluetooth HR two error-handling mechanisms are provided. Before presenting these mechanisms error detection in Bluetooth HR will be discussed. Error detection is provided by means of CRC (Cyclic Redundancy Check) and takes place after each packet reception. When a node in Bluetooth HR receives a data packet it calculates the CRC of the header of the HR packet and the CRC of each particular segment. When this CRC corresponds to the CRC field of respectively packet header or segment, the packet or segment is received correctly. Retransmission of data only takes place in case the packet was sent on an acknowledged (or reliable) link. On an acknowledged link each transmitted segment belonging to a particular data packet is acknowledged by the receiving node. Segments that are not received correctly are requested for retransmit by the receiving node. If a header error occurred all segments of the particular packet need to be retransmitted. This mechanism is called Selective Automatic Repeat Request (sARQ).

Using SARQ, in addition to a reliable transport layer protocol (for example TCP), the data link layer (MAC/DLC) can also provide a reliable byte stream service. This means that whenever a node has received a corrupted segment belonging to an incoming HR packet, this node will request the transmitter to retransmit the concerning segment. The transmitter will only retransmit a segment in case it is explicitly requested by the receiving node. In case that a header error occurs the packet will be dropped and the token will be lost. The receiver has no knowledge of this packet, since it is dropped and so it will not issue an acknowledge. After some time the transmitting node will again possess the token because of token distribution or a priority slot. The transmitter will continue to send waiting data from the same datagram or in case all data is already sent, issue a request for an acknowledge to the receiving node since it has not received an acknowledge corresponding to the last send packet. Retransmission of data can start after reception of the corresponding acknowledge. More information about selective repeat mechanisms can be found in [8].

The second error handling mechanism provided in Bluetooth HR is Forward Error Correction (FEC). This is not discussed further in this report.

2.7 Conclusion

The outline that is given in this chapter describes the characteristics of the Bluetooth High rate extension. Topics as network topology, channel access and packet format, which are needed to understand the Bluetooth HR principles are explained. On the basis of this phase; understanding the Bluetooth High Rate draft specification, the model can be implemented to the needs of the performance analysis.
3 Simulation environment (Network Simulator and High Rate model)

Abstract
Simulations were carried out using the Network Simulator (NS). The Network Simulator [7], open source software freely distributed by Berkeley University, provides a good way to carry out performance evaluation using the TCP/IP protocol suite. In this chapter the modelling of the MAC/DLC behaviour in NS is described. This project continues the work reported in [6], with a model which is modified with respect to among others the error handling mechanism, scheduler for token passing and extended with an error generator as described in section 3.2 to 3.4. The modified model is tested and compared with calculations.

3.1 Introduction
Modelling the Bluetooth HR behaviour in NS was initiated in a preceding project. The outcome of this project was reported in [6]. Reasons why NS was chosen:

- Open source software
- Simulation code in C++
- Simulation script in OTCL
- Used TCP/IP implementation
- Widely used in industry
- Wireless LAN implementation

NS is a discrete event driven simulator targeted at networking research. The network simulator provides a wireless node architecture and substantial support for TCP/IP. In addition, NS is object oriented which makes it easy to define new behaviour to certain layers of the provided wireless protocol stack. Describing the behaviour of the MAC/DLC layer of Bluetooth High Rate in the wireless node architecture provides a good way to simulate TCP/IP traffic over Bluetooth HR. Using a model also the influence of protocol parameters on data traffic can be measured and optimised.

The result of [6] is a model describing the MAC/DLC layer of Bluetooth HR, which replaces the default WLAN 802.11b MAC layer in the wireless node architecture of NS. In the following section the wireless node architecture according to [6] will be explained. The model had to be modified and extended to be able to evaluate the performance of the MAC/DLC layer of Bluetooth HR. The following functionality was already provided in [6]:

- Timing aspects
- Priority slots
- Addressing
- Segmentation of data
- Segment level buffers
- Error detection
- Supervisor node
- Acknowledged unicast and unacknowledged multicast/broadcast
- 3 modulation types

Also some extensions were made into the NS architecture to provide for, among others, queue handling and supervisor node. Using this Bluetooth HR model, a two-node scenario can be simulated in ideal conditions. More information about the
implementation aspects can be found in [6]. In the following this model will be referred to as the initial model.

Following modifications were made to the initial model during the course of this project, because some functionality was incomplete, not according to the specification, not provided or erroneous:

- sARQ functionality
- Broadcast inquiry functionality
- Sticky token functionality
- Supervisor functionality
- Usage of multiple links connected to one node
- Enhanced simulation time calculation
- Scheduler for token passing
- Increased capacity of the buffers on a particular node
- Topology set up using script
- TCP/IP datagram header size setting using script
- Provision for datagram delay measurement
- Unreliable service functionality
- Enhanced Quality of Service for time critical traffic
- Memory leakage problem solved
- Problem with datagram drops in queue solved

An extension has been made to the wireless node architecture to provide for error generation.

In section 3.3 and 3.4 the modifications and extensions are discussed in detail. Section 3.5 describes some test results of the modified and extended model.

### 3.2 Wireless node architecture according to [6]

The NS simulator provides support for simulating wireless networks [7]. This is used as input for the work that is carried out in [6]. In the figure below the protocol stack of one wireless node, including the extensions made in the preceding project is shown.

The protocol stack is composed out of 4 layers; the Net interface or physical layer, the MAC layer, the Interface Queue and the Logical Link layer. In addition, three separate modules are defined that are used at the NetIf, MAC and LL layer. Note that the Interface Queue is only used for datagrams going down the protocol stack.

Implementation of the Bluetooth HR functionality takes place in the modules that are shown dark-shaded in Figure 3.1, i.e. the Mac layer and the Bluetooth HR Supervisor. To adapt the model to specific simulation needs, the interface queue, the MAC layer, the Bluetooth HR supervisor and the physical layer can be configured with a script that serves as input for the Network Simulator. More about the script can be found in chapter 5.

The Network Simulator currently produces four output files during simulations. The first, called *trace file*, records each individual packet as it arrives or departs at a particular layer. The MAC layer generates three output files. Each specific action, such as segmentation of data and transmitting, receiving and buffering of packets, is recorded and written to a log file by the MAC layer. In the second and third output file that is
written by the MAC layer respectively datagram delays and expired datagrams of a specified link are recorded.

![Diagram of protocol stack wireless node]

**3.3 Modifications to the wireless node architecture**

The model in the previous section is used as basis for this project. With this model initial throughput measurements were carried out, using a topology of two nodes and one logical link on which an FTP session takes place. Not taken into account in these simulations is the fact that errors could appear on the traffic link due to non-ideal conditions.

To provide for a model that is capable of simulating traffic between multiple nodes in a non-ideal Bluetooth High Rate piconet a number of modifications and extensions had to be made to the model.

Below a list of added features or modifications is shown. For each feature is explained why it was needed to add it to the High Rate model.
• Selective Automatic Repeat Request (sARQ) functionality
  The first modification comprises an error handling mechanism called Selective
  Automatic Repeat Request (sARQ) as discussed in 2.6. When a reliable link is
  required, the sARQ mechanism can be enabled.
• Broadcast Inquiry
  This modification comprehends the ability of a node to use acknowledged
  broadcast as discussed in 2.5.1.
• Supervisor function (SPS, round robin token distribution)
  A provision is made so that one of the nodes in the network topology can be
  assigned the supervisor function, also in case this node already got assigned
  another priority slot. An algorithm is added to the model to allow for multiple
  priority slots assigned to one node.
• Error generator
  An extension has been made to the wireless node architecture to provide for the
  ability to generate segment and header errors. A header error implies a packet
  drop, whereas a segment error implies a corrupted segment. More about the
  implementation of the error generator can be found in section 3.4.
• Usage of multiple links connected to one node
  The initial model did not provide for the simulation of a topology in which one
  node is connected to multiple (separate) links. An incoming control packet
  implied the reception of a token and the traffic point with the highest number
  would get possession of the token. This means that starvation could occur in
  case there were other traffic points on the same node having data to serve. A
  modification to the model is made to provide for fairness in token passing (see
  also 2.5.1).
• Enhanced simulation time calculation
  When a node gets a token on exactly a time slot boundary it would in the original
  model transmit data on the next time slot boundary. Since a node is allowed to
  transmit data from a time slot boundary, the initial implementation was at this
  point not very efficient with time slot usage. Other enhancements are made with
  respect to switching time. In the initial implementation it was possible to send
  data if there were 2 time slots free before the next upcoming priority slot. As the
  minimal packet duration is 4 time slots, this would result in contention. Finally,
  switching time was also reserved just after a priority slot while the specification
  allows for data transmission directly from a priority slot. This means that per
  priority slot 2 Bluetooth HR timeslots are unnecessary accounted which reduces
  the transmission time and throughput.
• Sticky token functionality
  When two nodes, participating on the HR channel, are exchanging traffic the
  sticky token functionality is provided so that the token sticks between these two
  nodes. When this link is exhausted, i.e. both nodes have no data for each other,
  the token will be passed to another traffic point that has data to serve on the
  node where the sticky token transfer ended. A broadcast inquiry will be executed
  to check if the receiving node is still synchronized. If no other traffic point on the
  inquiring node has data to serve, the token will be passed to another node that
  requested the token during the broadcast inquiry.
• Scheduler for token passing
  The scheduler is implemented as presented in the flow charts in chapter 2. As
  the initial model was based on a two node scenario the scheduler had to be
adapted to allow for a correct traffic flow according to the Bluetooth HR specification when simulating more than 2 nodes on a Bluetooth HR channel.

- Increased capacity of the buffers on a particular node by using extended datagram numbers.
  In case a node has more than one traffic point and sticky token is used for data transmission the following situation could occur. At some time a node with multiple traffic points and an equal amount of links to other nodes receives a data packet. Due to the sticky token protocol this node will look in its buffers for data for the reverse link. If the buffers are empty it will get a datagram, if any, from the queue. And here a potential problem rises. If the datagram is destined for another traffic link thus another traffic point, it will go into the outgoing buffer of the respective traffic point. When this process is repeated it is possible that more than two datagrams will at some time be stored in an outgoing buffer. And this is not possible because the initial model only takes into account a one-bit datagram number, which only allows two datagrams in the buffer. Extending the number of bits, while still using the least significant bit as the payload datagram number in the packet header and retaining the right FIFO order in the buffers makes the modification.

- Topology set up using script
  The initial model uses hard coded topologies, which means that for each scenario the code has to be changed and re-compiled. For convenience, there is now a possibility to define the topology of a test in the script file that serves as input for the Network Simulator.

- Datagram header size inconsistency
  In NS the default header sizes are not implemented very consistent. For UDP no header size is taken into account and using TCP, the normal IP header size of 20 bytes is taken twice in account. Therefore some changes are made into NS that allows a user of the simulator to specify the header size of the UDP and TCP protocol in the script used for executing NS. Default, NS will maintain its old behaviour.

- Memory leakage
  During some initial tests with more than two nodes on the Bluetooth HR channel short simulation times (5-6 seconds), i.e. the time a scenario is run on the simulator, were experienced due to very greedy consuming of memory by the network simulator. The first optimisation of memory usage consisted of destruction of packets in the MAC/DLC layer of the HR model, which was not taken into account in the initial model. With this optimisation there seemed to be no restrictions on memory any longer in case only two nodes on the High Rate channel were simulated. Studying the physical layer implementation of the network simulator gave insight about the memory leakage that is experienced when simulating with more than two nodes. The extensive memory consuming can be explained by the following. Each node participating on the wireless channel receives a copy of a packet that is transmitted by a particular node on this channel while the initial implementation did not account for copies of packets. The reason why it is implemented this way is that each receiver evaluates if the packet could be received by this node or not considering the geographical distance between nodes. If a node cannot receive a packet due to its geographical location, this node should destruct the packet. In the initial implementation the receiving node ignores (not destructs) the packet if the packet is not destined for this node. This resulted in exponential memory
usage when more than two nodes were simulated. With the destruction of packets that are ignored by a wireless node, the simulation time for more than two nodes can be increased significantly.

- Datagram delay measurement
  In order to calculate the datagram delay, an adaptation has been made to the NS interface queue. When a datagram is generated and enters the interface queue of a sending device, a time stamp ($T_{time\_stamp}$) is set in the Bluetooth packet header. At the moment ($T_{RX}$) that a receiving device received all packets on the MAC/DLC layer belonging to a particular datagram and will pass the datagram to an upper layer, the delay is calculated by: $T_{RX} - T_{time\_stamp}$.

- Unreliable service
  As described earlier, a reliable service can be realized with sARQ. Reliable service can result in contention when serving time critical traffic. If bad channel conditions are experienced it could occur that a datagram is still handled by the MAC layer while a new datagram is already being generated. To reduce undesirable transmission delay for time critical traffic an unreliable service is used. The unreliable service is derived from the reliable service, which means that sARQ is still used but now bounded by time or for a specified number of retransmissions. In the headers of the datagrams a timestamp is kept which indicates at what time the datagrams will expire. These timestamps are derived from the generation time of the datagrams. When the transmission time is expired or the number of retransmissions is reached for a specific datagram, the datagram is flushed in the buffer of the transmitter. The payload number (one bit) will be toggled when a new datagram is sent. At the moment the receiver receives the new datagram with the toggled payload number it will conclude that the previous datagram is expired and will flush this datagram in his buffer.

