NTNU
Innovation and Creativity

WISA vs. WLAN: Co-existence challenges

- A tool for WLAN performance testing

Erlend Barstad Strand

Master of Science in Communication Technology
Submission date: June 2007
Supervisor: Geir Egil Øien, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

WISA – Wireless Interface to Sensors and Actuators - is ABB's proprietary 1Mbit/s wireless protocol for industrial automation. It operates within the 2.4GHz ISM band. WISA targets manufacturing & robotics applications (i.e. not process automation). WISA communication is based on Frequency Hopping (FH), Frequency Division Duplex (FDD), and Time Division Multiple Access (TDMA).

Wireless Local Area Networks (WLANs) - which typically occupy a fixed portion of the same 2.4 GHz ISM band - are also becoming more and more common on the factory floor. One example is bar-code scanners. Customers need confirmation that WISA can co-exist with other systems – particularly WLANs – without degrading their performance 'too much' due to interference. Preferably they would like WISA to have some form of adaptive frequency hopping, in order to be able to detect any other equipment that occupies part(s) of the 2.4GHz band – and stay away from those frequencies. WISA today has frequency hopping which spreads traffic uniformly over 77MHz of the 83MHz wide band. However it cannot at present - adaptively or otherwise - avoid any part of the band.

The objective of this project is to develop a tool which will aid the future testing of WLAN performance when interfered by WISA. The tool should be a combination of hardware and software for generation of WLAN traffic and interfering WISA signals. Relevant measures of the WLAN traffic should be observable and recorded by the tool.

Assignment given: 23. January 2007
Supervisor: Geir Egil Øien, IET

Acknowledgements

This report is the result of Erlend Barstad Strand's master thesis at the Norwegian University of Science and Technology, Department of Electronics and Telecommunications in the spring 2007. The report is written on the instructions of ABB Corporate Research, Billingstad, Norway.

A special thanks goes to MSc. Anne Elisabeth Vallestad at ABB Corporate Research, Billingstad, Norway, for her invaluable guidance and inspiration in the process of this diploma project. Also Dr. Dacfey Dzung at ABB Corporate Research, Switzerland, should be thanked for his helpful technical guidance and Dr. Dagfin Brodtkorb at ABB Corporate Research, Billingstad, Norway, for advice on radio equipment and signal strengths. Finally a special thanks goes to Professor Geir. E. Øien, representing the Department of Electronics and Telecommunications at NTNU.

Erlend Barstad Strand
June, 2007, Trondheim

Abstract

Wireless Interface for Sensors and Actuators (WISA) is ABB's proprietary wireless protocol for industrial automation on the factory floor. It operates in the 2.4GHz ISM band. Wireless Local Area Networks (WLANs), which typically occupy a fixed portion of the same 2.4GHz ISM band, are becoming more and more common on the factory floor. This raises a question of co-existence and how the performance of traffic over WLAN is affected when interfered by WISA.

This report is a result of the development of a software tool and assembly of hardware that can aid the future testing of the effect WISA has on nearby WLANs. Together with the explanation of the usage of this software tool, this report will also investigate different arrangements of hardware components that are used to demonstrate and test the functionality of this new software tool.

The software tool and the hardware components enable the measurement of important traffic metrics between two computers that communicate over a WLAN. The hardware components include a WISA Base Station (BS) that is configurable through the software tool and is used to cause interference on the WLAN.

Contents

List of Figures

List of Tables

Abbreviations

AP Access Point

BS Base Station

FDD Frequency Division Duplex

FH Frequency Hopping

GUI Graphical User Interface

ISM Industrial, Scientific and Medical

MAC Medium Access Control

OFDM Orthogonal Frequency Division Multiplexing

PHY Physical Layer

QAM Quadrature Amplitude Modulation

SA Sensors and Actuators

TCP Transport Control Protocol

TDMA Time Division Multiple Access

UDP User Datagram Protocol

WISA Wireless Interface for Sensors and Actuators

WLAN Wireless Local Area Network

Chapter 1

Introduction

1.1 Background

Wireless Interface for Sensors and Actuators (WISA) is ABB's proprietary wireless protocol for industrial automation on the factory floor. It operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band. So do most Wireless Local Area Network (WLAN) systems, and WLANs are becoming more and more common on the factory floor. The chance of co-located WISA and WLAN installations is therefore increasing.

WISA does frequency-hopping over most of the ISM band, but has currently no means of avoiding parts of the band occupied by other wireless systems. ABB will benefit from a test system that can investigate how the current, and new, frequency hopping strategies will affect other systems, in particular WLANs.

True adaptive frequency hopping consists of detection of other wireless systems as well as avoidance of the frequencies occupied by those systems. Only the latter is considered here, i.e. it is assumed that the user knows which parts of the band he wants WISA to avoid.

1.2 Objective and scope

The objective of this diploma project has been to develop a **software tool** which can aid the future testing of the effect of WISA on nearby Wireless LANs.

Along with the new software tool, a combination of hardware and off-the-shelf software modules have been assembled. Together, this forms a **test setup** that can generate typical WLAN traffic, vary the WISA frequency-hopping strategy,

and measure the effect of WISA on given WLAN traffic parameters.

The tasks of the new software tool are to capture WLAN and WISA parameters from the user, control the hardware and the standard software accordingly, run measurements, capture relevant results on-the-fly, and present the results to the user.

Doing actual performance measurements is not part of the scope. Even so, a few measurements have been carried out, in order to demonstrate the software tool and ensure that the hardware/software setup works as intended.

The test setup will include two computers for generation and reception of network traffic. They will be connected by a network consisting of a combination of Ethernet and WLAN links. The WLAN link will be interfered by a signal generator generating WISA signals. Part of the project is to find suitable hardware and to assemble them to form a test setup similar to the one illustrated in figure 1.1.

The delay jitter, bandwidth and packet loss of the network traffic between the

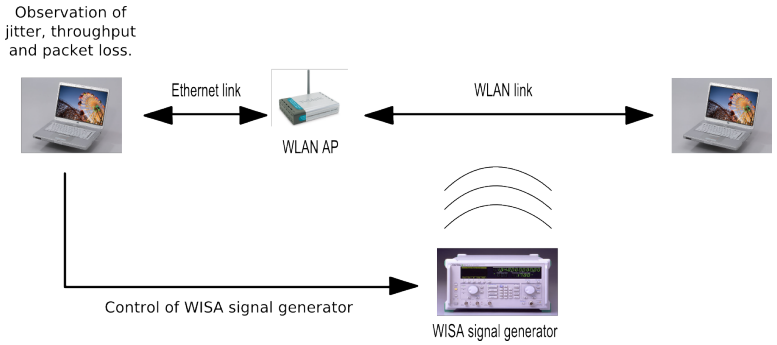


Figure 1.1: An assembly of hardware for testing the performance of WLAN interfered by WISA.

computers will be measured and recorded by the software tool. Also the data rate used by the WLAN Access Point (AP) connecting the computers, will be observable and recorded. These are the metrics used to test the performance of the WLAN.

Ideally it should have been possible to adjust the signalling strength of the signal sources through the software tool to see what effect it might have had on the WLAN performance. Due to lack of time this functionality was not implemented, but adjusting the signal strengths can also be done by using attenua-

tors attached to the signal sources.

1.3 Outline

Chapter 2 - Theory Theoretical information about the two wireless technologies and some information of the WLAN performance to be expected.

Chapter 3 - Hardware Explanation of how the different hardware components are connected and a short description of the most important components.

Chapter 4 - Software Tool This chapter will give the reader a thorough description of how to use the software tool for testing the performance of the network connecting two computers. Also some dependent software will be presented.

Chapter 5 - Sample Tool Results Results produced by the developed software will be presented in this chapter.

Chapter 6 - Discussion Discussion of the software and hardware, and its suitability for evaluating the performance of WLAN traffic.

Chapter 7 - Conclusion Final comments of the diploma project and this report.

Chapter 2

Theory

In this chapter the two wireless technologies WISA and WLAN will be presented. The purpose of this is not to give the reader a complete understanding of these technologies, but to give enough information for the reader to see that co-existence is a challenge. There will also be a brief introduction to the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP), since these protocols will be used to test the performance of a WLAN.

2.1 Introduction to WISA

This section is an abstract of [1] and [2], and will give an short introduction to the WISA technology.

2.1.1 Overview of WISA

Factory automation today usually depends on a large number of Sensors and Actuators (SA). All the SA's require a power supply and a way to communicate, usually by cables. These cables are not only costly when considering the installation and acquisition, but are also a source of failure of communication and power supply. Industry environments also increase the unreliability of cables because of their harsh environmental conditions.

Several existing wireless communication systems are both relatively inexpensive and well tested, such as Bluetooth, WLAN and ZigBee. However all these systems have drawbacks that make them unsuitable as a replacement for the traditional cabled communication systems. None of the mentioned systems meet all the requirements that SA's have to a communication system such as latency, data rate, reliability, power consumption, node density and range.

WISA is a wireless communication system designed to circumvent the problems associated with a cabled system and to meet the requirements that SA's have for communication in a closed control loop. WISA basically consists of two main parts:

- Communication (WISA-COM)
- Power Supply (WISA-POWER)

WISA-COM links the SA's to a Base Station (BS). This system has high reliability, fast response time, high density of SA's (hundreds within a radius of up to 10 meters) and guarantees high data transmission integrity, even when radio propagation is impaired by interference and shadowing. The response time is the time elapsed from a SA has detected an event to the BS's reception of a message of this particular event. This response time is typically 5ms, but up to 20ms in worst-case scenarios. A more in-depth study of WISA-COM will be given in section 2.1.2

WISA-POWER provides wireless power supply to the SA's and makes them truly wireless. This power system operates with magnetic fields, similar to RFID and anti-theft devices and provides power in a similar fashion to a transformer, but without a core and with a huge air-gap. WISA-POWER will not be discussed any further in this thesis.

2.1.2 WISA Communication

WISA operates in the license free 2.4GHz Industrial, Scientific and Medical (ISM) frequency band. The frequency band available ranges from 2.4GHz to 2.4835GHz, a total of 83.5MHz. WISA, however utilizes only the frequencies from 2.4025GHz to 2.4795GHz. This band is divided into 77 channels, spaced 1MHz apart. The channels are numbered from 3 to 79 and the number indicates the centre frequency of the channel. An example is channel 3 that is centred at 2.403GHz.

The channels are grouped into 7 sub-bands, each containing 11 consecutive channels. See figure 2.1 for an illustration of the frequency use and channel allocation.

Wireless communication systems are always at risk of being interfered by other wireless systems or frequency generating devices (e.g. microwave), or to suffer from frequency selective fading. To combat such threats WISA uses techniques

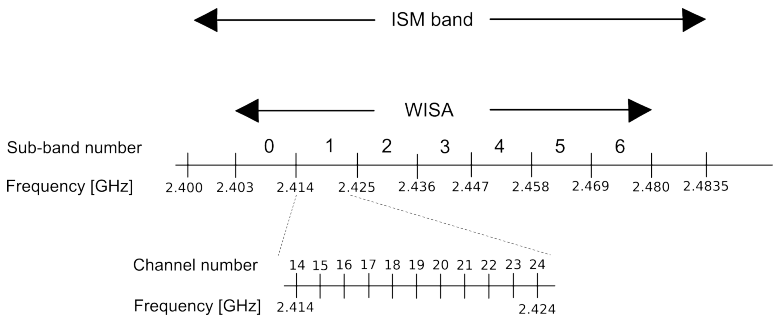


Figure 2.1: An illustration of the WISA channel allocation.

that exploit both frequency and space diversity. Frequency diversity is achieved by using Frequency Hopping (FH), while space diversity is realized by using multiple antennas.

Being able to support a high density of SA's and real-time communication is a challenge when the available frequency is limited. This is solved in WISA by deploying a cell based network topology and assigning each cell a specific FH sequence. Cells that are spaced sufficiently apart so they do not interfere with each other, may use the same FH sequence. Neighbouring cells use FH sequences that cause low self interference.

2.1.3 Physical Layer of WISA

The physical layer of WISA is very similar to the IEEE 802.15.1 (Bluetooth) standard. The modulation is a Gaussian-shaped Frequency Shift Keying (GFSK) with a Bandwidth Time Product (BT) of 0.5. The modulation index is between 0.28 and 0.35 and causes a frequency deviation of 140kHz to 175kHz. The modulation allows a data rate of 1Mbit/sec for both uplink and downlink.

The transmission power is nominally 0dBm and complies with regulations for the 2.4GHz ISM band. A single channel will appear on a spectrum analyzer as in figure 2.2.

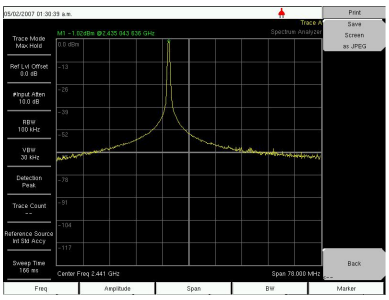


Figure 2.2: The spectrum of WISA channel 35.

2.1.4 Medium Access Control of WISA

The intention of WISA to allow reliable communication, in real-time, for a high number of devices demands a strict medium access control. WISA uses a combination of Time Division Multiple Access (TDMA), Frequency Division Duplex (FDD) and Frequency Hopping (FH) to accomplish this.

The WISA Medium Access Control (MAC) divides the time into frames and each frame is further divided into slots. The downlink communication, from the BS to its associated SA's is always on, and every slot in each frame is used. The uplink however is shared between all the SA's, up to 120 devices. This implies that an SA can only occupy the medium for a short period of time during a frame in order to share the medium with the other SA's. The SA's are organised into 4 groups and each group is assigned a channel, this allows 4 SA's to transmit simultaneously in a slot.

The downlink is used for configuring the SA's access to the medium, by assigning them one uplink slot per frame. Acknowledgement of an SA's successful transmission of data is also accommodated on the downlink.

Uplink details:

- The TDMA frame lasts for 2048μs and is divided into 32 slots, each lasting 64μs.
- An SA may transmit only for one slot per TDMA frame.
- An SA can optionally be assigned a Dslot (Double slot) lasting 128μs for increased capacity. As for slots, the SA may transmit only for one Dslot per TDMA frame.
- The last Dslot of the frame is used for tuning to the next frequency in the

FH sequence.

- The uplink frequencies are different from the downlink frequencies (FDD).

Downlink details:

- The TDMA frame lasts for 2048μs and is divided into 16 Dslots, each lasting for 128μs.
- The downlink Dslot contains slot number and payload data for 8 SA's.
- The last Dslot of the frame is used for tuning to the next frequency in the FH sequence.

As mentioned in section 2.1.2, WISA uses a cell based network topology. Each cell uses its own hopping sequence based on a cell identity, the cell identity being an integer ranging from 0 to 59. The sequences allow up to 6 cells to be located close together with acceptable self-interference.

Some properties of the FH sequence:

- There are 60 different sequences, and they are periodic with a period of 77 frames.
- The frequencies in consecutive frames are in different sub-bands.
- The uplink and downlink are spaced 3 sub-bands apart, i.e. at least 22MHz.
- Concurrent uplinks are in the same sub-band and spaced at least 2MHz (mostly 3MHz) apart.

For a BS with cell identity 19, the 10 first frequencies of its hopping sequence would be as listed in table 2.1. More in-depth information of the WISA system can be found in [1] and [2].

Channel no.	Sub-band no.	Frequency [GHz]
3	0	2.403
36	2	2.436
56	4	2.456
77	6	2.477
21	1	2.421
42	3	2.442
62	5	2.462
6	0	2.406
26	2	2.426
47	4	2.447

Table 2.1: Frequency Hopping sequence for a Base Station with cell identity 19

2.2 Introduction to WLAN

A Wireless Local Area Network (WLAN) enables communication between two or more computers without using wires. Part of this project's objective will be to develop software for testing the performance of a specific WLAN standard developed by the IEEE 802.11 working group. The results presented later in this report are produced by WLAN equipment using the IEEE 802.11g standard. The following section will present this standard.

2.2.1 Overview of IEEE 802.11

The original specification was released by the IEEE 802.11 working group in 1997 and allowed data rates of 1Mbit/sec or 2Mbit/sec. The standard uses the 2.4GHz ISM band and is the called IEEE 802.11 standard. In 1999 both the 802.11a and the 802.11b standards were released. The 802.11a operates in an unlicensed band around 5GHz and use Orthogonal Frequency Division Multiplexing (OFDM) with Quadrature Amplitude Modulation (QAM) as sub-carrier modulation and can support data rates up to 54Mbit/sec. The 802.11b standard use the same band as the original 802.11 standard but makes use of a different modulation to support data rates of 5.5Mbit/sec and 11Mbit/sec in addition to the original rates.

IEEE 802.11g was released in 2003, and is backwards compatible with 802.11b while being able to operate at the same rates as 802.11a. The standard is a collection of several Physical Layer (PHY) specifications, some are optional. To be backwards compatible with 802.11b it must support the same PHY specifi-

cations. To reach data rates of 54Mbit/sec it practically uses the same PHY as 802.11a, but in the 2.4GHz band instead. The equipment used for this project use the 802.11g standard, but only the mandatory PHY's, therefore the remaining sections will focus only on the mandatory PHY's of 802.11g.

The following sections is an abstract of the most important features of the IEEE 802.11g standard found in [3],[4] and [5].

2.2.2 Spectrum Usage of IEEE 802.11g

The Physical Layer (PHY) of IEEE 802.11g uses the same unlicensed Industrial, Scientific and Medical (ISM) frequency band as WISA, ranging from 2.4GHz to 2.4385GHz. This band it divided into 13 channels[1], each spaced 5MHz apart as listed in table 2.2. Each channel is further divided into 52 subcarriers using

Channel no.	Frequency [GHz]
1	2.412
2	2.417
3	2.422
4	2.427
5	2.432
6	2.437
7	2.442
8	2.447
9	2.452
10	2.457
11	2.462
12	2.467
13	2.472

Table 2.2: The 802.11g channel frequencies.

OFDM. The subcarrier modulation is rate dependent and vary between BPSK, QPSK, 16-QAM and 64-QAM. Coding is used to provide error correction and various coding rates are used for varying protection on top of the different constellations.

The standard specifies a transmit spectrum mask (see figure 2.3), which regulates the transmitted spectral density density of the signal. From this mask one can observe that a channel occupies a total of 60MHz. The standard however

[1] This is country dependent, but for most of Europe this is true. For more details see [4]

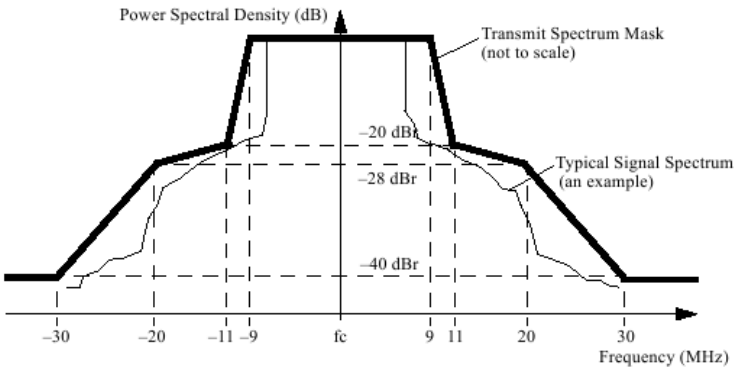


Figure 2.3: The spectrum mask of 802.11g OFDM symbols.

defines adjacent channels to be placed at $\pm$25MHz, this enables 3 channels to operate simultaneously in the available frequency band with the channel allocation as specified in table 2.2.

ABB has commenced work on developing new adaptive FH for the WISA system, where the objective is to avoid a WLAN channel. This work suggest that only 17MHz of a WLAN channel should be avoided by WISA [8], therefore a maximum of only 17 WISA channels is assumed interfering with a single WLAN channel.

To illustrate how a WISA BS may cause interference on a WLAN one can make a simple calculation of the percentage of time a BS will transmit in the spectrum of a WLAN channel. A WLAN using channel 1 (this is the channel used later) would occupy the spectrum from 2.404GHz to 2.422GHz. The WISA channels that occupy the same spectrum and may cause interference, are the channels from 4 to 22, 19 channel out of a total of 77. This means that approximately 25% of the time the WLAN will be interfered by a WISA BS. This result can not be used directly to predict the reduced performance of the traffic over WLAN because other factors are involved as well, such as the signal strengths of both the WLAN and WISA. The result does however illustrate that co-existence might be a challenge.

2.2.3 Mulitrate support in IEEE 802.11g

The original 802.11 specification states that different PHY may have multirate support that allow dynamic rate switching with the objective of improving performance. The 802.11g has support for the rates 6, 9, 12, 18, 24, 36, 48 and

54Mbit/sec. The standard does not specify any further how these rates are to be dynamically chosen under varying conditions and this choice is left to the implementer.

2.2.4 Carrier Sense in IEEE 802.11g

The different 802.11 standards all use a mechanism for avoiding that different stations transmit at the same time. This is an important mechanism when the medium is shared and there is no central control that assigns each station a specific time to transmit, since simultaneous transmissions may cause a corruption of the transmitted data and thus require retransmission. The different stations will first sense the medium before transmitting and not transmit if a signal of fulfilling a criteria is detected. The 802.11b has different modes that are used for detecting a busy medium:

Mode 1 When the energy on the channel exceeds an energy detection threshold.

Mode 2 When a valid 802.11b signal is detected on the channel.

Mode 3 A combination of Mode 1 and Mode 2. Only a valid 802.11b signal above a certain energy threshold will cause a busy medium.

The 802.11g standard uses only Mode 3 for carrier sensing, as specified in [5] section 19.3.5. In other words, only a valid 802.11g signal above a certain energy threshold will cause an 802.11g to report a busy medium. This should imply that a signal from a WISA BS will not cause an 802.11g station to detect that the medium is busy.

2.3 Transport Layer Protocols

The software developed to test the performance of WLAN in co-existence with WISA will use two Transport Layer Protocols for sending data from one computer to another. The two protocols used are Transport Control Protocol (TCP) and User Datagram Protocol (UDP), both which will be briefly introduced in the following sections. Finally, the expected bandwidth using these two protocols on a 802.11g network will also be discussed.

2.3.1 Transport Control Protocol

The Transport Control Protocol (TCP) is designed to provide a reliable end-to-end byte stream over an unreliable internetwork. An internetwork may have

widely different topologies, bandwidths, delays, packet sizes and other parameters. TCP dynamically adapts to the properties of the network and is robust in case of network failures. The protocol is used for many Internet applications that require reliability and sequential delivery. Some of the higher layer protocols that rely on TCP are File Transfer Protocol (FTP), Telnet, Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP)[7].

2.3.2 User Datagram Protocol

The User Datagram Protocol (UDP) is a connectionless protocol that may deliver data out of order, duplicated or not deliver the data at all. It does provide an interface for applications to the Internet Protocol (IP) and a means for application-to-application communication. UDP has less overhead than TCP, since it does not need to check if every packet actually is delivered. This property has made UDP attractive for real-time multimedia applications as Internet radio, Internet telephony, videoconferencing and video-on-demand[7].

2.3.3 Bandwidth

The bandwidth is in this thesis defined as the number of bits with which a user can send and receive data between two computers connected over a 802.11g WLAN. A term used for this is *throughput*, but to be consistent with the software discussed later in this report, *bandwidth* will be used instead.

The 802.11g standard supports data rates of up to 54Mbits/sec, the experienced bandwidth is less than this for several reasons:

- Each packet includes overhead, such as preambles, headers and checksums.
- When every packet is received, the receiver transmits a short acknowledgement packet back to the sender.
- Transmitters wait for short random times between packets to allow other users to contend for and share the channel.

For these reasons, the theoretical maximum transport layer bandwidth for a 802.11g network is 24.4Mbits/sec for TCP and 30.5Mbits/sec for UDP[6].

NB: The term *bandwidth* is also commonly used as a description of the width of a frequency band, but this report use the term as description of transferred bits.

Chapter 3

Hardware

This chapter gives a description of the hardware components used and how they are connected to perform tests of a WLAN. The description of the hardware is brief, a more thorough listing of the specifications and parameters is given in appendix A.

3.1 The Test Setup

The test setup includes components that can be arranged in various ways to test the performance of a WLAN. Three different setups were tried and each of them will be presented in the following sections. Each arrangement has it advantages and disadvantages which will be pointed out. Which setup to use will depend on the need for measurement of WLAN data rate and the use of coax cables as the wireless medium.

The generation of WISA signals that cause interference on a WLAN could be by any equipment capable of generating WISA BS signals. Signal generators capable of generating the WISA FH scheme was hard to find and therefore an actual WISA BS, with modified software, was used in the hardware setups. Interference caused by SA's is believed to have little effect on a WLAN since they transmit for only a fraction of the time and is therefore not considered in this hardware setup.

3.1.1 Alternative 1

This test setup includes the following components:

- Wireless Local Area Network (WLAN) Access Point (AP)

- Wireless Interface for Sensors and Actuators (WISA) Base Station (BS)
- Ethernet-to-WLAN bridge
- 2 laptop computers
- Spectrum Analyzer (Anritsu MS2721A)
- Coax cables
- 2 T-Adapters
- 1 serial cable with a USB-to-serial adapter
- Attenuators: 20dB, 10dB, 9dB.

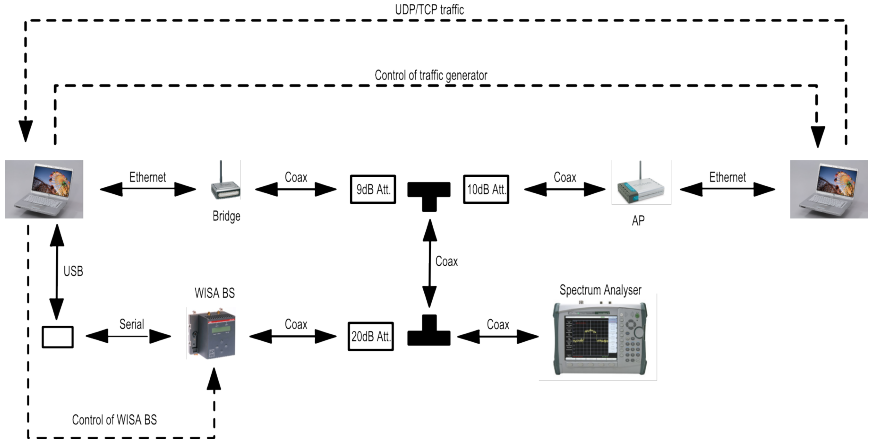


Figure 3.1: The test setup Alternative 1.

Description

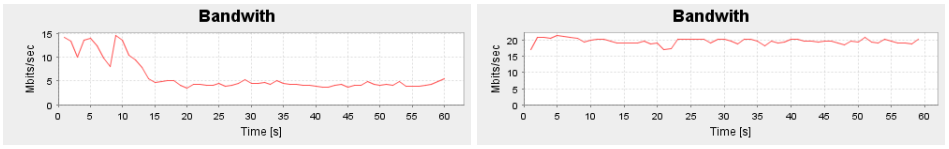
The Ethernet-to-WLAN bridge is connected to the first computer with a standard Ethernet cable. The external antenna on the bridge is replaced by a 3 meter long coaxial cable. The other end of the cable is connected to a 9dB attenuator, and then to the first T-adapter. The second computer is connected to the WLAN AP with a Ethernet cable. Another coaxial cable connects the AP to a 10dB attenuator and then to the first T-adapter. This T-adapter is then connected to the second T-adapter with a coax cable. The WISA BS is also connected to the second T-adapter with a coax cable and a 20dB attenuator. The last port of the second T-adapter is used to connect a spectrum analyzer. Finally the first computer is connected to the WISA BS using a USB-to-Serial adapter. An illustration of the setup can be seen in figure 3.1, where the dashed lines are used to illustrate the function of software on the computers.