- Quality of Service
  For each link that carries priority traffic a priority slot is assigned to provide for minimal channel access so that the quality requirements for the link can be met. Additional quality of service is provided by means of the SPS. After token distribution at a supervisor priority slot the node that receives the token will determine which traffic point will be served. Priority traffic will have precedence above other traffic (see also 2.5.1).

- Queue behaviour
  During some simulations datagram drops in the queue were observed, which resulted in empty queues. Because of this phenomenon, link throughput was affected. After some research of simulations results it turned out that it is very likely that broadcast messages are used in NS to exchange status information between the nodes on the network. Due to the fact that unacknowledged broadcast is not a reliable service it is possible that these messages do not arrive. This will cause the traffic generators to stop supplying new datagrams and the queue to drop datagrams that are in the queue at that moment. As the real meaning of these messages is not found in the documentation of NS, the problem of queue drops is solved by two adaptations. The first one consists of making the transmission time of these messages zero, so if a broadcast message is fetched from the source node's queue, the message is at that moment also received by all the nodes participating on the Bluetooth HR channel. Second adaptation consists of not affecting broadcast messages by errors. Which means that the messages will be guaranteed received if they are sent.
3.4 Extension of the wireless node architecture

To allow for simulating errors on a link, an extension has been made to the protocol stack of the wireless node architecture. As can be seen from Figure 3.2 a new layer is introduced called the error generator.

![Diagram of extended protocol stack wireless nodes]

---

**Figure 3.2: Extended protocol stack wireless nodes**

The error generator can be seen as some sort of ‘filter’ where packets flow through and where some will be affected with a segment or packet header CRC error (the last one implies a packet error). The error generator provides the ability to generate random errors on basis of an algorithm (an error model) or generating scenario specific errors. The error generator is present in all nodes participating in a Bluetooth High Rate piconet.
Errors are generated in the incoming link. This way each link will experience different (random) error patterns, because in case of multicast or broadcast traffic all the receiving nodes will in practice almost never experience the same segment or header error in a packet.

### 3.5 First results

For the first test with the modified model the same simulation set-up is used as the one used in [6]. The simulation consists of two nodes (A and B) on a High Rate channel. Both nodes are stationary and are in radio range of each other. Node A is assigned to be supervisor, a priority slot (SPS) for this node is assigned every 12.5ms (1000 HR time slots). Node A also starts an FTP transmission, which will take place under ideal conditions, to node B. The FTP transmission is simulated for 30 seconds. The 8-DPSK modulation method is used for this simulation.

A calculation is made of the throughput to be expected before the modified implementation was simulated. In order to verify the model also latency of datagrams on the MAC layer is calculated and measured. Section 3.5.1 and 3.5.2 present throughput calculations and measurements while section 3.5.3 and 3.5.4 show the latency calculations and measurements.

#### 3.5.1 Throughput calculations

The calculation is executed on the condition that no errors occur during transmission.

Simulation time \( t_{\text{sim}} = 30 \text{sec} \)

Number of priority slots in simulation interval

\[
\# PS = \frac{t_{\text{sim}}}{T_s \times PS_{\text{int}}} = \frac{30\text{sec}}{12.5\mu s \times 1000} = 2400
\]

where \( T_s \) is the Bluetooth HR timeslot length and \( PS_{\text{int}} \) the Priority Slot interval length in timeslots.

**TCP datagram size**

\[
TCP_{\text{mes\_size}} = \text{payload\_size} + \text{size}_{\text{TCP\_data}} + \text{size}_{\text{TCP\_header}} + \text{size}_{\text{IP\_header}} + nr_{\text{segments}} \times \text{size\_segment\_header} = 1500 + 20 + 20 + 13 \times 4 = 1592 \text{ bytes}
\]

where

\[
\text{payload\_size} = 1500 \text{ bytes of user data; size}_{\text{TCP\_header}} \text{ and size}_{\text{IP\_header}} = 20 \text{ bytes; size\_segment\_header} = 4 \text{ bytes; size\_max\_segment\_header} = 128 \text{ bytes}
\]

**TCP acknowledge size**

\[
TCP_{\text{ack\_size}} = \text{size}_{\text{TCP\_header}} + \text{size}_{\text{IP\_header}} + nr_{\text{segments}} \times \text{size\_segment\_header} = 40 + 20 + 1 \times 4 = 64 \text{ bytes}
\]
\[ n_{\text{segments}} = \frac{\text{size}_{(TCP+IP)_{\text{header}}}}{\text{segment}_{\text{size \ max}}} = \frac{40 + 20}{128} = 1 \]

and where: \( \text{size}_{TCP_{\text{header}}} = 40 \) bytes; \( \text{size}_{IP_{\text{header}}} = 20 \) bytes; \( \text{size}_{\text{segment_header}} = 4 \) bytes

Modulation with 8-dpsk (conversion factor \( 12Mb/s = 1.5MB/s = 1.5e10 B/s \)) yields

\[ t_{\text{data}} = \frac{\text{TCP}_{\text{msg}_{\text{size}}}}{\text{conv \ factor}_{8\text{DPSK}}} = \frac{1592B}{1.5e10 B/s} = 0.0010613s \]
\[ t_{\text{ack}} = \frac{\text{TCP}_{\text{ack}_{\text{size}}}}{\text{conv \ factor}_{8\text{DPSK}}} = \frac{64B}{1.5e10 B/s} = 0.00004267s \]

Header transmission time (22 bytes header + 6 bits trailer)

\[ t_{\text{header}} = \frac{\text{header}_{\text{size}}}{\text{conv \ factor}_{8\text{DPSK}}} + \frac{\text{trailer}_{\text{size}}}{\text{conv \ factor}_{8\text{DPSK}}} = \frac{22B}{0.5e10 B/s} + \frac{0.75B}{1.5e10 B/s} = 44.5\mu s \]

where \( \text{conv \ factor}_{\text{DBPSK}} = 4Mb/s = 0.5MB/s = 0.5e10 B/s \)

Total TCP datagram transmission time

\[ t_{\text{TCP \ message}} = t_{\text{data}} + t_{\text{header}} + t_{\text{switch}} = 1.0613ms + 44.5\mu s + 25\mu s = 1.1308ms \]

this result rounded to the next Bluetooth HR slot time slot gives

\[ \left[ \frac{1.1308ms}{12.5\mu s} \right] * 12.5\mu s = 1.1375ms \]

Total TCP acknowledge transmission time

\[ t_{\text{TCP \ ack}} = t_{\text{ack}} + t_{\text{header}} + t_{\text{switch}} = 42.67\mu s + 44.5\mu s + 25\mu s = 112.17\mu s \]

this result rounded to the next Bluetooth HR slot time slot gives

\[ \left[ \frac{112.17\mu s}{12.5\mu s} \right] * 12.5\mu s = 112.5\mu s \]

Together this will give a total of \( 1.1375ms + 112.5\mu s = 1.25ms \)

In the simulation set-up described in the introduction of section 3.5, node A is the supervisor node. Round Robin is adopted so that node A and node B will alternately have access to the channel. When node A receives the token according to the Round Robin scheme no broadcast inquiry is carried out. Now one priority slot interval will be examined (\( 12.5\mu s * 1000 = 12.5ms \)) for the case that node A possesses the token.

Total simulation time (1 slot) = 12.5ms

Number of uninterrupted datagram transmissions

\[ \text{pkt}_{\text{trans \ unint}} = \left[ \frac{12.5ms}{1.25ms} \right] = 10 \]

So exactly 10 transmissions of 1500 bytes of user data will fit in one priority slot interval of 12.5ms.
The throughput yields:
\[
\frac{\text{pkt}_{\text{trans}} \times \text{unint} \times PS \times \text{payload}_{\text{TCP-data}}}{t_{\text{sim}}} = \frac{10 \times 2400 \times 1500B}{30s} = 1.2MB/s = 9.6Mb/s
\]

3.5.2 Measured throughput

The model is configured for the simulation set up described above. After simulation, the link throughput is measured from the trace file.

In the table below the results of the measurements and the calculation of the previous section are shown, including a measurement result obtained using the initial implementation and an almost identical simulation set-up. The only difference made is the usage of a priority slot instead of a supervisor priority slot as is used with the current implementation.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Token Distribution Interval</th>
<th>Throughput (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>12.5ms</td>
<td>9.60</td>
</tr>
<tr>
<td>Simulated (as reported in [6])</td>
<td>12.5ms</td>
<td>9.46</td>
</tr>
<tr>
<td>Simulated (current implementation)</td>
<td>12.5ms</td>
<td>9.04</td>
</tr>
</tbody>
</table>

Table 3.1: Analytical and measurement results

First results with the modified model show that compared with the initial model the link throughput is 0.42 Mbps lower when simulating with the 8-DPSK modulation method. This can be explained by the fact that because of ARQ rules only the last segment of a datagram is allowed to be partially filled, whereas in the initial model any segment of a datagram that is sent just before a priority slot could be partially filled. Figure 3.3 illustrates this. Because the current implementation causes more dead time, the throughput will decrease.

![Figure 3.3: Segmentation in (a) initial implementation and (b) modified implementation](image)

The simulated throughput is lower than the analytical value as calculated in the previous section. The difference (0.56Mb/s) between the analytical and the simulated throughput can be explained by the changes made, shown in Figure 3.3, and the following reasoning:

The calculations from 3.5.1 do not account for the exceptions mentioned below, which will introduce more overhead on the channel and will cause the throughput to decrease.
- Node A (supervisor) has direct access to the channel in case it is its turn in the round robin scheme, on the other hand Node B first needs a token passing from the supervisor when it is its turn in the round robin scheme. In case Node B has no data to serve, it will carry out a broadcast inquiry what will cause additional overhead on the channel.
- TCP acknowledgements are not always sent directly on the reverse link if they are not present in the interface queue at that moment. This will cost another extra MAC acknowledge to retain the token between a token pair.
- If a packet is sent from node A to B it is possible that there is not enough time to send the acknowledge of the packet from Node B to A because of an approaching priority slot. The acknowledge will be sent after the next priority slot, provided that Node B will get the token at that priority slot. Else node A must explicitly request for an acknowledge of Node B what will cost an additional header only packet on the link.

### 3.5.3 Latency calculations

The time needed to transmit a 60 byte TCP acknowledgement with the 8-DPSK modulation method on the MAC/DLC layer is:

\[
t_{\text{ack}} + t_{\text{header}} = 0.00004267 \text{s} + 44.5 \mu\text{s} = 87.2 \mu\text{s}
\]

where \(t_{\text{ack}}\) and \(t_{\text{header}}\) as calculated in 3.5.1.

The time needed to transmit a 1540 byte TCP datagram with the 8-DPSK modulation method on the MAC/DLC layer is:

\[
t_{\text{data}} + t_{\text{header}} = 0.0010613s + 44.5 \mu\text{s} = 1.1058\text{ms}
\]

where \(t_{\text{data}}\) and \(t_{\text{header}}\) as calculated in 3.5.1.

### 3.5.4 Measured latency

Using the simulation set-up described above a second measurement is carried out to verify the datagram latencies that are experienced end-to-end at the MAC layer. Latency of a datagram is measured by calculating the difference between the time the transmitting device fetches the datagram from the interface queue and the time of passing the datagram to the upper layer by the receiving device. In Figure 3.4 the datagram latencies that occur during the 3D-second simulation are shown. The peaks in the figure at 93.9\(\mu\text{s}\) and 1.1061ms are the latencies of respectively 60 byte acknowledge datagrams and 1540 byte data datagrams in case the datagram is transmitted directly without any interruptions of priority slots.

The deviation between the calculated and the measured values can be explained by the fact that a transmission always starts exactly at a time slot. So the time, a transmitter is waiting for a new time slot, called offset, has to be added to the transmission time. Because of priority slots, interruption patterns in data transmission are experienced when simulating with ideal channel conditions. As there are a few interruption patterns possible in ideal channel conditions a certain 'periodicity' in the measured values is found. Which means that some latencies occur more than once if a certain interruption pattern is experienced more than once.

The peaks next to the largest one and of which the difference in delay is within range of the Bluetooth HR slot time (12.5us), are caused by fetching a datagram from the interface queue close to an approaching time slot i.e. there is a certain offset.

All values in the figure that are left are caused by additional latency due to interruptions of traffic by priority slots. Interruption patterns are:
• Data is ready to be sent but the number of timeslots left before a priority slot is not sufficient for transmitting a single segment. Offset can also be experienced before a priority slot. In these two cases some dead time is experienced.

• Data transmission is ongoing and interrupted by a priority slot. After this priority slot the last transmitting traffic point gets immediate possession of the token.

• After a priority slot the token is distributed to node B:
  o This node has a waiting transport layer acknowledge which will be sent and node A will get the token back
  o This node has no data and carries out a broadcast inquiry; node A will again get the token after token passing (because node A has always data to serve).
  o This node has no data but has to acknowledge a datagram that is received just before the last priority slot. Since there was no time left before the priority slot for this node to send the acknowledge it will be issued directly after the priority slot.
  o This node has sent a transport layer acknowledge just before the last priority slot but has not received a baseband acknowledge corresponding to that datagram yet. Therefore node B will issue an acknowledge request to node A. After receiving this acknowledge another transport layer acknowledge can be send to node A.

For a two-node scenario in an error free channel these are the delay factors that are introduced on the MAC/DLC layer during data transmission. When considering a non-ideal channel, more overhead due to retransmissions and dead time is to be expected and so transmission time will increase. In case more nodes participate on the channel, broadcast inquiry time and time needed for token passing will increase.