In chapter 4 the developed software will be described and several screen shots will be presented, all these screen shots are from the computer connected to the WISA BS. The other computer (the one not connected to the WISA BS) will only be used to generate the UDP/TCP traffic and is controlled through the program running on the first computer. This is also the case for the remaining setups.

Comments

The WLAN in this setup consists only of the AP and the Ethernet-to-WLAN bridge, therefore only the connection between these two components is wireless. The term wireless might be a little confusing in this case since they are connected with coax cables, but it is used to describe any connection communicating with an IEEE 802.11 standard.

This setup was first tried without any of the attenuators. The performance of the



(a) Bandwidth without attenuators (b) Bandwidth with attenuators

Figure 3.2: TCP bandwidth measurements of test setup Alternative 1.

WLAN was then very fluctuating, even when the WISA BS was turned off. The attenuators were then added as shown in figure 3.1, and the difference was easily observable. The bandwidth as function of time over a TCP connection between the computers can be seen in figure 3.2 for the setup with and without the attenuators, and the WISA BS turned off.

A possible explanation of this phenomenon might be that the received signals at both the WLAN AP and the Ethernet-to-WLAN bridge were stronger than the equipment is designed for. This matter was however not further investigated.

The weakness of this setup is that the computer receiving the UDP/TCP traffic is behind the Ethernet-to-WLAN bridge and not the WLAN AP. This is a problem if this computer should be able to request the AP of its current data rate.

Such a request would cause undesired traffic on the wireless link and might influence the UDP/TCP traffic over this link. An attempt to avoid this problem was done with a different setup as explained in section 3.1.2.

Results produced with this setup will be presented in section 5.1.

3.1.2 Alternative 2

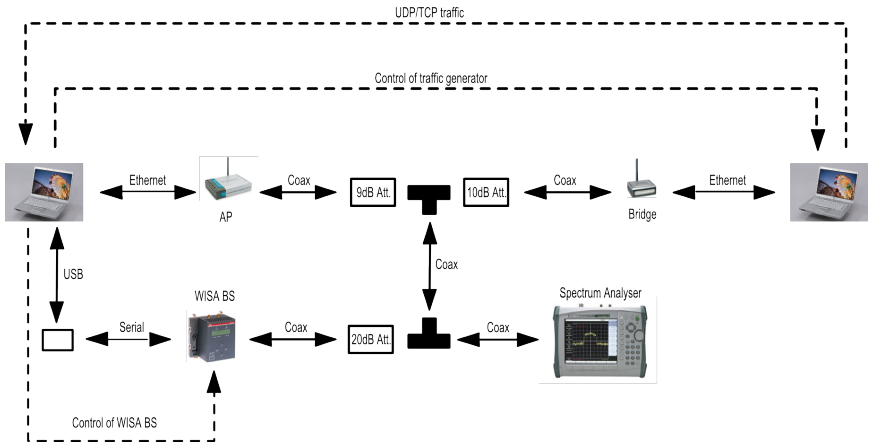


Figure 3.3: The test setup Alternative 2

The test setup includes the same components as described in section 3.1.1:

Description

This setup is very similar to the setup described in section 3.1.1, the only difference is the WLAN AP which has changed place with the Ethernet-to-WLAN bridge.

Comments

As for Alternative 1, the WLAN consists only of the Ethernet-to-WLAN bridge an the WLAN AP. Alternative 2 however allows the computer receiving the UDP/TCP traffic to request the AP of its data rate without using the wireless link, but it also has its drawbacks. The Ethernet-to-WLAN bridge might fail under heavy traffic load and cause the link between the bridge and the AP to loose its connection completely. This was a problem irrespective of the WISA BS being turned on or off, however the bridge would sustain heavier traffic when the BS was turned off. A consequence of this is that Alternative 2 requires careful restriction of traffic load to avoid the failure of the bridge. In Alternative 1 on the

other hand, a heavy traffic load would simply cause an increase in lost UDP datagrams and not a complete breakdown of the wireless link.

Results produced with this setup will be presented in section 5.2.

3.1.3 Alternative 3

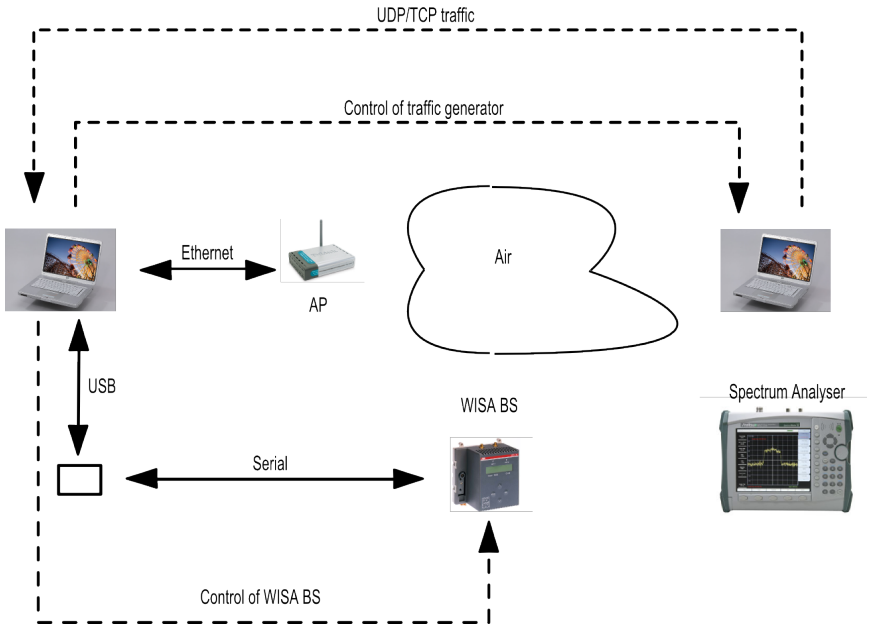


Figure 3.4: The test setup Alternative 3

The test setup includes the following components:

- Wireless Local Area Network (WLAN) Access Point (AP)
- Wireless Interface for Sensors and Actuators (WISA) Base Station (BS)
- 2 laptop computers
- Spectrum Analyzer (Anritsu MS2721A)
- 1 serial cable with an USB-to-serial adapter

Description

This setup has no coax cables, but instead uses air as the wireless medium. The Ethernet-to-WLAN bridge was therefore unnecessary as the motivation for using it was that it had an external detachable antenna that could be replaced by a coax cable. The WLAN in this setup therefore consists of the AP and the

internal WLAN adapter of the second computer.

The first computer is connected to the WLAN AP using an Ethernet cable, the second is connected using its internal WLAN adapter. The WISA BS is connected to the first computer using a USB-to-Serial adapter and a serial cable. Both the AP and the BS were fitted with antennas instead of connecting them to a coax cable. An illustration of the setup is found in figure 3.4.

Comments

The problems associated with the setups described in sections 3.1.1 and 3.1.2 are not present with this arrangement of components. The use of air as the wireless medium however does make the wireless link more exposed to the problems normally associated with a wireless link such as noise, multipath fading and other degrading factors that can be time variant and thus makes this setup less suited for measurements and comparison.

3.2 The components

The most significant hardware components used in the different setups will now be presented and briefly described. A more detailed description of their specifications and parameters can be found in Appendix A.

3.2.1 WISA Base Station

The WISA BS is the device that communicates to all the SA's in a WISA cell. The WISA communication system is designed so that the BS will transmit continuously whereas the SA's will, when compared to the BS, transmit only for a fraction of the time. The BS will transmit on 1 of 77 channels, each channel 1MHz wide, in the ISM frequency band. The channel is changed every 2048μs and the sequence of channels is deterministic.

The BS used in this project has a non-standard software that enables complete control of the channels that are used as well as the order they are used in. The standard software would only allow limited control of the sequence of the channels by changing the cell ID, this would not be sufficient for the goal of this project. The non-standard BS software also enables muting of selected channels. A BS normally transmits at 0dBm, but when a channel is muted the signal strength is reduced by approximately 35dB.

Figure 3.5: The WISA Base Station

The BS normally use antenna switching and alternates the antenna used for transmission for each time the channel is changed, this feature is disabled.

3.2.2 WLAN Access Point

In a WLAN the AP is a device that connects the WLAN communication devices together to form a wireless network. For this task a DWL-2100AP, produced by D-Link is used. This AP has been verified and successfully passed through the Intel Wireless Verification Program and the Wi-Fi Alliance, this gives assurance that a product has met rigorous interoperability testing requirements to ensure that compatible products from different vendors will work together.

The AP supports three IEEE 802.11 standards, these are 802.11, 802.11b and

Figure 3.6: The WLAN Access Point: D-Link DWL-2100AP

802.11g. This allows data rates of up to 54MBits/sec. The AP also has support for a proprietary standard which makes it possible to achieve data rates up to 108MBits/sec, this standard is not used in this project.

The AP is configurable through a web browser or Telnet, the latter will be used to read the instantaneous data rate of the AP. The AP has an external detachable RP-SMA antenna which is in some of the setups replaced by a coax cable.

3.2.3 Ethernet-to-WLAN bridge

The computers used in the setup have internal WLAN adapters that enable them to connect to a WLAN. This adapter does not have an external detachable antenna, which is required if coax cables are to be used as the wireless medium. To solve this problem, an Ethernet-to-WLAN bridge with an external antenna was used. The Ethernet-to-WLAN bridge is designed for making an Ethernet-only device able to connect to a WLAN. By connecting the bridge to the computers Ethernet adapter, it was possible to use the coax cables as the wireless medium.

The bridge by D-Link, with the product name Airplus XtremeG Ethernet-to-

Figure 3.7: The Ethernet-to-WLAN bridge: D-Link DWL-G810

Wireless Bridge DWL-G520 is chosen for this task. This bridge is also verified by the Wi-Fi Alliance and should therefore be guaranteed to work well with the WLAN AP. As the AP, the bridge supports the 802.11g and 802.11b standard. The external antenna contact is a female RP-SMA which is used to connect with the coax cables.

3.2.4 Computer

The WLAN performance will be tested by generating TCP or UDP traffic over the wireless link and observe how it is affected by a WISA BS. The traffic will be generated by one computer, received and analyzed by another using appropriate software.

The two computers used are standard off-the-shelf computers produced by

Figure 3.8: The computer: HP G5000

Hewlett-Packard with the model name HP G5000. They have an internal WLAN adapter as well as an Ethernet adapter, but for most of the setups previously described, an Ethernet-to-WLAN bridge will be used to connect one of them to the WLAN.

Chapter 4

Software Tool

The development of a software tool for analyzing the effect a WISA BS has on WLAN traffic was the objective of this diploma project. The software developed enables the user to start a TCP or UDP flow from one computer to another, and to control a WISA BS. The tool gives the user feedback of bandwidth, jitter and packet loss of the traffic flow between the two computers and is called ABB-Traffic. The software was developed in Java, using Netbeans as development tool.

In this chapter a thorough description of the usage of ABBTraffic will be given, as well as a short description of the software that ABBTraffic relies on. The Java classes that ABBTraffic consists of and how they collaborate will not be explained in this chapter; this information can be found in appendix B.

4.1 ABBTraffic

ABBTraffic is a distributed application, in other words it runs on more than one computer. One computer generates traffic that will flow through a network, while the other receives it. A simple illustration can be seen in figure 4.1. The two computers have been numbered to simplify the following discussion.

The software running on computer 2 has no Graphical User Interface (GUI) and allows no user input, it is completely controlled by the software running on computer 1. The software running on computer 1 is the part of ABBTraffic that provides the user a GUI to configure and start the traffic flow from computer 2. This GUI will also display the measurements of bandwidth, jitter, packet loss and data rate. As seen in the illustration, computer 1 is also connected to the WISA BS which the user can control through the GUI.

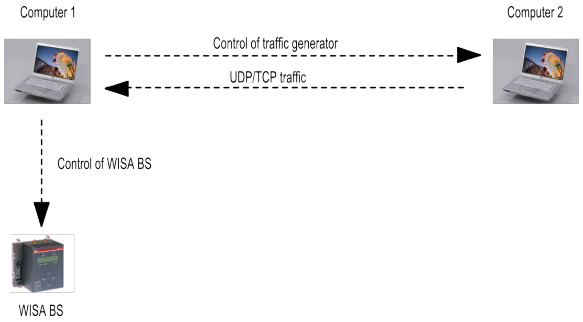


Figure 4.1: An illustration of the distributed application.

ABBTraffic uses two software tools for generating the data that flows between the computers and measuring the bandwidth, jitter, datagram loss and data rate. These are called Iperf and Expect and will be presented in section 4.2.

The following sections will be focusing on the part of ABBTraffic running on computer 1, and the GUI used to control it. The explanation of the GUI and the function of it will be presented in the following four sections:

4.1.1 Main application window, this is the window the user is first presented when launching ABBTraffic.

4.1.2 The GUI for configuring the traffic that flows from one computer to the other.

4.1.3 The GUI for controlling the WISA BS.

4.1.4 A window displaying bandwidth, delay jitter, datagram loss and data rate of a UDP/TCP flow.

To explain the GUI a combination of screen shots of ABBTraffic and text is used. The next page is left blank so the reader can read the text and view the associated screen shot at the same double-page.

4.1.1 The Main Window

The main window of the program is the window that the user is presented to when launching ABBTraffic. The window is divided into two areas, one area contains graphs that display the performance of the network connecting the two computers, while the other is used for configuration of the WISA BS and the traffic generator. The graphs will not show any information unless traffic is being generated, when traffic is generated they will be continuously updated and show up to 15 seconds of data. For generation of TCP traffic, only the bandwidth graph will show any data[1].