![Figure 3.4: Datagram latency on MAC layer](image-url)
3.6 Conclusions

This chapter gives insight in the way the Bluetooth High Rate extension is modelled. The wireless node architecture that was reported in [6] has been modified and extended so that simulations can be run with more than two nodes on the HR channel. Furthermore the wireless node architecture is extended with a new layer, which provides for generation of errors on the channel.

The modified and extended model is tested and compared with calculations and simulation results of the initial implementation. The measured throughput differs 0.56 Mb/s compared with the calculations. This can be explained because the calculations do not account for dead time and overhead due to among others additional baseband acknowledgements and broadcast inquiry.

In a second test the datagram latency on the MAC layer is verified. This gave insight in the delay factors that are introduced on the MAC/DLC layer during data transmission.
4 Simulation set-up

Abstract
In this chapter the simulation set-up will be outlined. Section 4.1 describes the way errors are generated, traffic sources that are used for simulations and the differences between the implementation and the draft specification of the MAC/DLC layer. Section 4.1 concludes with an overview of the simulation execution. In section 4.2 three metrics are presented that are used for performance measurement of networks.

4.1 What is simulated
Basis for the simulations that were carried out are the scenarios that are presented in the next chapter. Each scenario is described in a simulation script.
In the following sub-section an explanation is given on how and which non-idealities are taken into account in the model. Sub-section 4.1.2 presents the traffic sources that are used for simulations. The model differs at a number of points with the draft specification. In the third sub-section these will be described. In the last sub-section some general topics about the simulation are presented.

4.1.1 Channel conditions
Bluetooth can experience interference from various sources that also operate in the 2.4 GHz ISM band, some examples are:
• Microwave oven
• Other Bluetooth piconets
• WLAN 802.11b/g networks
• HomeRF wireless network
• Cordless telephones
• Some medical equipment

C/I is the ratio of the desired signal level (C) to the interfering signal level (I). When monitoring a Bluetooth HR channel over time, the Carrier-over-Interference (C/I) ratio of this channel will vary because of possible interferences that are experienced by Bluetooth. An example of a possible C/I ratio over time is shown in Figure 4.1.

![Figure 4.1: Channel C/I](image-url)
To provide for realistic error generation in Bluetooth HR simulations, a real time channel C/I ratio should be the input for the simulations. Since this interference dependent modelling would take too much time there is chosen to use independent error generation for simulations. Therefore the error generator described in chapter 3 will generate random errors with a specified error rate, where the header and segment errors are related with a certain ratio. On the basis of [9] is chosen to use a header failure to segment failure ratio of 1:2 for simulations described in this report. This means that the probability of segments failure is twice the probability of headers failure.

In this project is expected that at Header Error Rates greater than 5 %, Dynamic Channel Selection will be performed. Dynamic Channel Selection is a mechanism to position the HR channel dynamically at another position in the 2.4GHz frequency band. A reason why DCS could be performed is for example interferences at the frequency the HR channel is currently positioned. The Header Error Rates used for the simulations range from 0 to 5 %. Note that the SER in the simulations will range from 0 to 10 %. In this project is expected that Bluetooth HR will operate mainly at 0 to 2% HER, this region is referred to as operating region.

As a Bluetooth piconet can experience interference during data transmissions, from for example another Bluetooth piconet, data will be either successfully transmitted or fail to transmit. Therefore it is preferable to choose a modulation method that offers fastest data transmission for best performance, when there is no interference on the channel. For that reason 8-DPSK will be used for simulations.

Finally, non-ideal behaviour like fading and shadowing, which are experienced in wireless networks, are not taken into account in the simulations.

### 4.1.2 Traffic sources

In the table below the traffic sources used for performance evaluation in this report are outlined together with the transport protocol that should be used for each source. Both protocols are part of the TCP/IP protocol suite.

<table>
<thead>
<tr>
<th>Traffic source</th>
<th>Transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>a File Transfer Protocol (FTP)</td>
<td>TCP</td>
</tr>
<tr>
<td>b Constant Bit Rate (CBR)</td>
<td>UDP</td>
</tr>
</tbody>
</table>

Table 4.1: Traffic generators

In NS the traffic sources are provided by means of traffic agents. The FTP agent will generate datagrams, destined for a so-called receiving agent, of 1500 bytes of user data. After receiving the complete datagram, the receiving agent will issue a transport layer acknowledge of 60 bytes. The CBR agent is different in the way that it periodically generates datagrams with a certain interval, also destined for a receiving agent. The CBR receiving agent will not issue transport layer acknowledgements. The CBR source that is used in this report generates 627 byte datagrams each 26.125ms according to a bit rate of 192kb/s. The time in which the datagrams are generated is called the datagram inter arrival time (DIAT) in this report.
4.1.3 Differences between the draft specification and the NS model

In this section an outline is provided of the differences between the model and the draft specification version 0.65 as of February 26th 2003. Some differences are caused by restrictions of the Network Simulator while others are made for simplification of the model.

1) In the Network Simulator it is not possible to have multiple links between a node pair. Addressing of datagrams generated by the traffic generators does not allow for making differences between multiple links between a node pair.

2) Flow control is not implemented as it is assumed that the buffer size is infinite.

3) Priority slot contention is defined and implemented in a different way
   a) Priority slot contention in the current model is defined as two priority slots taking place in exact the same time slot.
   b) Priority slot contention is prevented by choosing the right interval and offset in the simulation script. However, priority slots can follow immediately after each other. So it is not guaranteed that just after a priority slot, assigned to a traffic point, any time is available to carry out a transmission. No precedence to priority slots is applied, except in the following case: When two priority slots follow after each other and the number of slots in between the priority slots are not sufficient to transmit any data, the priority slot that occurs later in time has precedence above the first one since that one couldn't be used by a traffic point to serve data. The draft specification [2] provides a more sophisticated priority slot contention resolution mechanism.

4) The specification [2] still allows for erroneous data to occur in case the unreliable service is used. The error is explained with the following traffic sequence in which two nodes A and B are involved.
   a) A sends a datagram to B of which the first segments will be corrupted on the channel. Payload datagram number 0 is used for this datagram.
   b) B sends an acknowledge back to A with the request for retransmission of the corrupted segments, which will be dropped on the channel.
   c) A gets possession of the token and sends a new datagram (because the previous one is expired) to B, this datagram is also dropped on the channel. Payload datagram number 1 is used for this datagram.
   d) A gets possession of the token and sends again data to B of a new datagram (payload datagram number is toggled back to 0) containing exactly the number of corrupted segments at a), B receives this datagram. Because this is the first datagram B receives after it requested retransmission of corrupted segments and the same payload datagram number is used, it will think that this is the retransmit data it requested at step 1. Node B will now mix data that it received at step a) with the data that it just received and will send this erroneous data to the upper layer.

The current implementation will not allow for erroneous data since the transmitter will take care that at least one packet that is part of a particular datagram is seen at the receiver. Applying this provision to the example just described, node B would have seen at least one packet of the datagram sent at step c). So it would have known that the previous datagram, with payload datagram number 0, is expired and that the datagram received at step d) is a new datagram.

5) Broadcast is implemented only to serve NS specific packets during simulations; general broadcast of packets cannot be used since the unacknowledged broadcast
functionality takes no transmission time in account. (See also section 3.3; topic Queue behaviour)

The following features described in the draft specification [2] are not included in the MAC layer model:
• Forward Error Correction
• Dynamic channel selection (in case of poor channel conditions)

4.1.4 Simulation set-up
To provide for reliable results of the simulations and approaching real life application durations, a value of 120 seconds is chosen for the duration of the simulations. All simulations will take place with only one HR channel. The assignment of a topology is not dynamic during simulations. Links are set up at the beginning of the simulation and will be maintained until the end of a simulation. Like in real deployments there will be a supervisor present in the HR channel, which will provide for token distribution at the supervisor priority slots. The token distribution is needed in real life because nodes will start and end traffic dynamically over time. Therefore all nodes on the channel need to have periodic channel access time. In the simulations the traffic generators will not start and end dynamically over time but however, the supervisor will be present to be able to measure the influence of the token distribution on the total performance. The Supervisor Priority Slot Interval and the node that acts as supervisor are specified in the script file. The traffic links between nodes on a Bluetooth HR channel may each carry different types of traffic. The HR MAC/DLC layer provides two services, reliable and unreliable, to handle specific needs for those traffic types.
• Reliable service: By using sARQ data will be guaranteed delivered at the receiver.
• Unreliable service: Like the reliable service with the restriction that when the transmission time of a datagram exceeds the DIAT or a maximum number of allowed retransmissions is consumed, this datagram will be flushed in the buffers of transmitter and receiver. When a datagram that is not yet or not completely send is expired, an acknowledge request will be send to the receiver of the datagram to prevent from the error as described at point 4) in the previous section. This introduces some extra overhead on the channel, i.e. the transmission time of the request and the corresponding acknowledge of the receiver.

Only FTP and CBR traffic sources are used for simulations and will start generating traffic after 2 seconds; the first two seconds are used by the Network Simulator to exchange set up messages.

4.2 What is measured
Following metrics are used for performance measurement of networks in general:
• Throughput (aggregate and link)
• Datagram delay
• Number of expired datagrams (for unreliable service)

Using the first measurement parameter, throughput, the user data offered at TCP level is measured, i.e. TCP/IP header and Bluetooth HR header are considered as overhead as is shown in Figure 4.2. In addition, also segment headers that are added at the MAC/DLC layer are counted as overhead. The link throughputs of all separate links, as well as the aggregate throughput of all links together will be measured.
A second measurement parameter, datagram delay, is used for measuring delay. This is measured as follows: At the moment a datagram arrives in the interface queue (IFQ, see also Figure 3.2) a time stamp is set in the header of the datagram. When this datagram is completely received by the receiving device and is sent to the upper layer of its protocol stack, the delay is calculated. Measuring delay at a different layer compared to the throughput measurement is not very consistent, however a delay measurement implementation on TCP level could not be made.

The last metric used is measuring the percentage of datagrams that arrive within the Datagram Inter Arrival Time (DIAT) in case time critical traffic is served on a link. At the moment a datagram has reached its transmission time limit or has reached the maximum number of allowed retransmissions the datagram will be declared expired (see also 3.3; unreliable service). The percentage of datagrams that successfully arrive within the DIAT will be measured. Since this last metric encapsulates information that can be derived from the second measurement parameter, only the last metric will be used for measuring performance of time critical traffic in this report.

![Figure 4.2: Packet length at different layers](image)

### 4.3 Conclusions

Several sources also utilize the 2.4 GHz ISM band where Bluetooth High Rate operates in. This causes interference in the radio traffic of Bluetooth HR. Research on interference modelling pointed out that there is a certain ratio between Header Error Rate and Segment Error Rate when considering unconditional segment failures [9]. Because only interference is considered for simulations, the 8-DPSK modulation method is used as data is preferably transmitted as fast as possible at moments the interference is not present. Scenarios that are presented in the next chapter provide the basis for the simulations.

Due to some imperfections of the Network Simulator or because of simplification of the model there are some differences between the Bluetooth HR specification and the model. Most essential differences are a bug fix to prevent from erroneous data that is not foreseen in the specification, the restriction of NS that multiple links between a node pair are not allowed because of addressing problems and a simplified priority slot contention resolution mechanism.
Typical metrics as throughput, datagram delay and number of expired datagrams are used for the performance measurements of networks. From these, measurements of throughput (aggregate and link) and number of expired datagrams are used in this report.
5 Scenario description

Abstract
In this chapter typical user scenarios will be described. In the introduction will be explained what the definition of a scenario is. The user scenarios are divided into two sections; unicast and multicast scenarios. For each scenario will be explained how the topology will look like, where the scenario is used for in practice, what will be measured and which behaviour is expected in the measurement results.

5.1 Introduction
Before carrying out a performance evaluation, the simulator has to be provided with a script in which the configuration of a simulation (in this report defined as a scenario) is described. The configuration specifies:
1) How many nodes are simulated
   On the HR channel a specified number of nodes (or devices) will be present.
2) The network topology
   The network topology describes how the nodes on the HR channel are related to each other. In other words how logical traffic points and logical links between the devices are set up.
3) Which traffic generators are used
   To keep the evaluation consistent and comparable the same set of two traffic generators is used at each scenario.
4) Which transport protocol is used
   Traffic generated by the generators run over different protocols that are part of the TCP/IP protocol suite; TCP for a reliable service and UDP for an unreliable service.
5) Which node is assigned the supervisor function
   In each simulation one node on the Bluetooth HR channel is assigned the supervisor function. For the simulations is chosen to assign the supervisor function to one of the nodes that are specified in the network topology and serve traffic.
6) Which nodes are assigned a priority slot
   For each traffic point a priority slot can be assigned. In addition a supervisor priority slot is always assigned to the specified supervisor node. The latter is used for carrying out token distribution by the supervisor.
7) What the individual priority slot and token distribution interval and offset is
   Each priority slot will occur periodically at a specified priority slot interval (PSI) and each supervisor priority slot at a specified token distribution interval (TDI). For both PSI and TDI, an offset (a value between 0 and PSI/TDI value) can be specified.
8) What type of data streaming service is used for each traffic link
   For each traffic link between a pair of traffic points can be specified what the type of service is: reliable or unreliable. The forward and reverse link will offer the same service.
9) The restriction parameters in case unreliable service is enabled
   Two parameters are needed for enabling unreliable service. The first one is the datagram inter arrival time of datagrams in time critical traffic. The second parameter is the maximum allowed number of retransmissions of a datagram. In the simulations the unreliable service is only used for CBR traffic.
10) The way errors are generated
As explained in chapter 4

11) The length of the simulation
Simulations will start at $t = 0\text{sec}$ . The traffic generators that are used in a certain simulation will start generating traffic at $t = 2\text{sec}$ and stop at $t = 122\text{sec}$ . An exception is made for CBR traffic to reduce the time between datagram generation and the time at which the datagram is served for the first time. CBR traffic generators will start just before the first PS, assigned to the traffic point that serves the CBR traffic, which takes place after $t = 2\text{sec}$ .

More details about the scripts that are used can be found in appendix A in which an example script is shown and explained.

With the above described configuration options, a scenario is set up. In the following section four user scenarios are described. All four are set up with unicast traffic links. In section 5.3 scenarios are described which are set up with multicast traffic links and can be used for future work. However, since multicast functionality is not provided in the current model the multicast scenarios will not be used for simulations.

The performance evaluation will start with a very basic scenario and will subsequently be extended to more complex scenarios.

### 5.2 Unicast scenarios

In this section typical user scenarios, which are based on unicast links, are presented. For each scenario the topology is explained together with an outline of the measurements that are carried out on these scenarios. Also expected behaviour of the measurements is discussed.

#### 5.2.1 Scenario 1.1: Single logical link

For Bluetooth data-communication the most basic scenario one can think of is a scenario where two devices participating on the same Bluetooth HR channel and in radio range of each other are communicating. Between the nodes only one unicast logical traffic link (or two if the forward and reverse link are considered as separate links) resides for traffic from node 0 to node 1 and vise versa. A traffic generator will be connected to node 0. Node 0 will also be the supervisor node, which carries out token distribution. In Figure 5.1 the topology is shown. The scenario is very practical, as it will occur daily in various user applications.

![Figure 5.1: Topology of scenario 1.1](image)

**Measurement 1:** Throughput measurements will take place using an FTP traffic generator over TCP. This measurement will give insight in the maximum throughput to be attained using Bluetooth HR. Also the effect of the token distribution interval length on the throughput can be measured here. When reducing the token distribution interval,
which implies a higher token insertion rate, a decrease of dead time caused by packet loss on the channel and thus increase of the throughput is expected.

The following measurements will be carried out with constant bit rate traffic served on the traffic link. In these measurements the percentage of datagrams that arrive within the datagram inter arrival time will be measured.

**Measurement 2:** For this measurement only the supervisor priority slot is used on node 0 so CBR traffic will be served when node 0 is in turn in the round robin scheme or the token is passed back from node 1 to node 0. Although in real deployments a traffic point that serves time critical traffic is always assigned a priority slot, it is interesting to see what the performance with respect to number of expired packets is, using only the (periodic) supervisor priority slot.

**Measurement 3:** In addition to the previous measurement, a priority slot is assigned to the traffic point that serves the traffic. So a minimal degree of channel access is guaranteed for the traffic point that serves the constant bit rate traffic. Because of this provision, the percentage of datagrams that will arrive within the datagram inter arrival time should increase compared to the results of measurement 2.

The results obtained in this measurement will be used for the practical example in which Bluetooth HR is used for streaming audio to or from a headset. Measurement results will be analysed to find a setting in which typical audio requirements, usually expressed in terms of **Frame Error Rate (FER)**, are met. In this report the frame error rate will be referred to as the percentage of datagrams sent that did not arrive within the datagram inter arrival time.

### 5.2.2 Scenario 1.2: Two logical links connected to one source node

A simple extension on the previous scenario is to extend one of the two nodes with an additional unicast link to a third node. The scenario presented here is used in practice when one device is involved in two data exchanges. So there are three nodes on the Bluetooth HR channel and they are all in radio range of each other. Between node 0 and 1 and between node 0 and 2 a unicast link resides. The topology of this scenario is shown in Figure 5.2. Two traffic sources are connected to node 0, which acts as supervisor node. Now node 0 will serve datagrams for node 1 and node 2.

![Figure 5.2: Topology of scenario 1.2](image)

**Measurement 1:** First measurement consists of FTP transfer on both links to see what the aggregate throughput of the two links is. The aggregate throughput should approach
the link throughput measured at scenario 1.1. If the token in the round robin scheme is assigned to node 0 at a Supervisor Priority Slot, two traffic points (254 and 251) have to share this token. Interesting to see in this case is whether the Bluetooth HR scheduler provides fair channel access so that both links will have about the same degree of channel access.

**Measurement 2:** In the second measurement an FTP source as well as a CBR source will be connected to node 0. An optimal setting has to be found in which the aggregate throughput approaches the aggregate throughput with two FTP sources and the CBR traffic just satisfies quality requirements. Also here will be looked for a configuration that satisfies audio requirements.

**Measurement 3:** The last practical usage case that is applied to this scenario is an audio server that provides two users (devices) of a high quality audio stream. Settings obtained from the third measurement of scenario 1.1 will be used to verify if quality requirements will still be met. Because of the additional priority slot that is assigned in this scenario compared to measurement 3 of scenario 1.1 and the fact that the channel is not continuously loaded with traffic compared to the previous measurement, thus more chances to transmit a datagram successfully, this scenario should provide less datagram loss compared to those measurements when using same priority slot configuration as in scenario 1.1.

### 5.2.3 Scenario 1.3: Two logical links connected to different source nodes

Another way of adapting the first scenario is to extend the scenario with an additional pair of nodes. In this scenario multiple pairs of devices are exchanging data with each other. All four nodes are on the same Bluetooth HR channel and are in radio range of each other. Between each node pair (0 and 1, 2 and 3) a unicast link resides. Figure 5.3 shows the topology. Nodes 0 and 2 are connected with separate traffic sources. Node 0 will be the supervisor node.

![Figure 5.3: Topology of scenario 1.3](image)

**Measurement 1:** First objective of this measurement is to examine if the aggregate throughput with two FTP sources meets the throughput measured in scenario 1.1. Token passing plays an important role in this scenario. For that reason there will be investigated if the token passing in this scenario provides a fair channel access for all nodes participating on the channel.

**Measurement 2:** The second measurement is comparative to the third measurement that is carried out for scenario 1.2, with the difference that two different nodes will serve
audio streaming. Also here datagram loss should be less than is measured for scenario 1.1 because of an extra priority slot, which creates additional chances to transmit a datagram successfully.

### 5.2.4 Scenario 1.4: Two logical links connected to one source node and a single logical link

In the last unicast scenario, scenario 1.2 is extended with an extra node pair. Between this node pair one unicast traffic link resides. So there are 5 nodes on the HR channel, all in radio range of each other. Node 0 will be the supervisor node. Figure 5.4 shows the topology of this scenario.

![Figure 5.4: Topology of scenario 1.4](image)

**Measurement 1:** In this simulation the influence of an idle link will be measured. So two FTP traffic generators are connected with node 0, both serving different links, while no traffic generator is connected to node 3 or 4. The results will be compared with scenario 1.2 where the additional idle link was not taken into account. Compared to scenario 1.2, token distribution of the supervisor will in this scenario cause the token to be distributed out of the three-node set what will lead to lower throughput because of additional token passing and broadcast inquiry overhead.

### 5.3 Multicast scenarios (For future work)

In this section some typical user scenarios using multicast are presented. However these will not be used in simulations because multicast functionality is not implemented at the moment. The scenarios could serve as input for future work.

#### 5.3.1 Scenario 2.1: Single multicast logical link

![Figure 5.5: Topology of scenario 2.1](image)
The first multicast scenario exists of the most basic multicast scenario of three nodes and one multicast link. In this scenario the effect of the acknowledged multicast on throughput can be measured when varying channel conditions.

### 5.3.2 Scenario 2.2: Single multicast and unicast logical link

![Diagram of Scenario 2.2](image)

Compared to the previous scenario one unicast link is added to simulate for example the situation in which one device is multicasting an audio stream, while at the same time an FTP download takes place between a particular node pair in the topology.

### 5.3.3 Scenario 2.3: Three multicast logical links

![Diagram of Scenario 2.3](image)

The scenarios presented in Figure 5.7 are more complicated. Again there are three nodes and each node has a multicast link to the other two nodes. A practical example of this scenario is a scenario in which three users are playing a multi-player game on a mobile device. When one of the users takes action in the game the other users must be informed about this action so that the screen can be updated. Additionally it is possible that all three users are wearing a headset that also utilizes Bluetooth HR. The latter is shown in the right of Figure 5.7. With the help of this scenario can be investigated whether or not the delays introduced by the MAC/DLC layer of Bluetooth are acceptable for multi-player games.
5.4 Conclusions

Configuration options, which define a scenario, are discussed. Some typical user scenarios are presented together with the set of measurements that is carried out on these scenarios. Expectations of measurement results are discussed. Only unicast link based scenarios will be used for simulations since multicast functionality is not implemented in the current model.
6 Measurements and analysis

Abstract
In the previous chapter typical user scenarios are described. This chapter presents the results that are obtained by measurements on these scenarios. On basis of these results performance analysis is carried out and conclusions are drawn.

6.1 Measurements based on unicast scenarios
In this section all unicast scenarios that are presented in chapter 5 are simulated and analysed.

6.1.1 Measurements of scenario 1.1
Measurement 1:
To measure the maximum throughput to attain with Bluetooth High Rate the link has to be continuously active in transferring data. Therefore an FTP traffic source is used.

Figure 6.1: Link throughput (measurement 1 of scenario 1.1)
In 1:2 token distribution intervals the token will because of Round Robin be passed to node 1 (see Figure 5.1). In case node 1 has no traffic to serve i.e. it has no transport layer acknowledge to transmit the following traffic sequence between node 0 and 1 can occur directly after the supervisor priority slot; token pass from node 0 to 1, broadcast inquiry from node 1, token pass from node 1 to 0, transmission of datagram from node 0
to 1 and transmission of transport layer acknowledgement from node 1 to 0. The time needed for this traffic sequence is:

\[ T_{\text{token\_passO}} + T_{\text{broadcast\_inquiry}} + T_{\text{token\_passI\_O}} + T_{\text{TCPmessageO\_I}} + T_{\text{ACK1\_O}} = \]

\[ 69.5\mu s + 5.5\mu s + 200\mu s + 69.5\mu s + 5.5\mu s + 1.1308ms + 0.0067ms + 112\mu s + 0.5\mu s = 1.6ms \]

Note that rounding of the transmission time to the next time slot is also accounted.

So the token distribution interval (TDI) should be at least 1.6ms to minimize additional overhead caused by fragmentation, i.e. that a datagram is transmitted using multiple HR packets. From this perspective at first TDI values of 1, 2, 3, 4, 5, 10 and 15ms are chosen for use in this scenario. Figure 6.1 shows the link throughput as function of the header error rate and for the chosen TDI's. One can see that the throughput is at its maximum for ideal channel conditions and decreases as the HER increases. Using a large token distribution interval the interruptions in data transfer should reduce to a minimum. This can be observed from Figure 6.1 as the largest TDI (15ms) will give highest throughput in ideal channel conditions (HER 0%) whereas a TDI of 1ms gives lowest throughput for 0% HER. However when header errors occur on the link, causing packets to be dropped, the link will be in an inactive state (dead-time) until the moment the token is re-inserted. For a TDI of 15ms this behaviour will lead to the lowest throughput for error rates above 1% as is shown in Figure 6.1. The larger the distribution interval the larger the potential dead time will be and so the lower the throughput will be. This is the reason why the throughput belonging to a TDI of 15ms in Figure 6.1 decreases in greater steps than the throughput belonging to a TDI of 5ms. Because of the behaviour just described the figure shows a cross over area from 0 to 3%. The cross over area is enlarged and shown in Figure 6.2 with interpolated HER values. From Figure 6.1 can also be seen that the link throughput with 3% and 5% HER and a TDI of 15ms gives respectively 50% and 68% throughput decrease compared to the case of 0% HER. So a header error rate of 3% shows already great impact on the throughput.

<table>
<thead>
<tr>
<th>HER (%)</th>
<th>TDI 1ms</th>
<th>TDI 2ms</th>
<th>TDI 5ms</th>
<th>TDI 10ms</th>
<th>TDI 15ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.63</td>
<td>8.58</td>
<td>8.87</td>
<td>9.01</td>
<td>9.06</td>
</tr>
<tr>
<td>0.5</td>
<td>7.27</td>
<td>8.05</td>
<td>8.36</td>
<td>8.29</td>
<td>8.18</td>
</tr>
<tr>
<td>1</td>
<td>7.01</td>
<td>7.72</td>
<td>7.89</td>
<td>7.67</td>
<td>7.36</td>
</tr>
<tr>
<td>1.5</td>
<td>6.76</td>
<td>7.41</td>
<td>7.45</td>
<td>7.00</td>
<td>6.55</td>
</tr>
<tr>
<td>2</td>
<td>6.54</td>
<td>7.12</td>
<td>7.01</td>
<td>6.40</td>
<td>5.82</td>
</tr>
<tr>
<td>2.5</td>
<td>6.33</td>
<td>6.86</td>
<td>6.60</td>
<td>5.88</td>
<td>5.15</td>
</tr>
<tr>
<td>3</td>
<td>6.12</td>
<td>6.58</td>
<td>6.22</td>
<td>5.30</td>
<td>4.56</td>
</tr>
</tbody>
</table>

Figure 6.2: Link throughput (measurement 1 of scenario 1.1); cross over area
Overall the throughput measured with a TDI of 1 ms performs bad compared to the throughputs obtained with TDI's of 1.6 to 2 ms as can be seen in Figure 6.3.