A screen shot of the main window can bee seen in figure 4.2, the most important features are encircled and numbered:

1. The *File*-menu is used for loading of previous results of traffic generation; how these results are generated and saved will be explained later. The *COM port*-menu is used for selecting the COM port to which the WISA BS is connected.
2. This shows the current bandwidth that a TCP or UDP flow is able to deliver from one computer to the other.
3. The graph displays the current data rate that the wireless device connected to the WLAN AP uses to communicate.
4. This graph shows the current delay jitter in the network, it will only show any information when a UDP flow is generated by the traffic generator.
5. This displays the current datagram loss through the network, as for the delay jitter graph, this graph will only show information during a UDP flow.
6. The WLAN traffic and the WISA BS is controlled by selecting one of the two tabs in this area. Each tab will be explained separately in sections 4.1.2 and 4.1.3 respectively.

[1] This is a limitation of Iperf, the software that ABBTraffic uses. This software will be presented in section 4.2.1

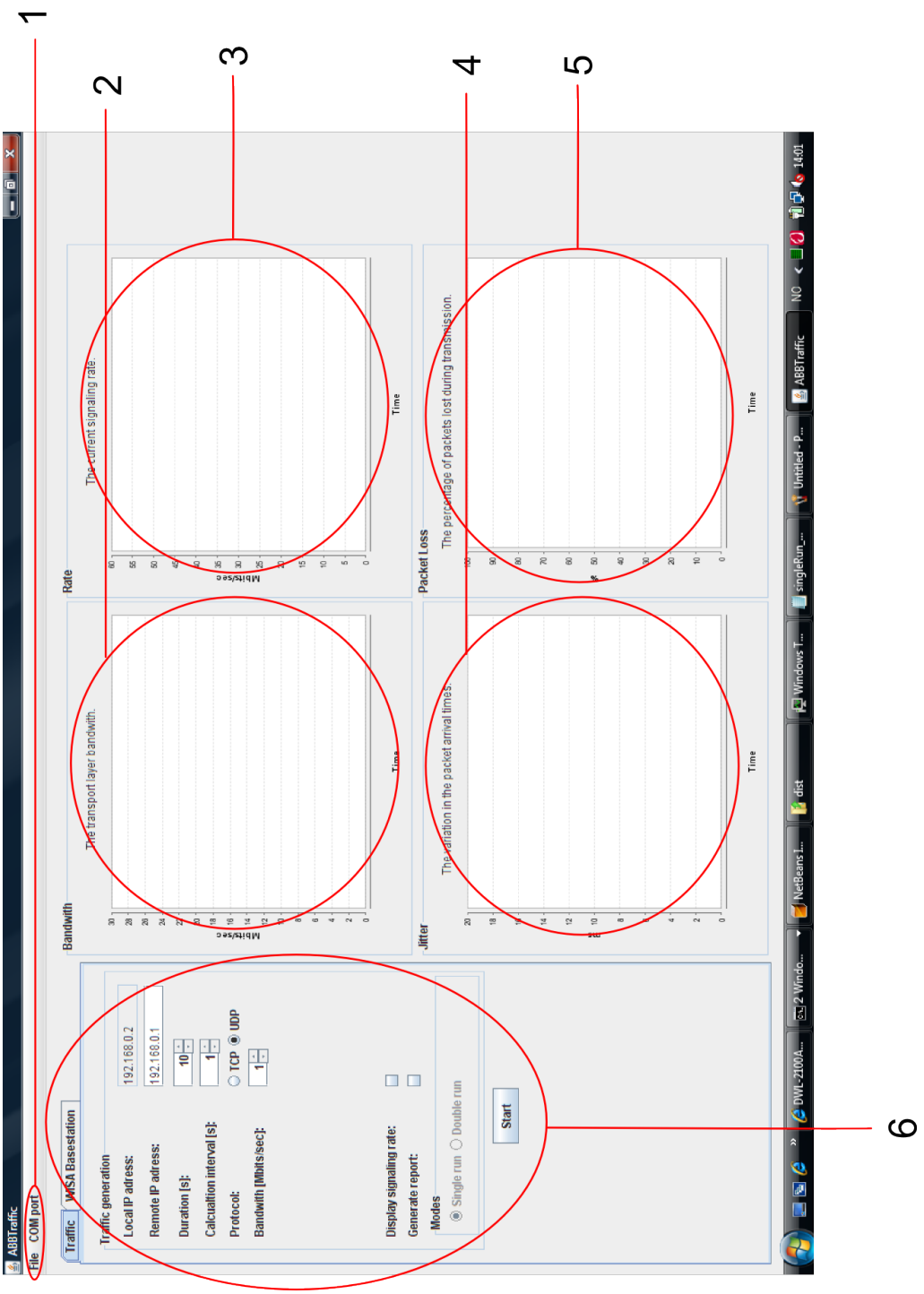


Figure 4.2: A screen shot of the main window (rotated 90 degrees).

4.1.2 Traffic generation

ABBTraffic generates an artificial stream of data that is transported from one computer to another over a network, using either UDP or TCP. The bandwidth, delay jitter and datagram loss of the stream is calculated at regular intervals and displayed in the graphs shown in figure 4.2. A stream is configured before it is started, once it is started it can not be changed.

The GUI for this configuration is as shown in figure 4.3 and is a close-up of circle 6 in figure 4.2. The important features and functions of this GUI are encircled and numbered:

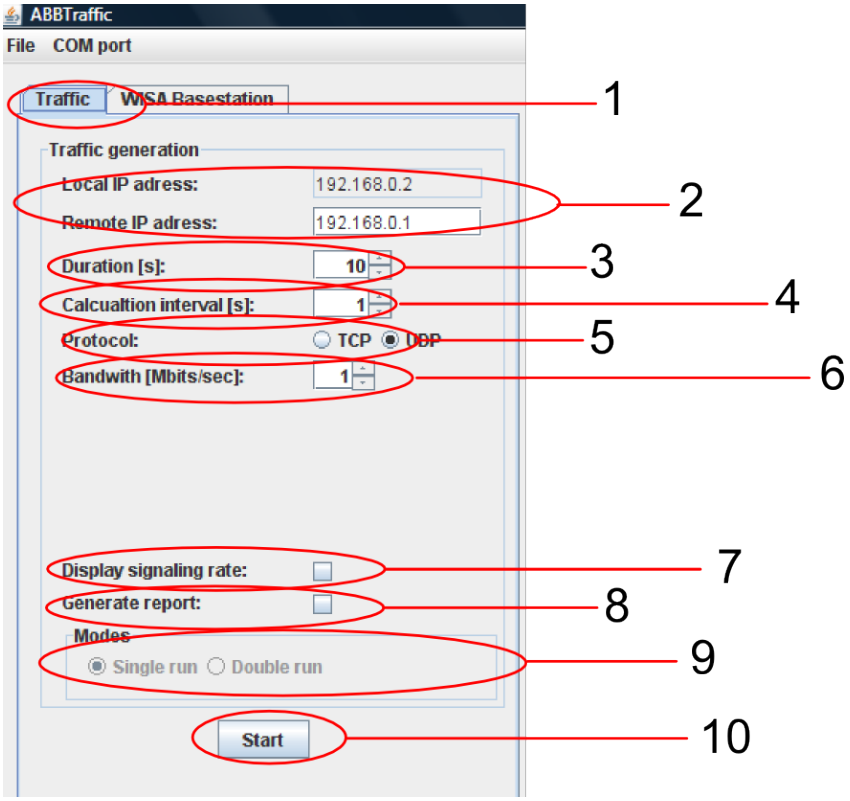


Figure 4.3: A screen shot of the GUI for controlling traffic generation of ABBTraffic.

1. The tab that brings the traffic generation controls to the front.
2. The IP addresses of the computer running this tool and of the remote computer that will generate the stream of data. The IP address of the remote computer must be specified by the user, otherwise the stream will not be started.

3. Specifies for how long traffic should be generated.
4. This changes the interval at which bandwidth, delay jitter and datagram loss are computed.
5. Selects the protocol used to transport the data between the two computers.
6. If UDP is selected as protocol, this spinner selects the attempted bandwidth of the data stream between the computers. In section 3.1.2 it was mentioned that a heavy traffic load might cause the Ethernet-to-WLAN bridge to fail using the particular setup of figure 3.3 and cause the connection between the two computers to be lost. To avoid this, the user should not use a bandwidth higher than 20Mbits/sec when the WISA BS is turned off, and not more than 12Mbits/sec when turned on and the BS using the normal FH scheme. Note that this restriction exists for the setup described in figure 3.3 only.
7. ABBTraffic will request the WLAN AP of its current data rate to the device connected to it if this box is checked. The connected device should either be the Ethernet-to-WLAN bridge of figure 3.3, or the internal WLAN adapter of the second computer of figure 3.4. This option should not be enabled when the setup is as in figure 3.1.
8. The graphs shown in figure 4.2 will only show data that is less than 15 seconds old. To see graphs for the entire duration the traffic was generated, this box should be checked. After traffic generation, ABBTraffic will then display a separate report window with graphs of bandwidth, delay jitter, datagram loss, data rate and other useful information.
9. When a report is requested it also possible to chose a simulation mode, *Single run* or *Double run*. If *Single run* is selected, traffic generation is performed with the WISA BS as currently configured. In *Double run* the traffic generator is started twice, first with the WISA BS as configured by the user, then with the WISA BS turned off. The latter mode is useful for comparison of WLAN performance with and without the WISA BS.
10. This starts the traffic generation as configured by the user, once it is started it will display the measured bandwidth, jitter, datagram loss and data rate in the graphs of figure 4.2.

4.1.3 Configuration of WISA Base Station

ABBTraffic provides the user the ability to configure a connected WISA BS for the purpose of observing how different FH schemes cause interference on a

WLAN. The GUI for configuring a connected BS is found under the tab called *WISA Base Station* in the area numbered as 6, in figure 4.2.

The BS operates in one of two modes: *Normal* or *Table*. The *Table*-mode is used for customization of the WISA FH sequence. The *Normal*-mode is used when the BS should use its standard WISA FH sequence based on the cell ID. To set the cell ID of the BS one must use the interface on the BS itself.

A screen shot of this tab can be seen in figure 4.4, the important buttons and features have been encircled and numbered:

1. The tab that brings the WISA BS controls to the front.
2. Selects the mode of the BS, *Normal* or *Table*.
3. For *Table*-mode, the FH sequence is specified in a text file. The name of the text file is entered into the text field and uploaded by clicking on the *Load* button. The file should be located in the same directory as the ABBTraffic executable. The file that specifies the FH sequence should be a simple text file containing only the numbers of the WISA channels to be used, in their correct order. A sample text file which contains the FH sequence of a WISA BS with cell ID 19 is found in appendix B.
4. This area contains 77 radio buttons, each representing a WISA channel. When a text file is loaded, the channels used by the FH sequence will show as enabled radio buttons. The enabled radio buttons can now be used to mute the desired channels of the FH sequence. A channel is muted by clicking on the radio button with the associated channel number. A FH sequence may use a channel multiple times, but it is only possible to mute it every time or not at all. The radio buttons 3 to 25, with the gray numbers are not enabled. The radio buttons from 26 to 29 are enabled and the channel is muted. The remaining buttons, from 30 to 79, are enabled but not muted.
5. The *Mute all*-button selects all the enabled radio buttons representing the channels used by the FH pattern. The *Mute none*-button de-selects any selected radio buttons. The *Mute a WLAN channel*-button brings up a dialog which asks the user for a WLAN channel, the WISA channels surrounding the centre of this WLAN channel is then muted. 17 WISA channels are muted, the user may manually mute additional or less channels. See section 2.2.2 for the motivation for muting 17 channels.
6. The *Load to basestation*-button starts a writing process to the COM port that will configure the WISA BS as the user has configured in the GUI. If

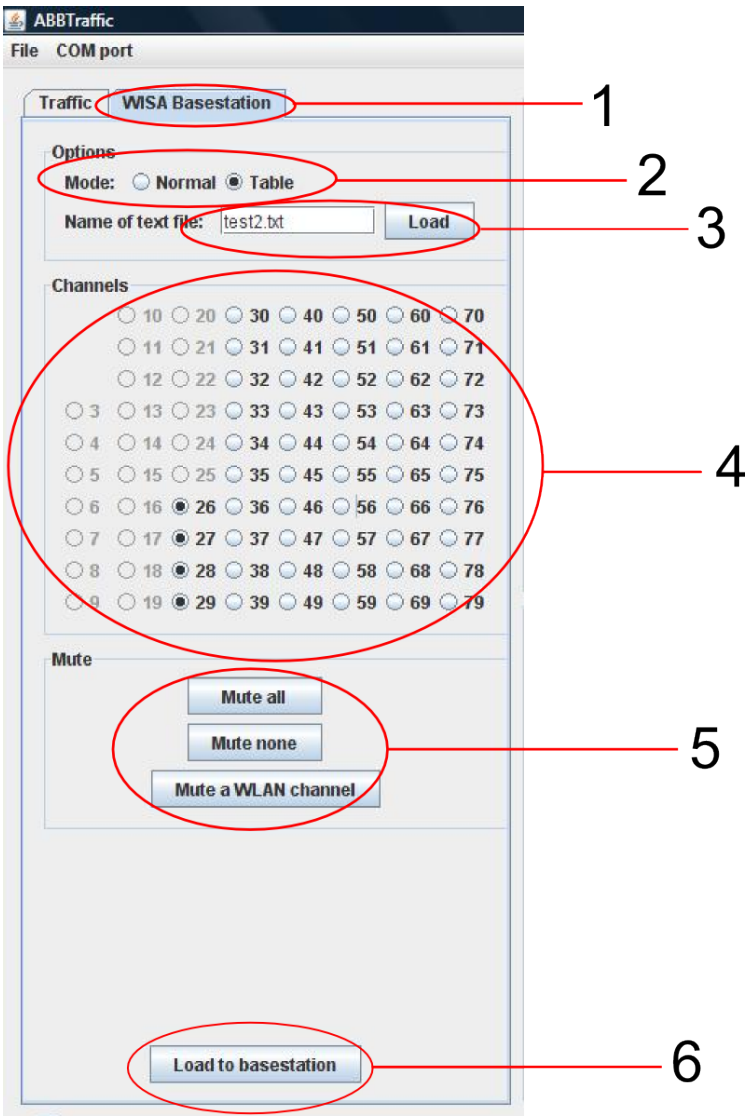


Figure 4.4: Screen shot of the GUI for configuration of the WISA Base Station in ABBTraffic.

the user has not selected a COM port the process will be stopped and a dialog will appear asking the user to select a COM port.