An interesting observation made during simulations is the fact that for 5% error rate the throughput is bounded at 5.56 Mb/s. This can be explained with the fact that statistically seen at 5% HER (implies 10% HER) almost each packet has a segment error. So the probability of segment retransmission for each transmitted datagram is rather high. If the 1.6 ms TDI as calculated above is taken as a starting-point and the retransmission time of one segment is added to it, the minimal TDI should be 1.9 ms to minimize fragmentation. The throughput of TDI's around 1.9 ms is bounded at 5.56 Mb/s for 5% HER as is shown in Figure 6.3. Smaller and larger TDI will only cause the throughput to decrease by respectively more overhead of packet headers through an increased number of packet transmissions (fragmentation) or increased dead time. From all TDI values (1.6 – 2 ms) that are bounded at 5.56 ms a TDI of 2 ms gives best performance because the throughput is higher, up to 0.3 Mb/s compared to a TDI of 1.9 ms, in the operating region.

Although a higher throughput is expected when increasing the token distribution interval, because of fewer interruptions by tokens, this is not always the case as is observed from Figure 6.1 and Figure 6.3. A TDI of 2 and 3 ms will have the same throughput and a TDI of 1.6 ms turns out to have a greater throughput than for TDI's of 1.75 and 1.9 ms. To find the cause of this behaviour the simulations are monitored. From the monitoring appears that stationary behaviour occurs in the simulations of the monitored TDI's. Which means that a particular traffic sequence, once it is occurred, is repeated till the end of the simulation. In the table below the traffic sequences for the respective TDI values are shown together with a calculation of the throughput belonging to these values. From this
Table 6.1: Traffic sequences and respective throughputs

<table>
<thead>
<tr>
<th>TDI</th>
<th>Traffic sequence</th>
<th>time slots needed for sequence</th>
<th>time needed for datagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6ms (128 slots)</td>
<td>T→,←TLA,MES(13)→,←TLA,MES(1)→, M(12)→,←TLA,MES(4)→,D(1)</td>
<td>6+9+9+13</td>
<td>84+9+34+1</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(9)→,←A,MES(5)→,D(3)→,←TLA,MES(8)→,D(1)</td>
<td>6+9+63+6+41+3</td>
<td>56+9+62+1</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(5)→,←A,MES(9)→,D(3)</td>
<td>6+9+35+6+69+3</td>
<td>28+9+91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6+9+9+6+27+1</td>
<td>70+9+55+6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6+9+42+6+76+1</td>
<td>21+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6+9+9+9+63+6+4</td>
<td>121+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8*152 time slots = 15.2ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 datagrams / 15.2ms = 1 datagram / 1.38ms</td>
<td></td>
</tr>
<tr>
<td>1.75ms (140 slots)</td>
<td>T→,←TLA,MES(13)→,←A,MES(3)→,D(1)→,←TLA,MES(7)→,←A</td>
<td>6+9+9+6+27+1</td>
<td>70+9+55+6</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(6)→,←A,MES(10)→,D(1)</td>
<td>6+9+42+6+76+1</td>
<td>21+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td>MES(3)→,←TLA,MES(13)→,←TLA,A,D(4)</td>
<td>6+9+9+9+63+6+4</td>
<td>121+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4*140 time slots = 7ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 datagrams / 7ms = 1 datagram / 1.37ms</td>
<td></td>
</tr>
<tr>
<td>1.9ms (152 slots)</td>
<td>T→,←TLA,MES(3)→,←A,MES(13)→,←TLA,A→,D(4)</td>
<td>6+9+9+6+27+1</td>
<td>70+9+55+6</td>
</tr>
<tr>
<td></td>
<td>MES(13)→,←TLA,MES(6)→,D(4)→,T→,←TLA,MES(7)→,←A,MES(10)→,←A</td>
<td>6+9+42+6+76+1</td>
<td>21+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td>MES(3)→,←TLA,MES(13)→,←TLA,MES(2)→,D(2)</td>
<td>6+9+9+9+63+6+4</td>
<td>121+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(11)→,←A,MES(6)→,←A,MES(7)→,←TLA,MES(13)→,D(3)</td>
<td>6+9+9+9+63+6+4</td>
<td>121+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(13)→,←TLA,MES(4)→,D(3)→,MES(9)→,←TLA,MES(10)→,D(4)</td>
<td>6+9+9+9+63+6+4</td>
<td>121+9+91+9+6+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8*152 time slots = 15.2ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 datagrams / 15.2ms = 1 datagram / 1.38ms</td>
<td></td>
</tr>
<tr>
<td>2.0ms (160 slots)</td>
<td>T→,←AR,MES(13)→,←TLA,MES(6)→,MES(7)→,←TLA,MES(13)→,←TLA,D(2)</td>
<td>6+9+9+9+48</td>
<td>49+9+91+9+2 = 2*160 time slots = 4ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 datagrams / 4ms = 1 datagram / 1.33ms</td>
<td></td>
</tr>
<tr>
<td>3.0ms (240 slots)</td>
<td>T→,←TLA,MES(13)→,←A,MES(13)→,←TLA,MES(3)→,D(1)</td>
<td>6+9+9+9+48</td>
<td>49+9+91+9+2 = 2*160 time slots = 4ms</td>
</tr>
<tr>
<td></td>
<td>MES(10)→,←TLA,MES(13)→,←A,MES(7)→,←A</td>
<td>6+9+9+9+48</td>
<td>49+9+91+9+2</td>
</tr>
<tr>
<td></td>
<td>T→,←TLA,MES(6)→,←A,MES(13)→,←TLA,MES(10)→,D(1)</td>
<td>6+9+9+9+48</td>
<td>49+9+91+9+2</td>
</tr>
<tr>
<td></td>
<td>MES(3)→,←TLA,MES(13)→,←TLA,MES(13)→,←TLA,A,D(4)</td>
<td>6+9+9+9+48</td>
<td>49+9+91+9+2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4*240 time slots = 12ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 datagrams / 12ms = 1 datagram / 1.33ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: Traffic sequences and respective throughputs


In Figure 6.4 the link throughput is shown as function of the token distribution interval choices. From this figure can be observed that for 0-2% HER and TDI values of 2 to 5 ms the throughput is steady or increasing, while the throughput only decreases for these TDI values when looking at header error rates of 3-5%. From a TDI of 5 ms the throughputs at 1 and 2% HER will decrease, but this decrease is smaller than for header error rates of 3-5%. TDI values in the range from 2-5 ms provide for optimal performance in the operating region (error rates ≤ 2%) and give minimal decrease of throughput for error rates of 3-5% compared to other choices of TDI values (10-15ms).
Chapter 6: Measurements and analysis

Figure 6.4: Link throughput (measurement 1 of scenario 1.1); throughput versus TDI

From the observations presented, a TDI of 2 to 5 ms for this scenario and set up is an optimal trade-off between the throughput at 0% HER and dead time in case of HER>0%.

Measurement 2:
The following results are obtained using a constant bit rate traffic source. Channel access can only be obtained via the supervisor priority slot.

Figure 6.5: Successful datagram arrivals (measurement 2 of scenario 1.1)
In Figure 6.5 the successful datagram arrivals using only a supervisor priority slot are shown. When a TDI of 26.125ms (the exact datagram inter arrival time) is chosen, the number of datagrams that arrive on time are decreasing rapidly due to the fact that there is only one chance that the datagram can be send. If this chance fails the datagram will never arrive within the datagram inter arrival time. Because only a supervisor priority slot is used, the token is passed each second token distribution interval to the node that is not serving traffic. This node will issue a broadcast inquiry since it has no data to transmit (no transport layer acknowledge for UDP). If the token goes lost during the token passes or broadcast inquiry due to a header error, the chance of sending a datagram will fail. Each second token distribution interval will thus introduce an increased probability of failure. To provide traffic points serving priority traffic with a minimum and guaranteed channel access the specification prescribes that those traffic points are assigned a priority slot, in addition to the supervisor priority slot.

Measurement 3:
In this measurement a priority slot is assigned to the traffic point that serves the constant bit rate traffic. The percentage of successful arrived datagrams will be measured for a PSI of respectively exact the OIAT and exact half the OIAT. For the token distribution interval, values are chosen that are respectively a bit smaller (25ms), a bit larger (30ms) and almost double (50ms) the datagram inter arrival time. When the TOI is equal to 25ms and the priority slot interval (PSI) is exact the datagram inter arrival time, each datagram has at most three chances and at least two chances that it can be send. Whereas TOI values that are greater than the OIAT do not provide for two chances of sending a datagram within the OIAT.

![Figure 6.6: Successful datagram arrivals (measurement 3 of scenario 1.1)](image)

The number of expired datagrams using a PSI of 26.125ms (the datagram inter arrival time) should be lower now than is shown in Figure 6.5. The chance that a datagram will
be sent is guaranteed because of the PSI and it will be increased due to the fact that the TDI offers an additional chance for sending a datagram.

In Figure 6.6 the percentage of datagrams that are transmitted successfully is shown. As expected, the number of datagrams that expire for a PSI of 26.125ms is lower than is shown in Figure 6.5 using a TDI of 26.125ms.

Now lets take a look at a practical example when Bluetooth HR is used for streaming audio with a headset. Allowed frame error rate (FER) in common used requirements for high quality streaming audio, for example MPEG layer 3 files, is smaller than 1%. The configuration that satisfies this requirement should contain a PSI that is equal or smaller than half the datagram inter arrival time (13.0625ms). A less stringent FER requirement of 3% is satisfied in case \( \text{PSI} \leq \text{DIAT} > \text{TDI} \).

### 6.1.2 Measurements of scenario 1.2

![Graph showing link throughput](image)

**Figure 6.7:** Link throughput (measurement 1 of scenario 1.2)

**Measurement 1:**
Both traffic points on node 0 are provided of FTP traffic sources to serve FTP traffic on both links. The links should have, due to fair token distribution, an equal amount of channel access. So the separate link throughputs should also be equal.
In Figure 6.7 the link throughputs of scenario 1.2 are shown. The throughput of both links is equal as expected, so the proposed scheduler of Bluetooth HR is fair towards channel access when a node serves traffic on multiple links at the same time.

The aggregate throughput is shown in Figure 6.8. As in this scenario extra overhead due to additional token passing and broadcast inquiry is introduced, the aggregate throughput is expected to be strictly equal or lower than the link throughput of scenario 1.1.

![Graph showing aggregate throughput](image)

**Figure 6.8: Aggregate throughput (measurement 1 of scenario 1.2)**

As can be seen from Figure 6.8 this condition is met. A difference of at most 0.31 Mb/s (at 0% HER) between both aggregate throughputs is observed; this can be explained with the additional overhead of broadcast inquiry and token passing that is introduced due to the additional node. Note that this overhead is largest for a TOI of 2ms. Due to the high token insertion rate the extra overhead causes for a TOI of 2ms and a HER of 5% up to almost 4% aggregate throughput difference when compared with the aggregate throughput from scenario 1.1.

The following observation is made during this measurement; for a TOI of 15ms and a HER of 4% the aggregate throughput of scenario 1.1 is 0.02 Mb/s smaller than for same conditions in scenario 1.2. The reason why the throughput of this scenario is for some cases greater than in scenario 1.1 could be explained with broadcast inquiry. When a broadcast inquiry is issued in scenario 1.1 there is only one responder. If the
acknowledge of the receiver got lost, dead time is introduced. Whereas in scenario 1.2 the probability that dead time is introduced by carrying out broadcast inquiry is smaller due to the fact that the acknowledgements of both responding nodes are required to go lost before introducing dead time.

Measurement 2:
Starting-point for this measurement will be the results of measurement 3 of scenario 1.1. These values are a PSI equal to half the DIAT and a TDI of 50ms for a high quality audio stream (≤1% FER). In case the FER of 1% is not met using these values there are two ways of adapting the priority slot intervals to a configuration that meets the requirement. First the PSI can be divided in half and secondly also the TDI can be divided in half, if needed more than once. Figure 6.9 shows the percentage of successful datagram arrivals on the link that serves CBR traffic.

![Figure 6.9: Successful datagram arrivals (measurement 2 of scenario 1.2)](image)

A FER of 1% for high quality audio is not met using the configuration from scenario 1.1. For a less stringent FER of 3% the configuration would be more than sufficient. However dividing the Priority Slot Interval in half provides very good performance; only at 5% HER a few (0.02%) of the transmitted datagrams will not arrive in time. When maintaining the PSI of half the DIAT and dividing the TDI twice in half the FER of 1% is also met.

Second part of the measurement results, aggregate throughput, is shown in Figure 6.10. To find a configuration that provides for optimal performance, the aggregate throughput is examined.

As can be seen from the figure the most optimal priority slot configuration, which is just discussed above provides also for most optimal aggregate throughput. For a HER of 5% the aggregate throughput is 15% higher than in case priority slot configuration is used in which the TDI is divided twice in half. No drastic decrease of aggregate throughput is observed compared to measurement 1 of this scenario. So Bluetooth HR is capable of serving time critical traffic and best effort traffic at the same time without significant loss of performance with respect to aggregate throughput.
Measurement 3:
For this measurement both links will have the priority slot configuration obtained from scenario 1.1. Offsets for the priority slots are chosen in the way that channel access is fairly divided among both links.

Compared to the previous measurement a 1% FER requirement should be met without adjusting the optimal configuration obtained from scenario 1.1. Because the traffic sources used for both links do, in contradiction to the previous measurement, not result in continuously loaded links, each supervisor priority slot provides a chance to send a CBR datagram. In addition, when a datagram is transmitted successfully from a traffic point on the transmitting node, another traffic point on that node may have a datagram.
waiting for transmit. The token will be passed to this traffic point and provides an additional chance for this traffic point to transmit a datagram successfully. Thus datagram loss should also be less than in the previous measurement.

As can be seen from Figure 6.11 the 1% FER requirement is met, using the configuration obtained from 1.1. Even more datagrams arrive successfully compared to results of measurement 3 of 1.1 and previous measurement, when using the same configuration. Because a better performance is measured, the priority slot interval is doubled to the exact datagram inter arrival time to investigate whether requirements can still be met. From Figure 6.11 appears that this is not the case. However if a FER of 3% was required this configuration would satisfy.

### 6.1.3 Measurements of scenario 1.3

**Measurement 1:**

Figure 6.12 shows the link throughput when both links serve FTP traffic. A clear asymmetric in throughput is observed from these results; the link throughput of link 1 is larger than for link 2.

![Link throughput (measurement 1 of scenario 1.3)](image)

Due to the fact that much time is lost by broadcast inquiries on link 2, the aggregate throughput in this scenario is lower than for 1.1. When a TDI of 2ms is considered, the maximum difference in aggregate throughput amounts 13% at 5% HER as is shown in Figure 6.13.
A modification is made to the flow charts that are presented in chapter 2 to prevent from the significant difference in link throughputs in this scenario. Since all nodes participating on a Bluetooth HR channel are still synchronized after a token pass provided by the supervisor, it is not necessary to issue a broadcast inquiry before starting a transmission to a node that is different from the supervisor node. This is useful in the case that a node receives a token but has only data for a node that is different from the node (supervisor) from whom it got the token. For example in the case that node 2 receives a token from node 0 (see Figure 5.3). Following the rules of the flow chart this node would issue a broadcast inquiry to verify if node 3 is still synchronized before sending data to node 3.

The results of the modification are shown in Figure 6.14 and Figure 6.15. From the first figure can be seen that the optimisation gives a considerable improvement on the link throughputs. The asymmetric that was experienced before is much less significant (maximum difference in link throughputs at TDI of 2ms and 5% HER from 30% to 5% thus up to 25% improvement for the measured TDI values).

Due to token pass overhead there is still difference in throughput on link 1 and 2. Since the supervisor node, when in turn in the round robin scheme, can directly start transmitting data, this node (and the link connected to it) has an advantage in the amount of available transmission time.

---

**Figure 6.13: Aggregate throughput (measurement 1 of scenario 1.3)**

The data shows the aggregate throughput (Mb/s) across different time intervals (TDI) and HER percentages. The table below summarizes the throughput values:

<table>
<thead>
<tr>
<th>TDI (ms)</th>
<th>HER (%)</th>
<th>Throughput (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDI 2ms</td>
<td>0</td>
<td>7.69</td>
</tr>
<tr>
<td>TDI 5ms</td>
<td>0</td>
<td>8.54</td>
</tr>
<tr>
<td>TDI 10ms</td>
<td>0</td>
<td>8.85</td>
</tr>
<tr>
<td>TDI 15ms</td>
<td>0</td>
<td>8.95</td>
</tr>
<tr>
<td>TDI 2ms (1.1)</td>
<td>0</td>
<td>8.58</td>
</tr>
<tr>
<td>TDI 5ms (1.1)</td>
<td>0</td>
<td>8.87</td>
</tr>
<tr>
<td>TDI 10ms (1.1)</td>
<td>0</td>
<td>9.01</td>
</tr>
<tr>
<td>TDI 15ms (1.1)</td>
<td>0</td>
<td>9.06</td>
</tr>
<tr>
<td>TDI 2ms (1.1)</td>
<td>1</td>
<td>8.58</td>
</tr>
<tr>
<td>TDI 5ms (1.1)</td>
<td>1</td>
<td>8.87</td>
</tr>
<tr>
<td>TDI 10ms (1.1)</td>
<td>1</td>
<td>9.01</td>
</tr>
<tr>
<td>TDI 15ms (1.1)</td>
<td>1</td>
<td>9.06</td>
</tr>
</tbody>
</table>

---

**Figure 6.14** and **Figure 6.15** provide visual representations of the aggregate throughput data.
### Figure 6.14: Link throughput (measurement 1 of scenario 1.3); after modification

<table>
<thead>
<tr>
<th>HER (%)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDI 2ms, link 1</td>
<td>4.29</td>
<td>3.85</td>
<td>3.55</td>
<td>3.28</td>
<td>3.03</td>
<td>2.76</td>
</tr>
<tr>
<td>TDI 5ms, link 1</td>
<td>4.43</td>
<td>3.94</td>
<td>3.53</td>
<td>3.10</td>
<td>2.75</td>
<td>2.39</td>
</tr>
<tr>
<td>TDI 10ms, link 1</td>
<td>4.51</td>
<td>3.81</td>
<td>3.23</td>
<td>2.66</td>
<td>2.21</td>
<td>1.81</td>
</tr>
<tr>
<td>TDI 15ms, link 1</td>
<td>4.53</td>
<td>3.68</td>
<td>2.96</td>
<td>2.33</td>
<td>1.79</td>
<td>1.42</td>
</tr>
<tr>
<td>TDI 2ms, link 2</td>
<td>4.09</td>
<td>3.75</td>
<td>3.45</td>
<td>3.15</td>
<td>2.87</td>
<td>2.63</td>
</tr>
<tr>
<td>TDI 5ms, link 2</td>
<td>4.39</td>
<td>3.31</td>
<td>3.44</td>
<td>3.03</td>
<td>2.65</td>
<td>2.31</td>
</tr>
<tr>
<td>TDI 10ms, link 2</td>
<td>4.48</td>
<td>3.80</td>
<td>3.13</td>
<td>2.63</td>
<td>2.18</td>
<td>1.77</td>
</tr>
<tr>
<td>TDI 15ms, link 2</td>
<td>4.52</td>
<td>3.65</td>
<td>2.93</td>
<td>2.26</td>
<td>1.79</td>
<td>1.40</td>
</tr>
</tbody>
</table>

### Figure 6.15: Aggregate throughput (measurement 1 of scenario 1.3); after modification

<table>
<thead>
<tr>
<th>HER (%)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDI 2ms</td>
<td>8.38</td>
<td>7.61</td>
<td>7.00</td>
<td>6.43</td>
<td>5.89</td>
<td>5.40</td>
</tr>
<tr>
<td>TDI 5ms</td>
<td>8.82</td>
<td>7.85</td>
<td>6.97</td>
<td>6.13</td>
<td>5.40</td>
<td>4.70</td>
</tr>
<tr>
<td>TDI 10ms</td>
<td>8.99</td>
<td>7.62</td>
<td>6.36</td>
<td>5.29</td>
<td>4.39</td>
<td>3.58</td>
</tr>
<tr>
<td>TDI 15ms</td>
<td>9.04</td>
<td>7.33</td>
<td>5.88</td>
<td>4.59</td>
<td>3.58</td>
<td>2.82</td>
</tr>
<tr>
<td>TDI 2ms (1.1)</td>
<td>8.58</td>
<td>7.72</td>
<td>7.12</td>
<td>6.58</td>
<td>6.04</td>
<td>5.56</td>
</tr>
<tr>
<td>TDI 5ms (1.1)</td>
<td>8.87</td>
<td>7.89</td>
<td>7.01</td>
<td>6.22</td>
<td>5.45</td>
<td>4.81</td>
</tr>
<tr>
<td>TDI 10ms (1.1)</td>
<td>9.01</td>
<td>7.67</td>
<td>6.40</td>
<td>5.30</td>
<td>4.40</td>
<td>3.64</td>
</tr>
<tr>
<td>TDI 15ms (1.1)</td>
<td>9.06</td>
<td>7.36</td>
<td>5.82</td>
<td>4.56</td>
<td>3.58</td>
<td>2.90</td>
</tr>
</tbody>
</table>
The maximum difference in aggregate throughput when comparing with scenario 1.1 is reduced to 3% at 5% HER, which is a 10% improvement for the case a TDI of 2ms is used.

Like in scenario 1.2 also here, now for a TDI of 15ms in combination with a HER of 2 and 3%, the aggregate throughput is higher (at most 0.06Mb/s) compared to scenario 1.1. The reasoning that is used in the previous section can also be applied here.

**Measurement 2:**
Like in measurement 3 of scenario 1.2 each link is assigned a priority slot with an offset that provides fair channel access for both links. The difference between the two measurements exists of the fact that in order to obtain the token for the second link at a supervisor priority slot, a token pass must take place in advance. Whereas both traffic source points in measurement 3 of scenario 1.2 can get immediately access to the channel when the source node is in turn in the round robin scheme. Like in the previous measurement the token pass modification is made to minimize protocol overhead by broadcast inquiries.

Results obtained from this measurement are comparative with those of measurement 3 of scenario 1.2. Also here the 1% FER requirement is met with the PS configuration obtained at scenario 1.1 and cannot be met when doubling the priority slot interval. The latter produces rather a 1% increase of datagram loss on both links at 5% HER compared to measurement results of measurement 3 of scenario 1.2. So that the less stringent FER of 3% for this scenario and configuration cannot be met.

![Figure 6.16: Successful datagram arrivals (measurement 2 of scenario 1.3)](image)

**6.1.4 Measurements of scenario 1.4**

**Measurement 1:**
The loss of aggregate throughput due to the idle link that is used in this measurement, increases for small token distribution intervals (2-5ms) as can be seen from Figure 6.18.
Figure 6.17: Link throughput (measurement 1 of scenario 1.4)

Figure 6.18: Aggregate throughput (measurement 1 of scenario 1.4)
For a TDI of 2ms the maximum difference in aggregate throughput amounts 21% at 5% HER compared to the same scenario without the idle link. The link throughputs of the links that are involved in data exchange are lower than presented for scenario 1.2 but traffic load is still fairly divided on both links. From the applied token distribution intervals 5ms is an optimal choice in the presence of idle links, as aggregate throughput for 0 to 5% HER is equal or greater than for a TDI of 2ms and aggregate throughput using a 10 and 15 ms TDI decreases in larger steps.

6.2 Measurements based on multicast scenarios
No results can be presented here since multicast is not implemented in the model at this moment.

6.3 Comparison Bluetooth HR and WLAN 802.11b
A comparison of the Bluetooth HR and WLAN 802.11b wireless technology is interesting, not only because of comparable data rates (12Mb/s for Bluetooth HR and 11 Mb/s for WLAN 802.11b) but also because there is an overlap in use cases (e.g. file transfer between two devices).

Although a more extensive comparison was envisaged, time did not permit this. Therefore the comparison that is reported here is limited to what is already mentioned in the literature and what is discussed in section 6.1. To make a fair comparison between both technologies also WLAN 802.11b should have been simulated on the Network Simulator for identical scenarios. However NS provided unreliable results for the WLAN measurements. Information about the WLAN set-up obtained from multiple sources on the Internet are examined [23, 24] and carried out, but did not lead to the expected (and by the sources described) behaviour. Because the research period didn't allow investing more time in getting a working WLAN 802.11b simulation in the Network Simulator, this comparison is restricted to Bluetooth HR simulation results and theoretical throughput of 802.11b in ideal channel conditions. Therefore the performance measurement of 802.11b is restricted to a theoretical maximum throughput research when considering a two-node (nodes are in short range) wireless local area network. Assumed is that the reader is familiar with the CSMA/CA and RTS/CTS protocol. An explanation of these protocols for WLAN 802.11b can be found in [12].

An experimental paper [10] describes the theoretical maximum throughput for the CSMA/CA protocol in 802.11b and the case in which the RTS/CTS protocol of 802.11b is applied. When the basic rate is 11Mb/s, payload is 1500 bytes and RTS/CTS is used, the theoretical maximum throughput is 4,52Mb/s. This throughput is higher when the CSMA/CA scheme is used without RTS/CTS; due to fewer control frames the theoretical maximum throughput in this case is 6,06Mb/s. From the results in this paper can be seen that the larger the packet size the higher the throughput. A conclusion quoted from this paper is that throughputs of over 6,1Mb/s are almost impossible to see in real deployments where IP packets carrying segments over 1500 bytes are not so common.

In [11] the overheads that cause the loss in throughput in 802.11b are described. The percentage of overhead in case of a gross data rate of 11Mb/s amounts to 53%. The overhead according to [11] consists of collisions, request frames, idle time (inter frame space), physical and MAC layer overhead and finally TCP overhead.
In the light of the above-mentioned findings, it appears that throughput that can be attained by Bluetooth HR is higher than that of 802.11b WLAN in case of a two-node wireless network under ideal channel conditions. HR provides up to 9,06 Mb/s with a TDI of 15 ms and 802.11b WLAN on the other hand 6,06 Mb/s as reported in [10]. When RTS/CTS is applied it reduces to 4,52 Mb/s, i.e. half of the throughput of Bluetooth HR.

6.4 Conclusions

Scenario 1.1: Maximum throughput that can be attained with Bluetooth HR amounts to 9,06 Mb/s. This is achieved with ideal channel conditions and a token distribution interval of 15 ms. When considering errors on the channel, dead time due to token loss shows a strong relation with the token distribution interval. Throughput decrease is largest for large TDI values. A decrease of 50% and 68% on throughput is observed compared to the throughput at 0% HER, for a HER of 3% and 5% respectively and a TDI of 15 ms. Based on measurement results, it can be concluded that a TDI of 2 to 5 ms appears to be an optimal trade-off between throughput at 0% HER and dead time in case of HER > 0%.