4.1.4 The Report Window

The Report window will be displayed if the box encircled as number 8 in figure 4.3 is checked. This window will be displayed as a separate and new window when a traffic generation has ended. It will contain graphs of bandwidth, delay jitter, datagram loss and data rate for the entire duration of traffic generation. A screen shot of such a window is found in figure 4.5, the most important features are encircled and numbered:

1. This graph shows the bandwidth for the network with the WISA BS on (red) and off (blue), for the entire duration of the traffic generation.
2. An area displaying some information about the time the generation was started, when it stopped, the protocol used, bandwidth, calculation interval, local IP address, remote IP address, WISA BS mode and the name of the file that specifies the FH sequence.
3. This graph shows the delay jitter of the network. A single graph can be saved by right clicking on the graph and selecting the *Save as* option from the menu appearing.
4. This graph shows the datagram loss of the network.
5. This graph shows the data rate between the WLAN AP and the device connected to it.
6. This button is used if it is needed to save this report to a text file. The text file can be viewed with any program capable of opening a *txt*-file. The text file can also be used to generate this summary window again, see figure 4.2 circle 1. A sample of a report file is found in appendix B.
7. This area contains tables of average, best and worst measurements of the bandwidth, jitter, loss and rate.

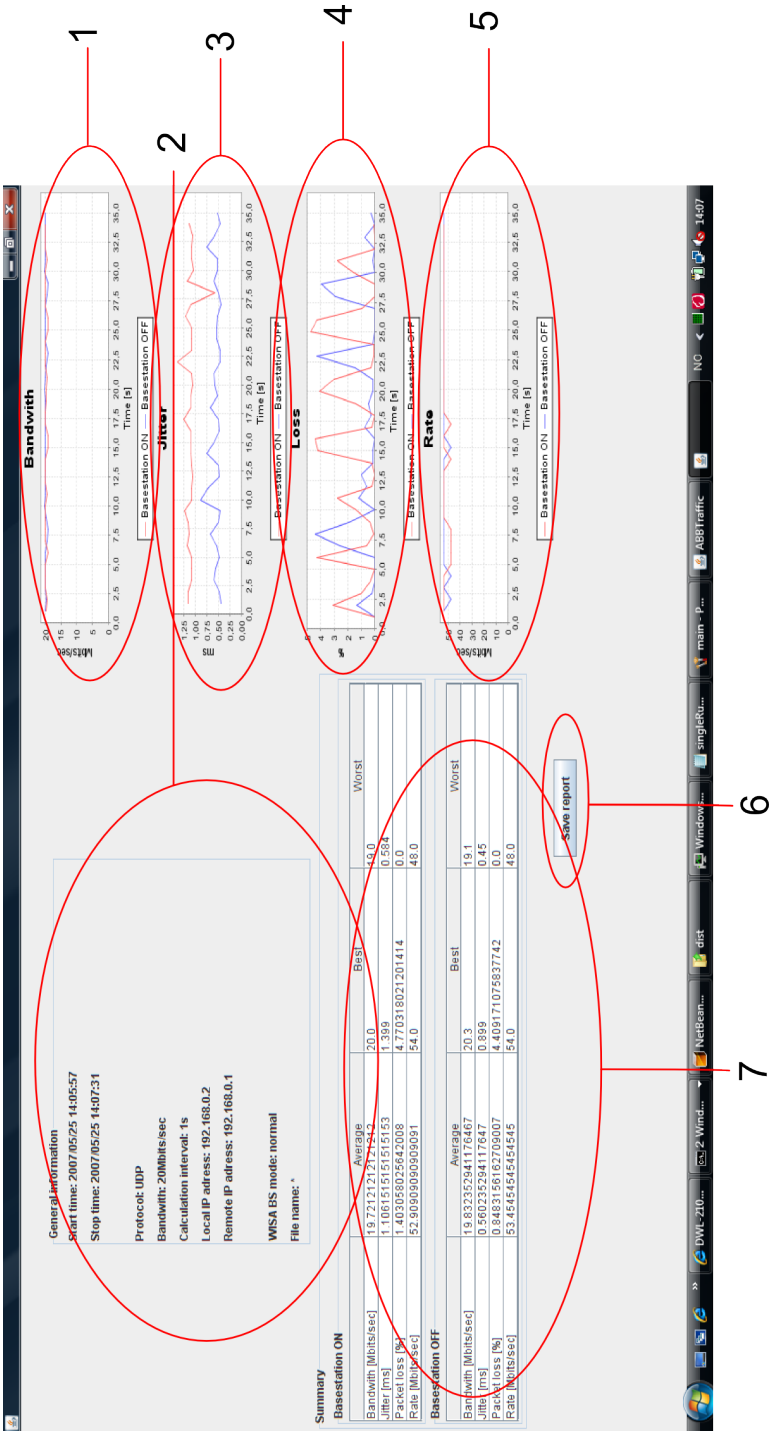


Figure 4.5: A screen shot of the Report window (rotated 90 degrees).

4.2 Libraries and Software

ABBTraffic relies on several software tools and Java libraries to complete its tasks; these will be shortly introduced in this section.

4.2.1 Iperf

ABBTraffic is dependent on a software tool called Iperf[2] for the generation and reception of UDP and TCP traffic. Iperf is developed by The National Laboratory for Applied Network Research and The Distributed Applications Support Team. The software has been developed for measuring TCP and UDP bandwidth performance. It reports measurements of bandwidth, delay jitter and datagram loss.

Iperf has two parts, a client and a server. The client will connect to a server and send data. The reported bandwidth is a measurement of the transported bits per second between the client and the server. Jitter calculations are continuously computed by the server. The client records a 64 bit second/microsecond timestamp in the packet. The server computes the relative transit time as (server's receive time - client's send time). The client's and server's clocks do not need to be synchronized, any difference is cancelled out in the jitter calculation. Jitter is the smoothed mean of differences between consecutive transit times. The jitter is an important metric for applications such as audio and video streaming, since it is more important for such applications that the inter transit times are constant than the transit time being low[7]. The reported datagram loss is the percentage of datagrams sent by the client, but not received by the server within the calculation interval.

Iperf was developed for the purpose of tuning parameters such as TCP window size and data payload to find the optimum settings for a given network between two computers. The motivation for using Iperf in ABBTraffic is to simply observe the difference in reported bandwidth, jitter and datagram loss when interfered by a WISA BS. Therefore the various parameters are set as default.

The Iperf server and client are started by ABBTraffic when the *Start* button, encircled and numbered as 10 in figure 4.3, is clicked by the user. The Iperf client is started on the remote computer, with the IP address as specified by the user. ABBTraffic will also start the Iperf server on the local computer and display its output in the appropriate graphs. In figure 4.1 *Computer 1* would be

[2]The project homepage: http://dast.nlanr.net/projects/Iperf/

the local computer, while *Computer 2* would be the remote computer. The Iperf server will not give any output unless the Iperf client is successfully started, and the Iperf client will not be started if the IP address of the remote computer is wrong.

Note: The failing of the Ethernet-to-WLAN bridge as described in section 3.1.2, may cause the Iperf server running on the local computer to wait indefinitely. ABBTraffic will wait for the Iperf server to finish before proceeding, so also ABBTraffic may wait indefinitely. To recover from such a situation the user must manually close the Iperf server process in the *Windows Task Manager*. Using either the setups described by figure 3.1 or 3.3 will avoid this situation.

4.2.2 Expect

ABBTraffic uses a software tool called Expect[3] to get the data rate of the WLAN AP, this is a tool for automating interactive applications like telnet[4]. Expect's procedure for interaction with the AP is defined by a script file, the file used by ABBTraffic is named *expectScript.exp* and is placed in the same folder as the ABBTraffic executable. This script file contains the IP address, as well as the administrator username and password of the WLAN AP. This file can be viewed in Wordpad and is fairly easy to understand so it will not be further elaborated here. It is important to remember though, that any change of IP address, username or password of the AP must also be changed in the *expectSript.exp* file. The *expectScript.exp*-file can be viewed in appendix B.

Through the ABBTraffic GUI, the user must check the box encircled and numbered as 7 in figure 4.3 to observe the data rate. This is because ABBTraffic should only be allowed to request the AP of its data rate if the setup is as described in figure 3.3 or 3.4. Observing the data rate when using the setup as describe by figure 3.1 is possible and will not cause ABBTraffic to fail, but Expect will then use the wireless link and this may affect the results calculated by Iperf.

4.2.3 RXTX

The WISA BS is connected to the computer running ABBTraffic with a Serial-to-USB adapter, this adapter will appear as a COM port on the connected computer. Communication with a COM port in Java is possible using a library, for

[3] The project homepage: http://expect.nist.gov/

[4] telnet is an application supported by D-Link DWL2100AP for configuration and administrative tasks

ABBTraffic the library named RXTX[5] is used. The usage of RXTX is well documented on the projects homepage and will not be any further explained in this report.

4.2.4 jFreeChart

ABBTraffic uses several graphs to display measurements of bandwidth, jitter, datagram loss, and data rate. To create these graphs a Java library called JFreeChart[6] is used.

[5]The project homepage: http://users.frii.com/jarvi/rxtx/index.html
[6]The project homepage: http://www.jfree.org/jfreechart/

Chapter 5

Sample Tool Results

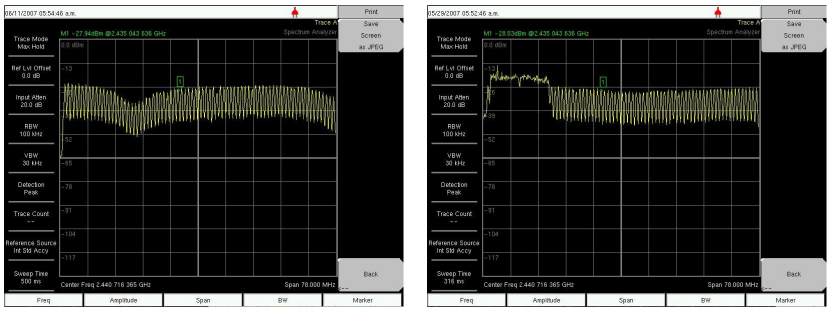
This chapter will present some results produced by ABBTraffic for setups described in 3.1. The purpose of this chapter is not to make any judgment or conclusion of WLAN performance in co-existence with WISA, but rather a demonstration of how ABBTraffic can be used to test the WLAN using the different WISA FH sequences. The results presented for each are graphs and average values of bandwidth, delay jitter, datagram loss and data rate. The different tests uses parameters as described in appendix A.

5.1 Reults of TCP

This section will present example of results of 60 seconds of TCP traffic between the two computers using the setup as described in section 3.1.2, with varying WISA FH sequences. The results include plots and averages of bandwidth and rate. Since Iperf does not calculate measurements of delay jitter and datagram loss when TCP is used, the results will not include such measurements either. ABBTraffic is used with the *Double run*-option enabled so the plots will show the performance under different FH sequences compared to the performance with the WISA BS turned off. Some screen shots of the spectrum analyzer are also included; these show the spectrum from 2.40Ghz to 2.48Ghz.

5.1.1 Normal Frequency Hopping sequence

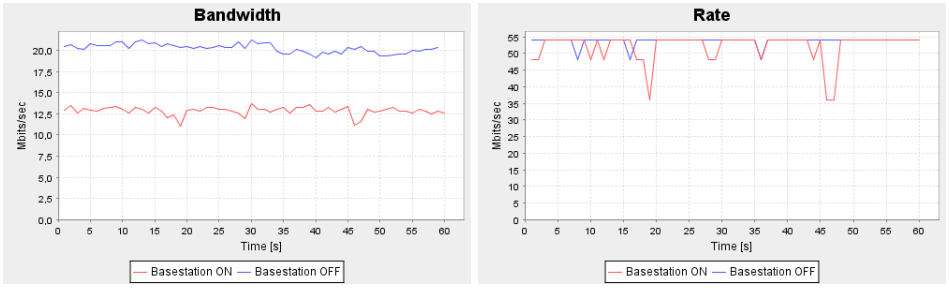
The WISA BS are with these results operating in *Normal*-mode and using cell ID 19. Screen shots of the spectrum analyzer is shown in figure 5.1: 5.1(a) shows the spectrum of the WISA BS only, 5.1(b) shows the spectrum of both WLAN channel 1 and WISA BS simultaneously.



(a) WISA spectrum with normal FH sequence.

(b) WLAN channel 1 and WISA spectrum with normal FH sequence.

Figure 5.1: Spectrum analyzer screen shots.



(a) Bandwidth

(b) Data rate

Figure 5.2: Results of TCP with normal WISA FH sequence.

WISA BS	**On**	**Off**
Bandwidth [Mbits/sec]	12.8	20.2
Data rate [Mbits/sec]	52.2	53.7

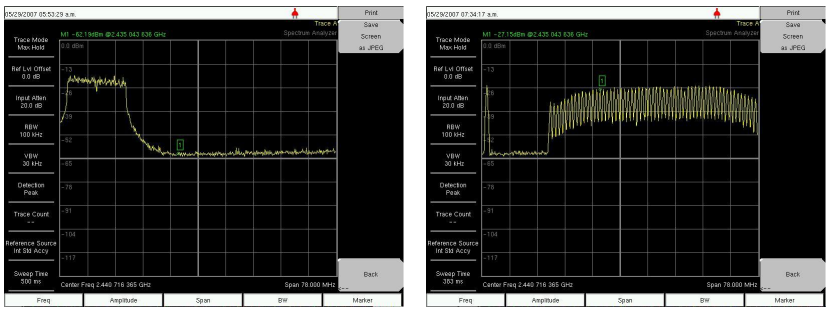
Table 5.1: Average results of TCP with the WISA Base Station using a normal WISA Frequency Hopping sequence.

The average bandwidth with the WISA BS using a normal FH sequence is approximately halved compared to the bandwidth when the BS is turned off. The average WLAN data rate is also reduced, but hardly enough to explain the reduction observed in the bandwidth. The data rate does however vary more when the BS is on and that might cause TCP congestion control mechanism to reduce the average bandwidth.

5.1.2 Nomal Frequency Hopping with muted channels

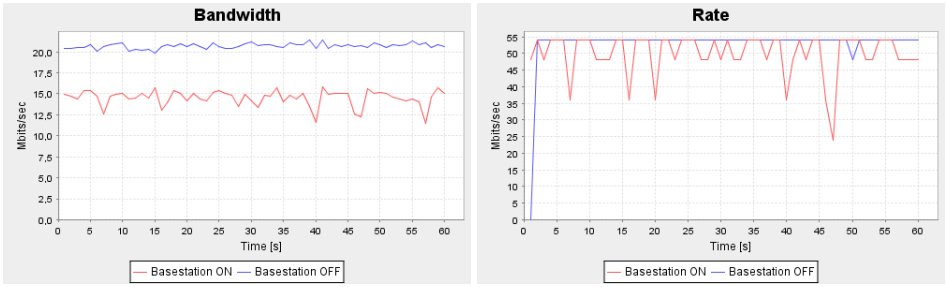
The WISA BS are with these results set to operate in *Table*-mode and thus use a FH sequence that is specified in a text file. The FH sequence used is identical to the sequence used by the BS if set to *Normal*-mode and with cell ID 19. The difference from the test in 5.1.1 is that the WISA channels within WLAN channel 1 are muted, i.e. channels 4 to 20. Two screen shots of the spectrum analyzer is shown in picture 5.3. Figure 5.3(a) shows the spectrum of the WLAN equipment only and the BS turned off. Figure 5.3(b) shows the spectrum of the BS with muted channels only, and the WLAN equipment turned off.

The average bandwidth with the muted channels shows a slight improvement



(a) Spectrum of WLAN channel 1 only. (b) WISA spectrum with muted channels. The muted channels range from 4 to 20.

Figure 5.3: Spectrum analyzer screen shots.



(a) Bandwidth (b) Data rate

Figure 5.4: Results of TCP and muted WISA channels. The muted channels range from 4 to 20.

WISA BS	On	Off
Bandwidth [Mbits/sec]	14.5	20.7
Data rate [Mbits/sec]	50.0	53.9

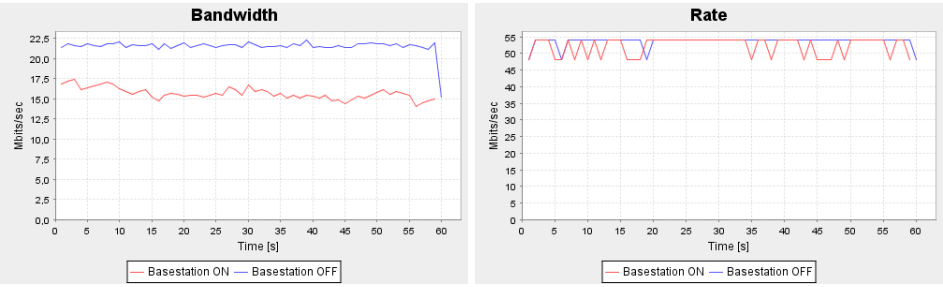
Table 5.2: Average results of TCP and with WISA channels. The muted channels range from 4 to 20.

compared to the results achieved in 5.1.1, but it is still considerably lower when compared to turning the BS off. The data rate is still very fluctuating and the average actually shows a minor reduction in performance.

5.1.3 Modified Frequency Hopping sequence

The WISA BS is set to operate in *Table*-mode as in sections 5.1.2 and 5.1.1. The FH sequence is however modified and does not use the channels from 4 to 20, the remaining channels of the FH sequence are used in the same order as previously. The screen shots of this spectrum looks exactly like the ones in figure 5.3.

The results show a slight improvement compared to the results in 5.1.2, but



(a) Bandwidth (b) Data rate

Figure 5.5: Results of TCP and modified FH sequence. The sequence is not using channels 4 to 20.

WISA BS	On	Off
Bandwidth [Mbits/sec]	15.6	21.5
Data rate [Mbits/sec]	52.2	53.7

Table 5.3: Average results of TCP and modified Frequency Hopping sequence. The sequence is not using channels 4 to 20.

still the difference in achieved bandwidth is clear when compared to the WISA BS being turned off.

5.1.4 Modified Frequency Hopping sequence ver. 2

The WISA BS is set to operate in *Table*-mode and with the FH sequence only using the channels from 26 to 79. The order of these channels are the same as the original FH sequence for a BS with cell ID 19. A screen shot of the spectrum analyzer in figure 5.6 show that the spectrum not used by the BS is wider than the WLAN channel use.

The results show that even now, with many WISA channels not being used,

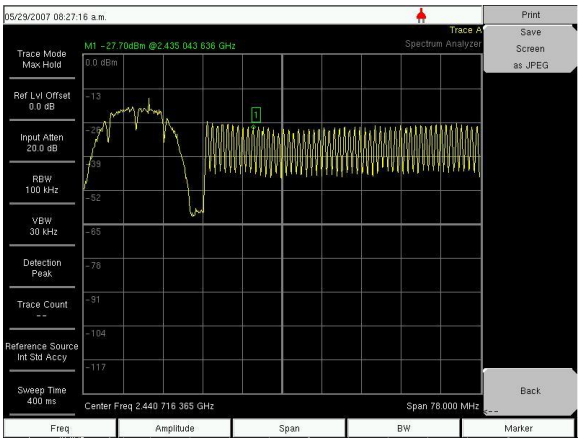
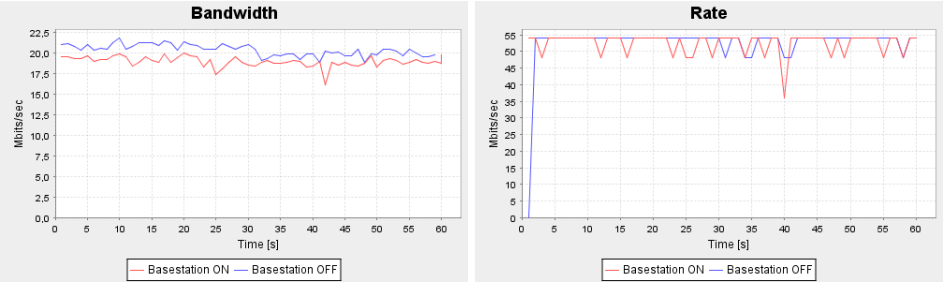


Figure 5.6: A screen shot of the spectrum analyzer with the WLAN equipment and the WISA BS on simultaneously. The FH sequence use channels ranging from 26 to 79 only.

WISA BS	**On**	**Off**
Bandwidth [Mbits/sec]	18.9	20.3
Data rate [Mbits/sec]	52.4	53.4

Table 5.4: Average results of TCP and modified FH sequence. The FH sequence use channels ranging from 26 to 79 only.

the bandwidth measurements are different when the BS is turned on compared to being turned off. The difference is however not as obvious anymore and may just as well be temporary fluctuations.



(a) Bandwidth (b) Data rate

Figure 5.7: Results of TCP and modified FH sequence. The FH sequence use channels ranging from 26 to 79 only.

5.2 Results of UDP

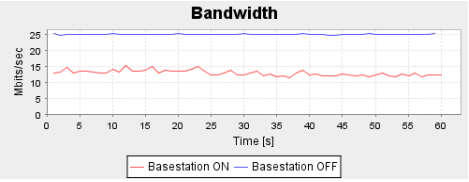
This section will present two results of 60 seconds of UDP traffic between the two computers using the setup as described in section 3.1.1. The result will not include measurements of data rate since this option should be disabled with this setup. The results are produced by ABBTraffic using the *Double run*-option, one with the WISA BS set to *Normal*-mode, the other using *Table*-mode. ABBTraffic is set to deliver 25Mbits/sec from one computer to the other in both cases.

5.2.1 Normal Frequency Hopping sequence

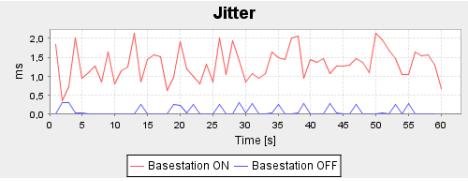
The WISA BS is set to *Normal*-mode and uses the cell ID 19. Similar to the results using TCP, the bandwidth is approximately halved when the BS is turned on. The datagram loss is close to 50%, but they are not necessarily lost on the wireless link. The network between the computers described in 3.1.1 consists of 3 links and only one of them is wireless and interfered by the BS. A datagram may be lost on any of these links. The delay jitter is close to zero when the BS is turned off, but is both variable and higher when the BS is on.

WISA BS	**On**	**Off**
Bandwidth [Mbits/sec]	13.0	25.0
Delay jitter [ms]	1.3	0.1
Datagram loss [%]	48.2	0.0

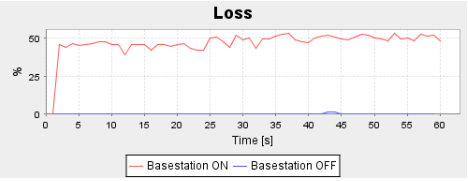
Table 5.5: Average results of UDP and normal FH sequence.



(a) Bandwidth



(b) Delay jitter



(c) Datagram loss

Figure 5.8: Results of UDP and normal FH sequence.

5.2.2 Normal Frequency Hopping sequence with muted channels

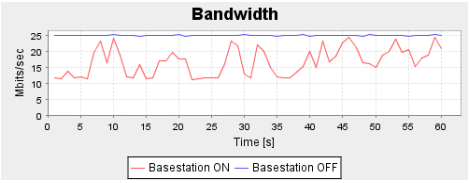
These results are produced with the WISA BS set to *Table*-mode, but the FH sequence is the same as used in the previous section. The WISA channels overlapping with WLAN channel 1 is muted, i.e. the channels 4 to 20.

The average bandwidth with the BS on is better when compared to the results

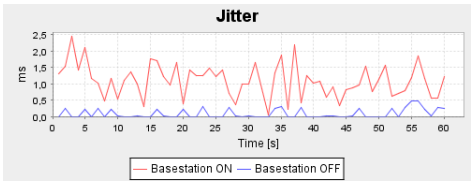
WISA BS	**On**	**Off**
Bandwidth [Mbits/sec]	16.8	25.0
Delay jitter [ms]	1.1	0.1
Datagram loss [%]	32.7	0.0

Table 5.6: Average results of UDP with muted WISA channels. The channels ranging from 4 to 20 are muted.

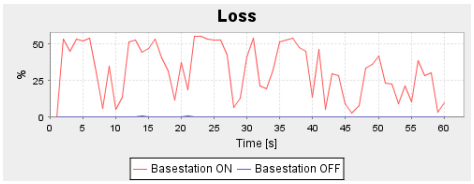
in 5.2.1, but as the graph shows, it is highly variable. At best it nearly achieves the same bandwidth as with the BS turned off, at worst it is just as low as it is in 5.2.1. The bandwidth is dependent on the datagram loss, therefore these graphs are inversely correlated, high datagram loss gives low bandwidth. The delay jitter shows similar characteristics to the results in section 5.2.1.



(a) Bandwidth



(b) Delay jitter



(c) Datagram loss

Figure 5.9: Results of UDP and muted WISA channels. The channels ranging from 4 to 20 are muted.

Chapter 6

Discussion

The developed software tool and assembled hardware setup presented in this project report does enable the testing of traffic over a WLAN interfered by a WISA BS. However, both the hardware and the software have issues to be considered, these will be discussed in this section.

The WLAN equipment used in this project was chosen for its support of the IEEE 802.11g standard, the standard with the highest data rate in the 2.4GHz ISM band. A motivation for using this particular 802.11g equipment was that it had external detachable antennas which enabled the use of coax cables as the wireless medium. All the tests and results in this report are produced with the WLAN equipment using the 802.11g standard, however if desired this equipment can be set to use the 802.11b standard instead. ABBTraffic should work perfectly fine with this WLAN setting, but would most likely not show as good performance as with 802.11g since the maximum data rate for 802.11b is only 11Mbps. It would also most likely require an even tighter restriction of the UDP bandwidth to prevent the Ethernet-to-WLAN bridge of failing, exactly how much is hard to say without practical tests.

The WLAN AP and the Ethernet-to-WLAN bridge could be replaced by equipment produced by any other manufacturer if desired. This might actually make ABBTraffic show different results for the traffic metrics under the interference of WISA since the mechanism for adapting the data rate is not specified in the IEEE 802.11g standard. ABBTraffic should work just as well with different WLAN equipment except for the logging of the WLAN data rate. The new WLAN AP would need to support Telnet and the script file used by Expect would probably need some alteration. ABBTraffic would also most likely need to be slightly modified to be able to read the output of the telnet session of the new

AP, but this would probably only be a minor modification.

The Ethernet-to-WLAN bridge did not perform well under heavy UDP traffic load when using the setup described in section 3.1.2. When this setup was used a restriction of the UDP bandwidth was required to prevent the bridge from failing and the connection to the AP to be lost. This problem did not occur when using TCP. The firmware for this bridge was upgraded as an attempt to "fix" the problem, but it made no difference. A newer version of this bridge, with the name DWL-G820 is now available for sale in the USA but not in Norway at present time. This new version might not have this same problem of failing under heavy traffic load. Another option for avoiding this problem is to use a second DWL-2100AP as a bridge; the DWL-2100AP used in this project has the capability of being used as a bridge. The DWL-2100AP did not fail under heavy UDP load when used as a AP (see section 3.1.1), but it was not tested how it performed when used as a bridge and therefore no guarantee can be given that it will not fail as the DWL-G810 did. To find a solution of the failing bridge would be useful since a hardware setup with the ability to log the WLAN data rate and use coax cables would then be available without the need for restriction of UDP bandwidth.