Bluetooth HR can be utilized for a high quality audio stream of 192 kb/s, for example between a mobile phone and a headset. A priority slot configuration is found that satisfies a common used high quality audio FER requirement of 1%; a priority slot interval that is equal to half the datagram inter arrival time and a token distribution interval that is almost double the datagram inter arrival time.

Scenario 1.2: Results of scenario 1.2 show that the scheduler described in chapter 2 provides fair channel access when one node serves two traffic streams on separate links. Aggregate throughput in case on both links FTP traffic is provided, is slightly lower compared to the link throughput of scenario 1.1. A difference of at most 0,31 Mb/s of aggregate throughput at 0% HER is observed, when a token distribution interval of 2 ms is used. The difference is caused by protocol overhead as additional broadcast inquiries and token passing take place.

In the case that two CBR links are considered, the 1% FER requirement for high quality audio on the CBR link can be met when the priority slot configuration is identical to the one obtained from scenario 1.1. Whereas doubling the priority slot interval does not meet the 1% requirement but satisfies a less stringent FER of 3%. From the FTP-CBR measurement appeared that the priority slot interval of this configuration has to be divided in half to meet a 1% FER requirement. Adapting the priority slot interval instead of the token distribution interval of the configuration obtained at 1.1 provides for a better performance, which means more successful datagram transmissions and higher aggregate throughput. The FTP-CBR measurement showed that Bluetooth HR is capable of serving time critical traffic (links with Quality of Service requirements) and best effort traffic at the same time without significant loss of performance with respect to aggregate throughput.

Scenario 1.3: Token passing has great impact on aggregate throughput in this scenario. Using an implementation for this scenario according to the token scheduling flow charts presented in chapter 2.5.1 leads to significant difference of link throughputs on both links. A difference of 30% between both link throughputs is observed for a token distribution interval of 2 ms at 5% HER. To prevent from this throughput difference an optimisation of the scheduler that is presented in chapter 2 is implemented. Using the optimisation the difference between both link throughputs decreases. Only a difference of at most 5% between both link throughputs for a TDI of 2 ms at 5% HER is observed.
For the case that both links serve CBR traffic the datagram loss results are compared with those obtained from scenario 1.2. Using the priority slot configuration obtained from scenario 1.1 the 1% FER requirement is met. Doubling the priority slot interval results in a FER that is larger than 3%.

Scenario 1.4: An idle link can result in up to 21% aggregate throughput decrease for a token distribution interval of 2ms, compared to scenario 1.2 in which the idle link is not present.

When comparing Bluetooth HR with another ad hoc wireless technology as WLAN 802.11b, research results [10] show that WLAN 802.11b throughput amounts at most 6.06Mb/s whereas Bluetooth HR throughput amounts 9.06Mb/s for a TDI of 15ms when considering ideal channel conditions. Low throughput of 802.11b is caused mainly by large overheads of MAC layer.
7 Conclusions and Recommendations

7.1 Conclusions
A model of the MAC/DLC layer of Bluetooth HR has been built using NS (Network Simulator). At this moment it is limited to simulating unicast scenarios. Multiple unicast links can be simulated, however.

The performance of the Bluetooth High Rate MAC/DLC layer with regard to throughput depends on the following parameters:
1) Token insertion rate
2) Number of priority slots defined on the channel
3) MAC layer overhead. Caused by broadcast inquiry, token passing, acknowledge requests and baseband layer acknowledges piggyback ARQ information on.
4) Dead time through header failures, which also implies token loss
5) Dead time due to segmentation introduced just before a priority slot. The Bluetooth HR ARQ algorithm does not allow fragmentation, which means splitting a Bluetooth HR segment in a size smaller than 128 bytes, unless it is the last segment of a datagram. So there may not be enough time available to transmit a (or more) complete segment(s) just before a priority slot.

A strong relation between dead time and token insertion rate is observed. Increasing the token insertion rate, results in a decreased dead time and increased throughput. Also unnecessary datagram fragmentation increases, which leads to additional overhead due to packet headers. However this overhead is not as high as dead time.

In Bluetooth HR the raw data rate is 12Mb/s. However 9.06Mb/s was the highest achieved user throughput in simulations (for a TDI of 15ms and ideal channel conditions) measured at TCP layer. The overhead including TCP/IP and HR MAC/DLC is thus less than 3Mb/s.

Increasing the token insertion rate does decrease throughput in ideal channel conditions. This decrease is marginal low (see Figure 6.1). Increased token insertion rate, however, significantly reduces throughput loss in case of header errors.

Simulations have shown that the impact of independent errors generation in single link scenarios is up to 68% throughput degradation in the case that TDI values equal or smaller than 15ms are considered and header error rate is varied from 0 to 5%.

Bluetooth HR is capable of serving time critical traffic (links with Quality of Service requirements) and best effort traffic at the same time. No significant loss of performance with respect to aggregate throughput is observed compared to the case that only best effort traffic is served.

When measuring performance on a link serving time critical traffic, e.g. voice or audio streaming, the most important parameter is the priority slot interval. The smaller the PSI the better the performance with respect to number of successful datagram arrivals. The number of successful datagram arrivals will increase when a link with priority traffic is added on the HR channel. The additional priority slot of the added link will provide for better performance. Because time critical traffic mostly consists of small datagrams,
much time is left after transmission initiated at a PS to serve another link with priority traffic.

A comparison with WLAN 802.11b is only performed for a two-node topology and ideal channel conditions. Bluetooth HR can provide a throughput of up to 9.06Mb/s. Whereas theoretical maximum throughput of 802.11b was reported to be 6.06Mb/s [10], which is only 2/3 of Bluetooth HR throughput.

7.2 Recommendations

Documentation of NS is very scarce in respect to describing how things are implemented. Wrong interpretation of the NS implementation has lead during this project to unnecessary errors in simulations multiple times. When performing a study into the influence of different TCP/IP parameters on Bluetooth HR a thorough study of implementation details should take place in advance.

7.3 Further study and open issues

Following issues could serve as input for further research:

- The performance of unicast scenarios is analysed in this report. Future studies should look into the performance of multicast scenarios.
- Focus in this report is mainly on the MAC/DLC layer of Bluetooth HR. A future study could focus on specific TCP/IP behaviour on the performance of Bluetooth HR.
- The results of a comparison between Bluetooth HR and WLAN 802.11b are only preliminary conclusions, which have been drawn from existing literature. More study has to be performed about how to configure a WLAN 802.11b simulation well. This enables a better comparison between Bluetooth HR and WLAN as the same topologies for both techniques can be used as input for the simulations. In this case it is also useful to take into account non-idealities on the channel, as a comparison would be valid in that case.
- A simplified priority slot contention resolution mechanism is implemented. An enhancement of the current implementation in respect to a more sophisticated priority slot contention resolution mechanism as specified in the draft specification [2] can be considered.
- Only FTP and CBR traffic generators are used for performance evaluation in this report. Other traffic generators, HTTP for instance, could be used.
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-DPSK</td>
<td>8-ary Differential Phase Shift Keying</td>
</tr>
<tr>
<td>ARQ</td>
<td>Automatic Repeat Request</td>
</tr>
<tr>
<td>BCP</td>
<td>Broadcast Control Point</td>
</tr>
<tr>
<td>BR</td>
<td>Basic Rate</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access / Collision Avoidance</td>
</tr>
<tr>
<td>DBPSK</td>
<td>Differential Binary Phase Shift Keying</td>
</tr>
<tr>
<td>DCP</td>
<td>Device Control Point</td>
</tr>
<tr>
<td>DCS</td>
<td>Dynamic Channel Selection</td>
</tr>
<tr>
<td>DIAT</td>
<td>Datagram Inter Arrival Time</td>
</tr>
<tr>
<td>DLC</td>
<td>Data Link Control</td>
</tr>
<tr>
<td>DQPSK</td>
<td>Differential Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Control</td>
</tr>
<tr>
<td>FER</td>
<td>Frame Error Rate</td>
</tr>
<tr>
<td>FH</td>
<td>Frequency Hopping</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>HER</td>
<td>Header Error Rate</td>
</tr>
<tr>
<td>HR</td>
<td>High Rate</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial Scientific Medical</td>
</tr>
<tr>
<td>L2CAP</td>
<td>Logical Link Control and Adaptation Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LCP</td>
<td>Logical Control Point</td>
</tr>
<tr>
<td>LTP</td>
<td>Logical Traffic Point</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MTP</td>
<td>Multicast Traffic Point</td>
</tr>
<tr>
<td>NS</td>
<td>Network Simulator</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>OTCL</td>
<td>Object oriented TCL</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PS</td>
<td>Priority Slot</td>
</tr>
<tr>
<td>PSI</td>
<td>Priority Slot Interval</td>
</tr>
<tr>
<td>QOS</td>
<td>Quality Of Service</td>
</tr>
<tr>
<td>RTS/CTS</td>
<td>Request To Send / Clear To Send</td>
</tr>
<tr>
<td>SER</td>
<td>Segment Error Rate</td>
</tr>
<tr>
<td>SIG</td>
<td>Special Interest Group</td>
</tr>
<tr>
<td>SPS</td>
<td>Supervisor Priority Slot</td>
</tr>
<tr>
<td>TCL</td>
<td>Tool Command Language</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>TDD</td>
<td>Time Division Duplexing</td>
</tr>
<tr>
<td>TDI</td>
<td>Token Distribution Interval</td>
</tr>
<tr>
<td>TP</td>
<td>Traffic Point</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UTP</td>
<td>Unicast Traffic Point</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
List of Figures
Figure 1.1: Bluetooth devices.................................................................9
Figure 2.1: Channel positioning............................................................13
Figure 2.2: FH channel with two HR subnets........................................14
Figure 2.3: Logical points in Bluetooth HR mode.................................15
Figure 2.4: Logical links in Bluetooth HR mode.................................16
Figure 2.5: High Rate packet format..................................................16
Figure 2.6: Payload segmenting............................................................17
Figure 2.7: Ping-pong protocol ...........................................................18
Figure 2.8: Ping-Pong protocol with priority scheme............................18
Figure 2.9: Token passing after priority slot........................................20
Figure 2.10: Token pass to traffic point or device control point ..........21
Figure 3.1: Protocol stack wireless node ............................................25
Figure 3.2: Extended protocol stack wireless nodes..............................29
Figure 3.3: Segmentation in (a) initial implementation and (b) modified implementation.................................32
Figure 3.4: Datagram latency on MAC layer........................................34
Figure 4.1: Channel C/I ..................................................................37
Figure 4.2: Packet length at different layers ........................................41
Figure 5.1: Topology of scenario 1.1....................................................44
Figure 5.2: Topology of scenario 1.2....................................................45
Figure 5.3: Topology of scenario 1.3....................................................46
Figure 5.4: Topology of scenario 1.4....................................................47
Figure 5.5: Topology of scenario 2.1....................................................47
Figure 5.6: Topologies of scenario 2.2..................................................48
Figure 5.7: Topologies of scenario 2.3..................................................48
Figure 6.1: Link throughput (measurement 1 of scenario 1.1).............51
Figure 6.2: Link throughput (measurement 1 of scenario 1.1); cross over area .......52
Figure 6.3: Link throughput (measurement 1 of scenario 1.1); small TDI values ........53
Figure 6.4: Link throughput (measurement 1 of scenario 1.1); throughput versus TDI...55
Figure 6.5: Successful datagram arrivals (measurement 2 of scenario 1.1) ......55
Figure 6.6: Successful datagram arrivals (measurement 3 of scenario 1.1) ......56
Figure 6.7: Link throughput (measurement 1 of scenario 1.2) ............57
Figure 6.8: Aggregate throughput (measurement 1 of scenario 1.2).........58
Figure 6.9: Successful datagram arrivals (measurement 2 of scenario 1.2) ....59
Figure 6.10: Aggregate throughput (measurement 2 of scenario 1.2) ........60
Figure 6.11: Successful datagram arrivals (measurement 3 of scenario 1.2) ....60
Figure 6.12: Link throughput (measurement 1 of scenario 1.3) .............61
Figure 6.13: Aggregate throughput (measurement 1 of scenario 1.3)........62
Figure 6.14: Link throughput (measurement 1 of scenario 1.3); after modification ......63
Figure 6.15: Aggregate throughput (measurement 1 of scenario 1.3); after modification ..........63
Figure 6.16: Successful datagram arrivals (measurement 2 of scenario 1.3) ....64
Figure 6.17: Link throughput (measurement 1 of scenario 1.4) .............65
Figure 6.18: Aggregate throughput (measurement 1 of scenario 1.4) ........65
List of Tables

Table 3.1: Analytical and measurement results ......................................................... 32
Table 4.1: Traffic generators ..................................................................................... 38
Table 6.1: Traffic sequences and respective throughputs .......................................... 54
References

[1] Haartsen, J.C.
*Bluetooth – The universal radio interface for ad hoc, wireless connectivity*

[2] Bluetooth SIG
*Bluetooth High Rate Mode Baseband Draft Specification*
Bluetooth SIG Radio Working Group, SIG confidential, version 0.65, 26-02-2003

[3] Bluetooth SIG
*Bluetooth High Rate Mode Physical Layer Draft Specification*
Bluetooth SIG Radio Working Group, SIG confidential

[4] Bluetooth SIG
*Specification of the Bluetooth System*
Bluetooth SIG, Version 1.0B, 01-12-1999

[5] Bluetooth SIG
*Specification of the Bluetooth System*
Bluetooth SIG, Version 1.1, 22-02-2001

[6] Jansen, B.
*Modelling Bluetooth High Rate behaviour in NS*
Ericsson, Limited internal report, 07-01-2003