The test setups described in sections 3.1.1 and 3.1.2 both use attenuators that are connected to the T-adapters. The attenuators were placed at this particular place and not directly on the WLAN equipment since the attenuators could only be connected using SMA, the WLAN equipment use RP-SMA. Using RP-SMA attenuators or an adapter so they could be connected directly to the WLAN equipment would be preferable since this also would attenuate any noise picked up by the coax cables. Additional attenuators can also be used to cause various attenuation of either or both the WISA and the WLAN signals. E.g. testing how a weaker WISA signal would affect WLAN traffic could be realised by adding additional attenuators on the WISA BS if the setup in figure 3.1 or 3.3 is used. The same reduction in signal strength of the WISA BS could be realised when the setup described by figure 3.4 is used, but then also by physically placing the WISA BS further away from the WLAN equipment.

To produce the WISA signals that cause interference on the WLAN, a WISA BS with a modified software is used. This modification enables the configuration of the FH sequence. This configuration of the BS is done through the GUI of ABBTraffic, but it is not fault free. The process that starts when the button encircled as number 6 in figure 4.4 might halt and ABBTraffic will wait indefinitely. This is believed to be a problem with the BS and not ABBTraffic, since

a similar problem is experienced when using a different program for configuration of the BS developed by ABB (RF Ctrl). This halt may eventually require a manual close of the ABBTraffic process. Such an operations may cause the computer to restart. The configuration might also fail even though the writing process finishes, and cause the BS to behave differently than what is configured through the GUI. The only way to detect this is by reading the spectrum analyzer to see if the correct frequencies are used. Usually this only requires a second attempt of writing to the BS, but if this also fails a restart of the BS is recommended. An attempt to avoid the latter problem was done by trying to read the current configuration of the WISA BS as a check to see if it had been configured correctly. This attempt was hampered by Java's lack of support of signed data values.

The use of Iperf and the transport layer protocols for testing the WLAN does not enable measurement of the delay of packets from one computer to the other. The delay might have been an interesting value to measure when the WLAN is interfered by WISA and would make ABBTraffic a more valuable software tool. Measuring the delay however, requires perfect synchronization of the clocks on both computers, which is not a straightforward task. A possible substitute of the delay measurement is Round Trip Time (RTT) which does not require clock synchronization. The measurement of RTT can not be done by Iperf and would require the use of *ping* or some other similar software. Although not measuring either RTT or absolute delay, ABBTraffic does provide a measurement of delay jitter, which in applications such as streaming of video and sound is a more important measure.

The software developed, ABBTraffic, does provide the ability to test the performance of the transport layer protocols TCP and UDP over a network. The network tested in this project is a combination of Ethernet and WLAN, were only the WLAN and its behaviour when interfered by WISA is of real interest. The metrics used are bandwidth, delay jitter and datagram loss of the entire network, not only the WLAN. The WLAN is however believed to be the bottleneck of the network, and the performance of the TCP and UDP protocols are clearly affected when the WLAN is interfered by a WISA BS. ABBTraffic also provides the ability to alter FH sequences for the WISA BS which seems to improve the WLAN performance: the less of the WISA channels that overlap with the WLAN channel that is used, the better the performance of the protocols. This does indicate that although ABBTraffic only reports the measurements of the metrics of the entire network, not only the WLAN, it is still useful for evaluating the performance of a WLAN interfered by a WISA BS.

Chapter 7

Conclusion

The developed software and assembled hardware presented in this report enable the generation of traffic between two computers communicating over a Wireless LAN. The bandwidth, delay jitter and datagram loss of this traffic can be observed on-the-fly as well as recorded for later analysis. Also the Wireless LAN data rate can be viewed on-the-fly or recorded. The hardware components include a Wireless Interface for Sensors and Actuators Base Station which is used to cause interference on the Wireless LAN. Through the software the effect of this interference can be observed by changes in bandwidth, delay jitter, datagram loss and data rates. The frequency hopping sequence used by the WISA Base Station is configurable and thus allows not only to test the effect of WISA's normal frequency hopping sequences, but also any future developed frequency hopping sequences. The software and assembled hardware thus form a tool which can aid the future testing of WISA's effect on nearby Wireless LAN's.

Bibliography

[1] Dacfey Dzung, Christoffer Apneseth, Jan Endresen, Jimmy Kjellson, Anne E Vallestad, Harald Vefling, “Air Interface Specifications“, Document number: 5382-03-10, ABB Corporate Research, Rev 1.11

[2] Richard Steigmann, Jan Endresen, “Introduction to WISA, WISA - Wireless Interface for Sensors and Actuators“, V2.0, July 2006.

[3] IEEE Std 802.11a-1999(R2003)

[4] IEEE Std 802.11b-1999(R2003)

[5] IEEE Std 802.11g-2003

[6] Atheros Communications, netIQ, “Methodology for Testing Wireless LAN Performance with Chariot“, http://www.atheros.com/pt/whitepapers/

[7] Andrew S. Tanenbaum, “Computer Networks“, fourth edition, 2003 Pearson Education Inc., Prentice Hall PTR.

[8] Dacfey Dzung, “Generation of new WISA FH sequences“, 2007-3-16.

Appendix A

Hardware details

WISA Base Station

Equipment Information

Product information:
Product designation: WDIO100
Product type: WDIO100-CONF-FBP
Serial number:
Software version:
Production date:

Producer information:
Producer: ABB STOTZ-KONTAKT GmbH
Address: ABB STOTZ-KONTAKT GmbH, Eppelheimer Str. 82, 69123 Heidelberg, Germany.
Internett address: www.abb.com

Equipment Specifications

Frequency band	$2.403gHz$ to $2.479GHz$
Medium access control	TDMA/FDD/FH
Transmit power	$0dBm$
Frequency dwell time	$2048\mu s$

DWL-2100AP

Equipment Information

Product information:
Product name: DWL-2100AP 802.11g/108Mbps Wireless Access Point
Model number: DWL-2100AP
Serial number: DR9X26C009545
Hardware version (H/W): A4
Firmware version (F/W): 2.20eu

Producer information:
Producer: D-Link Corporation
Address: No.289, Sinhu 3rd Road Neihu District Taipei city, 114, Taiwan.
Internett address: www.dlink.com

Equipment Specifications

Standards	IEEE 802.11g, IEEE 802.11b, IEEE 802.11,
	IEEE 802.3, IEEE 802.3u
Media Access Control	CSMA/CA with ACK
Wireless Frequency Range	$2.4GHz$ to $2.4835GHz$
Modulation Technology	OFDM, CCK, DQPSK, DBPSK
Wireless Transmit Power	$15dBm$ $(32mW)$ $\pm 2dB$
Receiver Sensitivity	From $-89dBm$ to $-66dBm$
	(Depending on signaling rate)
External Antenna Contact	RP-SMA

Parameters and settings

IP Settings	
IP Adress	192.168.0.50
Subnet mask	255.255.255.0
MAC Adress	00:19:5b:77:28:55
Services	
Limit Adminitrator IP	Disabled
SNMP	Disabled
Console protocol	Telnet
Console timeout	3 minutes

Wireless Settings	
Mode	Access Point
SSID	ABB
SSID broadcast	Enabled
Channel	Variable
Auto channel scan	Disabled
Authentication	Open system
Encryption	Disabled
Radio	On
Super G Mode	Off
Wireless Qos	Disabled
LAN Settings	
Get IP from	Static
IP Adress	192.168.0.50
Subnet mask	255.255.255.0
Default gateway	0.0.0.0
Advanced Wireless Settings	
Wireless band	IEEE802.11g
Data rate	Auto
Beacon interval	20
DTIM	1
Fragment length	2346
RTS length	2346
Transmit Power	Full
802.11g only	Enabled
Wireless Access Settings	
Wireless band	IEEE802.11g
Access control	Accept
MAC Adress	00:16:d4:bb:14:2e
MAC Adress	00:16:d4:ba:88:ff
MAC Adress	00:1a:73:1e:ef:08
MAC Adress	00:19:5b:44:e1:49
MAC Adress	00:13:ce:1d:94:82
AP grouping settings	
Load balance	Disabled
Link integrate	Disabled
DHCP Server	
DHCP Server Control	Disabled

Multi SSID	
Multi SSID	Disabled
WLAN state	Disabled
Administrator Settings	
Limit Administrator IP	Disabled
User name	admin
Password	
Console protocol	Telnet
Console timeout	3 minutes
SNMP	Disabled

Ethernet-to-WLAN bridge

Equipment Information

Product information:
Product name: Airplus XtremeG Ethernet-to-Wireless Bridge DWL-G810
Model number: DWL-G810
Serial number: B28I25C003203
Hardware version: C2
Firmware version: 3.15

Producer information:
Producer: D-Link Corporation
Address: No.289, Sinhu 3rd Road Neihu District Taipei city, 114, Taiwan.
Internett address: www.dlink.com

Equipment Specifications

Standards	IEEE 802.11g, IEEE 802.11b
Media Access Control	CSMA/CA with ACK
Wireless Frequency Range	$2.4GHz$ to $2.4835GHz$
Modulation Technology	OFDM, CCK, DQPSK, DBPSK
Wireless Transmit Power	$15dBm$ $(32mW)$ $\pm 2dB$
Device port	10/100BASW-TX Ethernet port
External Antenna Contact	RP-SMA

Equipment Parameters

IP Settings	
IP Adress	192.168.0.30
Subnet mask	255.255.255.0
MAC Adress	00:19:5b:44:e1:49
Wireless Settings	
Operating mode	Infrastucture
AP Name	DWL-G810
SSID	ABB
Remote Mac Address	000000000000
Channel	Variable
Authentication	None
Tx Rates	Auto
Transmit power	Auto
Super G Mode	Diabled
Administrator Settings	
User name	admin
Password	

Computer

Equipment Information

Product information:
Model number: HP G5000
Operating system: Windows Vista Home Basic

Producer information:
Producer: Hewlett-Packard Company
Internett address: www.hp.com

Equipment Specifications

Processor	Intel Core Duo T2250 $1.73GHz$
Memory (RAM)	$1024MB$
Java	1.6.0_01
Operating System	Windows Vista Home Basic
Network adapter	
Name	Realtek RTL8139
Connector	RJ-45
Transmission speed	10/100Mbps
Topology	Ethernet
Communication mode	Half/Full Duplex
Wireless network adapter	
Name	Broadcom 802.11b/g WLAN
Standards	802.11b, 802.11g
Modulation	CCK, DQPSK, DBPSK, OFDM
Transmit power	$15dBm$ (maximum)

Equipment Parameters

Computer no. 1	
Wireless network adapter	
IP address	192.168.0.3
Subnet mask	255.255.255.0
MAC address	00:1A:73:1E:EF:08
Network Adapter	
IP address	192.168.0.2
Subnet mask	255.255.255.0
MAC address	00:16:D4:BB:14:2E
Computer no. 2	
Wireless network adapter	
IP address	192.168.0.4
Subnet mask	255.255.255.0
MAC address	00:1A:73:29:13:96
Network Adapter	
IP address	192.168.0.1
Subnet mask	255.255.255.0
MAC address	00:16:D4:BA:88:FF

Cable

Equipment Information

Product information:
Product designation: CDF200-E LOW LOSS 50Ω COAXIAL CABLE
Product type: Low loss RF cable

Producer information:
Producer: Raison Enterprise Co., Ltd.
Address: 1F, No. 2-2 Alley 3, Lane 387, Nei-Hou Rd., Sec. 1, Taipei, Taiwan, R.O.C.
Internett address: http://www.commate.com.tw/

Specifications

Inner conductor	$1.2mm$ solid copper
Dielectric	Closed-cell PE
Outer Conductor	$3.66mm$ sealed AL./Mylar*/Al. + tinned copper braid
Standard jacket	$4.95mm$ black PVC
Nominal impedance	50Ω
Nominal attenuation	$55.4dB/100m$ at $2.5GHz$

Appendix B

Program details

The Java Classes

This section will list and briefly explain the function of each class that ABBTraffic consits of.

ABB_UI This contains all the GUI of the main window of ABBTraffic, it also contains some logic associated with this GUI.

Datagenerator An instance of this class is used for storing the measurements of bandwidth, dalay jitter, datagram loss and data rate. These data series are updated contineously and are fetched from an instance of the PacketReader class.

DynamicPlot This class is an implmentation of JPanel and basically contains all the parameters that control the look and behaviour of the real-tim graphs.

PacketGenerator This class creates a connection to the second computer (the one running ABBServer) and transfers the string that will be used to start the Iperf client on the second computer.

PacketReader The class that starts the Iperf server process on the local machine, this process accept the UDP/TCP traffic from the computer running the ABBServer. This class also parses the output of the Iperf server process and stores the them in local variables. Once a string of output from the Iperf server process in parsed, it will set a boolean that can be used by other classes to check if new data is available.

ReadFile An instance of this class can be used for reading a text file containing the data of previously saved results and presents them in a new Report Window.

RxRateReader This class can start an Expect process with a script file as input. This process' output is parsed and the data rate of devices connected to the AP with specified MAC's are read. This Expect process is repeated as fast as possible, and for as long as the RxRateReader thread is running.

WriteToFile The class that will write the results of traffic generation to a text file specified by the user.

generatorManager This class contains "all" the logic of ABBTraffic, it starts and controls the Datagenerator, PacketGenerator, PacketReader, RxRateReader and wisaBSwriter. A generatorManager thread is started by clicking the *Start*-button of ABBTraffic's main window. The generatorManager thread then starts the other threads in their correct order depending on the GUI iput of the user.

wisaBS A class for storing all the values that is to be used for configuration of the WISA BS.

wisaBSwriter This class controls the writing process to the COM-port that is used to connect the computer to the WISA BS. An instance of the wisaBS class stores the information to be written.

The source code for these files are found in zip-files accompnying this report.

Flowcharts

Two flowcharts are presented in this section; one illustrating the processes that take place when the *Start*-button encircled as number 10 in figure 4.3, the other is the processes that take place when the generatorManager thread runs. These two charts are meant to help understand the collaboration of the different java classes presented in B.

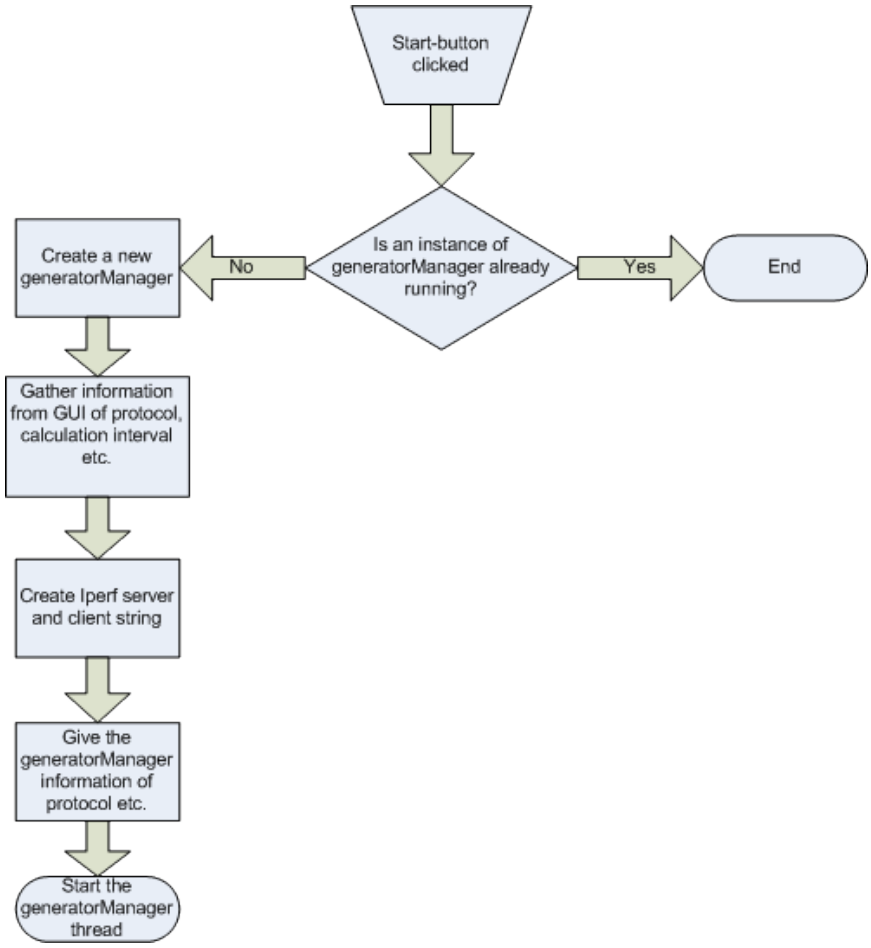


Figure B.1: Flowchart of processes after clicking the Start-button

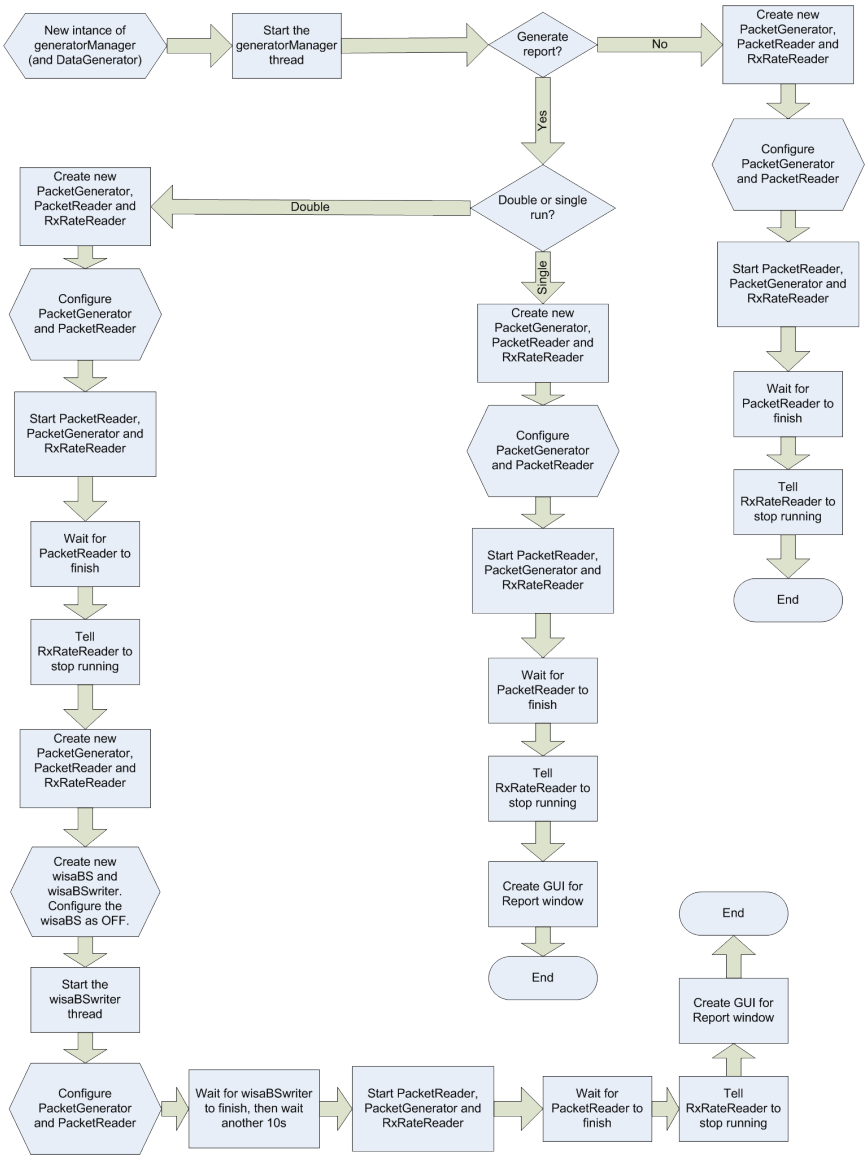


Figure B.2: Flowchart of the generatorManager thread

Configuration of WISA Base Station

The WISA signal will be generated with an actual WISA BS.

The BS should be configurable through the software and it should be possible to:

- Turn the BS on or off
- Mute selected frequencies of the hopping pattern
- Change the hopping pattern

To solve this list of requirements some modification of the software running on the BS was necessary, this was done by ABB.

The changes done by ABB allowed to set the BS in a mode. Two modes are used for this application, *Normal* and *Table*. In *Normal*-mode the BS follows the standard WISA FH pattern, which is a function of the cell-id. In *Table*-mode the BS will use two tables which are stored in its memory. The first table called the *Frequency*-table contains which channels to use and when. The second table is called the *Muting*-table and specifies which channels to mute. These tables are actually just byte addresses in the BS memory. The *Frequency*-table use address 0x2600 to 0x26FF, and can thus contain a maximum of 256 values. The *Frequency*-table contains entries of channels to use from 3 to 79. The BS reades the *Frequency*-table one entry after the other, starting from the 0x2600. A channel number may be repeated as many times as desired, upto 256 times. When the BS has reached the end of the table it will stop transmitting, unless the last entry is any valid channel number $+128$, then the BS will start reading the table from the start again and thus repeat the same pattern again.

The *Muting*-table works slightly different, by only containing 0's or 1's. The table have reserved the byte adresses from 0x2200 to 0x22FF, but only the adresses between 0x2203 to 0x224F[1] are actually used. This is a one-to-one mapping with the channels used by WISA, channel 3 is muted be writing a 1 to address 0x2203 and un-muted by writing a 0.

An example on how the *Frequency*- and *Muting*-table might be used can be seen in table B.1 and B.2 respectively. With these entries in the tables and the mode set to *Table* the BS will start its hopping pattern with channel 3 muted,

[1] 4F is the hexadecimal value for 79

jump to channel 5, jump to channel 78 muted and then to channel 79 before staring over again.

Byte address	Value
0x2600	3
0x2601	5
0x2602	78
0x2603	207

Table B.1: Sample *Frequency*-table on WISA Base Station.

Byte address	Value
0x2203	1
0x2604	0
0x2605	0
0x2606	0
...	...
0x264D	0
0x264E	1
0x264F	0

Table B.2: Sample *Muting*-table on WISA Base Station.

Writing to WISA Base station

These tables and modes are, as mentioned, simply memory addresses on the BS. The memory can store 2 bytes for each address and is indexed by a 4 byte word. These addresses are written to by a FPGA (Xillinx) based on values stored in its registers. The Xillinx writes to one address at a time based on the values in 3 of its registers. Each register can store a 2 byte word, therefore 2 registers are used to specify the address and 1 register for the actual value to be written. The Xillinx's registers cannot be accessed directly, but only through a PIC controller. The PIC controller also has 3 registers which are used in the same manner as on the Xillinx. To enable access to the Xillinx 'through' the PIC, a specific value must be written to a specific register on the PIC. Once this is done the memory on the BS can be written to be writing to three of of the PIC's registers. The values in the PIC's registers are written to the Xillinx's registers and finally to the BS memory. The register used an their functions on both the Xillinx and the PIC can be seen in table B.3. Writing to the PIC registers are done by writing 3 bytes[2] to the COM port connected to the BS. The first byte

[2]The bytes are treated as charachters in Java

Register number		Function
PIC	Xillinx	
11	3	Value
12	4	Low address
13	5	High address
15	7	Enable/disable peephole

Table B.3: Registers on PIC and Xillinx

is used to tell the PIC that this is a write operation, the second specifies the register the data should be written to and the third byte is the actual value to be written to the register. To change the mode of the BS to *Table*-mode one would want to write value 224 to address 0x2004 in the memory. The following procedure would then be required:

1. Enable peephole → Write value 132 to PIC's register 15
2. Set high address → Write value 32 to PIC's register 13
3. Set low address → Write value 4 to PIC's register 12
4. Write the actual data → Write value 224 to PIC's register 11
5. Disable peephole → Write value 128 to PIC's register 15

Sample FH Sequence File

This is an example of a text file containing the WISA BS, with cell ID 19, FH sequence. It is a simple text file containing only the numbers of the WISA channels, in their desired order.

```
3 35 56 77 21 42 63 7 28 49 70 14 46 67 11 32 53 74 18 39
60 4 25 57 78 22 43 64 8 29 50 71 15 36 68 12 33 54 75 19
40 61 5 26 47 79 23 44 65 9 30 51 72 16 37 58 13 34 55 76
20 41 62 6 27 48 69 24 45 66 10 31 52 73 17 38 59
```

Sample Report File

This is how a saved report file looks like. Such files are produced by ABBTraffic when saving the report by using button encircled as number 6 in figure 4.5. These files are used by ABBTraffic to generate a new report window when the user loads a previously saved result (see figure 4.2, circle 1).

```
Start time: 2007/04/23 10:00:59
Stop time: 2007/04/23 10:01:09

Protocol: UDP
Bandwith: 1Mbits/sec
Calculation interval: 1s
Local IP adress: 192.168.0.2
Remote IP adress: 192.168.0.1

WISA BS mode: normal
File name: *

**********************************************************
Bandwith data (1st run)
Time (s) -- c (Mbits/sec)
0.0 -- 0.0
1.0 -- 1.01
2.0 -- 1.0
3.0 -- 1.0
4.0 -- 0.988
5.0 -- 1.0
6.0 -- 1.0
7.0 -- 1.0
8.0 -- 1.0
9.0 -- 1.0
10.0 -- 1.01



**********************************************************
Jitter data (1st run)
Time (s) -- Jitter (ms)
0.0 -- 0.0
1.0 -- 3.477
2.0 -- 1.882
3.0 -- 0.062
4.0 -- 0.0
5.0 -- 0.017
6.0 -- 0.348
7.0 -- 1.907
```

```
8.0 -- 0.088
9.0 -- 0.139
10.0 -- 0.661



************************************************************
Packet loss data (1st run)
Time (s) -- Loss (%)
0.0 -- 0.0
1.0 -- 0.0
2.0 -- 0.0
3.0 -- 0.0
4.0 -- 0.0
5.0 -- 0.0
6.0 -- 0.0
7.0 -- 0.0
8.0 -- 0.0
9.0 -- 0.0
10.0 -- 0.0
```

expectScript.exp

The contents of the *expectScript.exp*-file.

```
spawn telnet 192.168.0.50
expect "D-Link Access Point login:"

send "admin\r"
expect "Password:"

send "\r"
expect ">"

send "get station\r"
expect ">"
send "quit\r"
```

File Structure

The file folder ABBTraffic is placed on computer 1 (see figure 4.1) and contains the files and directories specified in table B. All these files and folders exept

Name	Type	Description
lib	File Folder	Contains library files
ABB	Executable JAR File	ABBTraffic executable
expect	Application	Expect executable
expect52.dll	Application Extension	Expect dependency
ecpectlib52.dll	Application Extension	Expect dependency
expectScript	EXP File	Expect script file
iperf	Application	Iperf executable
slavedrv	Application	Expect dependency
tcl80.dll	Application Extension	Expect dependency
tclpip80.dll	Application Extension	Expect dependency
tclsh80	Application	Expect dependency
telnet	Application	Expect dependency
test	Text Document	FH sequence
tk80.dll	Application Extension	Expect dependency

Table B.4: Files at computer 1

for the “test.txt“ are necessary for ABBTraffic to run on computer 1. The ABB-Traffic is launched by starting a command window and executing the “java -jar ABB.jar“-command in the ABBTraffic folder.

The file folder ABBserver is placed on computer 2 (see figure 4.1) and contains the files specified in table B. The ABBServer is launceh in a command

Name	Type	Description
ABBServer	Executable JAR File	ABBServer executable
Iperf	Application	Iperf executable

Table B.5: Files at computer 2.

window by executing the command “java -jar ABBServer.jar“.