[7] Fall, K. and K. Varadhan
*The NS manual (formerly NS Notes and Documentation)*
http://www.isi.edu/nsnam/ns/ns-documentation.html, April 2002
date last check: 14 November 2003

[8] Tanenbaum, A.S.
*Computer Networks*
Upper Saddle River: Prentice Hall, 2002

[9] Zürbes, S.
*Header and Segment Failure Correlations of Bluetooth High Rate Mode Interfered by Bluetooth V1*
Ericsson internal, No. RN-01:063

*Theoretical Maximum Throughput of IEEE 802.11 and its Applications*
IEEE 2nd Int. Symp. on Netw. Comp. and App. (NCA 2003), April 2003, p. 249-256

*Net throughput with IEEE 802.11 Wireless LANs*

[12] 3com
*IEEE 802.11b Wireless LANs Wireless Freedom at Ethernet Speeds*
http://www.3com.com/technology/tech_net/white_papers/503072.html
date last check: 14 November 2003

[13] Rappaport, T.S.
*Wireless communications, principles and practice*
[14] Morrow, R.
*Bluetooth operation and use*

*The Bluetooth Radio System*
IEEE Personal Communications, Vol. 7(2000), No.1, p. 28-36

[16] Taori, R.
*Positioning Bluetooth High Rate*
Ericsson internal, Strictly confidential report, No. RT02-014, 07-07-2002

*Simplifications to the HR MAC*
Bluetooth SIG, Confidential report, RT02-013, 21-05-2002

*Supervisor Role*
Bluetooth SIG, Confidential report, No. RT02-026, 06-01-2003

[19] Taori, R.
*High Rate MAC/DLC Simulation Results*
Ericsson internal, Confidential report, RT02-013, 29-10-2002

[20] Bisdikian, C.
*An overview of the Bluetooth wireless technology*

*Personal Area Networks: Bluetooth or IEEE 802.11?*

*Short Range Radio Based Ad-hoc networking performance and properties*
IEEE Int. Conf. on Communications, Vol. 3(1999), p. 1414-1420

[23] The Ns-users Archives
[ns] max achievable throughput with 11Mbps 802.11b?
date last check: 14 November 2003

[24] The Ns-users Archives
[ns] Low 802.11 throughput
http://mailman.isi.edu/pipermail/ns-users/2003-June/033602.html
date last check: 14 November 2003

[25] 3GPP TSG-SA working group 1 (services)
*QoS performance requirements for UMTS 1 user requirements*
http://www.3gpp.org/ftp/tsg_sa/WG1_Serv/TSGS1_03-HCourt/Docs/Docs/s1-99362.pdf
date last check: 14 November 2003
Appendix A: An example NS script

Following script represents scenario 1.2 described in chapter 5 of this report.

```
# Copyright (c) 1997 Regents of the University of California.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
# 1. Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in the
#    documentation and/or other materials provided with the distribution.
# 3. All advertising materials mentioning features or use of this software
#    must display the following acknowledgement:
#       This product includes software developed by the Computer Systems
#       Engineering Group at Lawrence Berkeley Laboratory.
# 4. Neither the name of the University nor of the Laboratory may be used
#    to endorse or promote products derived from this software without
#    specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
# ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
# OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
# LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
# OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
#
# Define options
# ----------------------------------------------------------------------
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/BluetoothHr ;# MAC type
set val(ifq) Queue/DropTail/PriQueue;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 3 ;# number of mobilenodes
set val(rp) DSDV ;# routing protocol

# Main Program
# ----------------------------------------------------------------------

# Bluetooth High Rate parameters
if {$val(mac) == "Mac/BluetoothHr"} {
    puts "Simulating Bluetooth High Rate MAC\n"
    # set wireless channel properties
```
Phy/WirelessPhy set CPThresh_ 10.0 ;# capture threshold (dB)
Phy/WirelessPhy set CSThresh_ 1.559e-11 ;# carrier sense threshold (W)
Phy/WirelessPhy set RXThresh_ 9e-10 ;# receive power threshold (W)
Phy/WirelessPhy set Pt_ 1e-3 ;# transmit power is 1 mW
Phy/WirelessPhy set freq_ 2.45e9 ;# frequency is 2.45 Ghz
Phy/WirelessPhy set L_ 1.0 ;# system loss factor

# set queues to only send packets when polled
Queue set unblock_on_resume_ false
Queue set blocked_ true

# Initialize Global Variables
#
set ns_ [new Simulator]
set tracefd [open hr.tr w]
$ns_ trace-all $tracefd

# set up topography object
set topo [new Topography]
$topo load_flatgrid 500 500

# Create God (only used in IEEE 802.11)
#
create-god $val(nn)

# Create Supervisor (Bluetooth High Rate only)
# and schedule the install of the priority slots
#
if {$val(mac) eq "Mac/BluetoothHr"} {
    set supervisor_ [new Supervisor]
    $supervisor_ supervisornode 1
    #
    # configure Logical Points and Logical Links
    #
    $supervisor_ setup_node 0 1 255 254 252
    $supervisor_ setup_node 1 2 255 253
    $supervisor_ setup_node 2 3 255 251
    $supervisor_ setup_unicast_link 254 253 2 0 0 0.0 0.0
    $supervisor_ setup_unicast_link 252 251 2 1 10 0.026125 2.0
    $supervisor_ setup_control_links $val(nn)
    #
    # configure priority slots
    #
    $supervisor_ setup_priority_slot 1 100 1600
    $supervisor_ setup_priority_slot 252 0 400
    $ns_ at 0.0 "$supervisor_ install_priority_slots"
    #
    # in case packet delay must be measured;
    # specify the MAC address of calculating nodes
    #
    $supervisor_ latency_measurement_at_nodes 0 1
}
Create the specified number of mobilenodes [$val(nn)$] and "attach" them to the channel.
Here three nodes are created: node(0), node(1) and node(2)

```bash
$ns_node-config -adhocRouting $val(rp) \
  -llType $val(ll) \n  -macType $val(mac) \n  -ifqType $val(ifq) \n  -ifqLen $val(ifqlen) \n  -antType $val(ant) \n  -propType $val(prop) \n  -phyType $val(netif) \n  -channelType $val(chan) \n  -topolInstance $topo \n  -agentTrace OFF \n  -routerTrace ON \n  -macTrace ON \n  -movementTrace OFF \n  -IncomingErrProc UniformErr \n  -OutgoingErrProc UniformErr
```

Configure Bluetooth error model

```bash
proc UniformErr {} {
  set err [new BTErrorModel]
  $err rate 0.03
  $err enable 1
  $err unit packet_segment
  return $err
}
```

```bash
for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns_node]
  # disable random motion
  $node_($i) random-motion 0
}
```

Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes

```bash
$node_(0) set X 5.0
$node_(0) set Y 2.0
$node_(0) set Z 0.0
$node_(1) set X 5.0
$node_(1) set Y 3.0
$node_(1) set Z 0.0
$node_(2) set X 5.0
$node_(2) set Y 1.0
$node_(2) set Z 0.0
```

Setup traffic flow between nodes
# TCP connection between node_(0) and node_(2)
# CBR connection between node_(0) and node_(2)

set tcp [new Agent/TCP]
$tcp set class_ 2
$tcp set packetSize_ 1500
$tcp set tcpip_base_hdr_size_ 20
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp
$ns_ attach-agent $node_(1) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 2.0 "$ftp start"
$ns_ at 122.0 "$ftp stop"

set udp [new Agent/UDP]
$udp set udp_base_hdr_size_ 8
set null [new Agent/Null]
$ns_ attach-agent $node_(0) $udp
$ns_ attach-agent $node_(2) $null
$ns_ connect $udp $null
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetSize_ 627
$cbr set rate_ 192Kb
$cbr set random_ 0

$ns_ at 2.0 "$cbr start"
$ns_ at 122.0 "$cbr stop"

# Tell nodes when the simulation ends
#
# for (set i 0} ($i < $val(nn) ) (incr i) {  
  $ns_ at 125.0 "$node_($i) reset";
}
$ns_ at 125.0 "stop"
$ns_ at 125.0 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop () {
  global ns_tracefd
  ns_flush_trace
  close $tracefd
}
puts "Starting Simulation..."
$ns_run
Explanation of the script:

1. **Specifying the number of nodes**
   NS has to be provided the number of nodes in the to be simulated scenario. With the command `set val(nn) 3` the number of nodes in this example is set to 3.

2. **Physical layer configuration**
   In the first part of the script the Bluetooth HR parameters have to be provided that are needed for physical layer configuration.

3. **Initialising global variables**
   At this part a new simulator object ($ns$), trace file and topology grid are set up and initialised.

4. **Creating and configuring the supervisor**
   One of the nodes in the to be simulated topology has to be assigned the supervisor function. Using the command `set supervisor_ [new Supervisor]` a new supervisor object will be constructed. Next the mapping of this supervisor object to a node in the topology takes place using the command `$supervisor_ supervisor_node 1` which means that node with DCP 1 will act as supervisor in this simulation. In real life the supervisor will set up and broadcast any changes in network configuration, this is not happening in the simulation. All nodes can request at all times topology information at the supervisor. Topology configuration is done by two commands that will inform the supervisor of the setup.

   ```
   $supervisor_ setup_node 0 1 255 254 252
   0: MAC address of this node that is used in NS for addressing by the traffic generators.
   1: Unique Device Control Point address for this node ranging from 1 to 255
   255: Multicast traffic point ranging from 255 to 1 (although it is maybe not used in the simulation it has to be specified)
   254/252: Unique unicast traffic point ranging from 255 to 1 (also possible to specify just one traffic point instead of two in this example, however more than two is not provided in the current implementation)
   ```

   ```
   $supervisor_ setup_unicast_link 252 251 2 1 10 0.026125 2.0
   252: source traffic point of link
   251: destination traffic point of link
   2: modulation method used:
<table>
<thead>
<tr>
<th>Option</th>
<th>Modulation method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DBPSK</td>
</tr>
<tr>
<td>1</td>
<td>DQPSK</td>
</tr>
<tr>
<td>2</td>
<td>8-DPSK</td>
</tr>
</tbody>
</table>
   1: unreliable service enabled (0 for reliable service, if enabled the following three options can be specified using 0 or 0.0)
   10: number of retransmissions that are allowed for unreliable service
   0.026125: Datagram inter arrival time in seconds. This value added to the timestamp that is set when a datagram enters the queue represents the expire time
of that datagram. Retransmissions of this datagram will not take place after the expire time, instead the datagram will be flushed.

2.0: Time in seconds that the traffic generator, which generates packets for this link, starts generating datagrams.

Next the priority slots are set up:
$supervisor_setup_priority_slot 1 100 1600
$supervisor_setup_priority_slot 252 0 400

At the first line the Supervisor Priority Slot is set up for DCP 1 with an offset of 100 Bluetooth HR timeslots and a priority slot interval of 1600 time slots. At the second line a priority slot for traffic point 252 is set up. For this priority slot no offset is defined and the priority slot interval is equal to 400 Bluetooth HR timeslots. Currently it is possible to specify up to 4 priority slots (including the supervisor priority slot).

When simulating one or more links, which serve time critical traffic as for example constant bit rate traffic, it is possible to measure the latency that datagrams experience on a certain link. Using the following command it is possible to specify between which nodes the latency has to be recorded. Recording means that the latency of each datagram is written to a file (expire_timestampsX.txt where X stands for the link number).
$supervisor_latency_measurement_at_nodes 0 1

For each link the node pair has to be specified with their respective MAC addresses. The order of the specified addresses determines the direction in which datagrams will be recorded. So in this example the delay of all datagrams that are sent from 0 to 1 are recorded. Currently it is possible to specify up to 3 links (or 3 node pairs). Link numbers are assigned according to the order in which node pairs are specified (from left to right).

5. Configuration of nodes
In this part of the script each node will be constructed. The architecture of a wireless node is given in section 3.4 of this report. The error generator that is part of each wireless node can be configured with the following commands:
$err rate 0.03
The header error rate or segment error rate of the generator. When using the unit packet_segment, see below, the rate presents the header error rate while the segment error rate is automatically set to twice this value (6% in this example).
$err enable 1
Enable or disable the error generator
$err unit packet_segment
Specify the unit on which errors are generated. The following options are possible:

<table>
<thead>
<tr>
<th>unit</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packet</td>
<td>Only packet header errors</td>
</tr>
<tr>
<td>segment</td>
<td>Only segment header errors</td>
</tr>
<tr>
<td>packet_segment</td>
<td>Both packet and segment header errors</td>
</tr>
</tbody>
</table>

6. Initial coordinates of the wireless nodes
These coordinates are static during the simulation.

7. Set up traffic flow between nodes
For each traffic generator the type of a so-called transport layer agent has to be provided. Only TCP and UDP protocol are used for simulations described in this report. The header sizes of the respective transport layer protocol datagrams can be specified.
Each transport layer agent needs a receiving agent (sink/null). Next the TCP/UDP agent and the sink/null agent have to be attached to a node in the specified topology. Also needs TCP to be connected to the sink and UDP to be connected to the null. When the transport layer is set up, the desired application, which is also an agent in NS, is configured and connected to the transport layer agent. Finally the traffic generator is provided start and finish time. Because there is a lot of set up traffic in the first two seconds, the start time of the traffic generators in the example is set to 2 seconds.

8. **End the simulation**
Using the command \$ns_ at 125.0 "stop" the network simulator is forced to end the simulation at the specified time. In addition each node has to be reset at the end of the simulation.