Comparison of Time- and Event-triggered strategies for wireless communication in embedded systems

Design of a lab assignment for university level course on data communication in embedded systems

Thesis for the Degree of Bachelor of Science,

Computer Network Engineering

Students: Emil Sjöström, Martin Andersson

Examiner: Elisabeth Uhlemann

Supervisor: Svetlana Girs

Academy for Innovation Design and Engineering

Mälardalen Unversity

June 2016

ABSTRACT

Embedded systems (ESs) and wireless technologies are in constant rapid development and therefore it is important to educate people in these subjects. This thesis work has the goal of developing a lab assignment for a university level course on data communication in embedded systems. Embedded systems have reliability and timeliness requirements that have to be fulfilled. These criteria pose problems to be addressed while ESs are used in conjunction with communication over wireless medium, which has some underlying limitations. These limitations come in the form of interference, degradation of signal strength with the distance and possible unwanted delays. This thesis work starts with a literature study on medium access control (MAC) methods and their possibilities for adoption of time-triggered and event-driven strategies for ESs with time constraints and continues with the lab assignment design, implementation and testing. Taking into account timing requirements brought by ESs and the specifics of the lab, i.e., limited time given to the students to solve the problem and compulsory usage of available lab equipment, a design with two communicating raspberry Pis was proposed. The system consists of a sensor node sending its readings to a controller, which is responsible for analyzing the data and actuating a set of LED lights as a response to the input data. The communication is done over a WiFi network and three different programs, organizing the communication in time-triggered, event-driven or a combination of the two fashions are developed. Each program is tested under three environmental conditions and the results from these tests clearly show the limitation of the underlying CSMA/CA MAC adopted in WiFi and give a greater understanding of the advantages and disadvantages of different strategies for communication design.

CONTENTS

TABLE OF FIGURES

INDEX OF TABLES

1. Introduction

Today embedded systems are an integral part of our lives, even if we do not always realize it. Everything from industry to our daily lives is becoming more and more automated and optimized. This is, in part, made possible by a large number of embedded systems working together in a centralized or distributed manner. To continue the rapid expansion and progress brought by embedded systems it is imperative to keep the competence in the field high. Therefore, one of the goals of the thesis is to design and implement a lab assignment for a University-level course focused on communication in embedded systems. The course itself has the goal of increasing students' theoretical and practical knowledge and giving them the ability to comprehend and implement advanced data communication protocols in distributed embedded systems and analyze the collected data. Thus, in this thesis work we will focus on wireless technologies used in embedded systems with time constraints, to design and compare different solutions for communication between applications that generate time- and event-triggered traffic. The design of the lab assignment is limited to using Raspberry Pis, WiFi and components compatible with the Raspberry Pi.

1.1 Problem formulation

Both embedded systems and wireless technologies have been in rapid development in recent years and can be very convenient to use together. The problem with this is that most wireless technologies struggle with providing the required reliability and timeliness, which many embedded systems depend on. We will therefore make an evaluation of different ways of implementing the communication, i.e. time-triggered or event-driven, for systems with timing constraints and study how different design choices affect the performance of the system given the selected underlying MAC protocol. From the initial literature study, a lab assignment focused around this idea will be designed, which should increase the students' understanding of the relevant subjects. The following research questions have been the basis of this thesis work.

- How can embedded systems be classified based on their applications and timing requirements with respect to suitability for time-triggered or event-driven strategies?
- How can wireless communication be designed to better support the timing requirements brought by applications given that they generate time- or event-triggered traffic?
- Given the goals of the targeted course and the focus on wireless technologies, how should the lab assignment be designed to be relevant and beneficial for future students?

1.3. Method

The research methods used in this thesis work will include literature studies, tests performed on the real lab platform and numerical evaluations. The work starts with a literature study about different types of embedded systems, time- and event-triggered traffic, characteristics of wireless medium and various channel access protocols. Since the thesis work includes the design of a lab assignment using the Raspberry Pi, the literature study extends to include this topic as well. Literature research is expected to give more insight into the subject and the initial research questions and might result in a need to redefine this aspect of the work. From the knowledge gained, a hypothesis should be formulated and predictions made of outcomes of the future work. This is followed by testing the hypothesis in a lab environment designed specifically for this thesis. In the event that the results are inconclusive or the design proves lacking it might be necessary to go back to an earlier stage, reevaluate and try again. Figure 1 presents a visualization of the followed method.

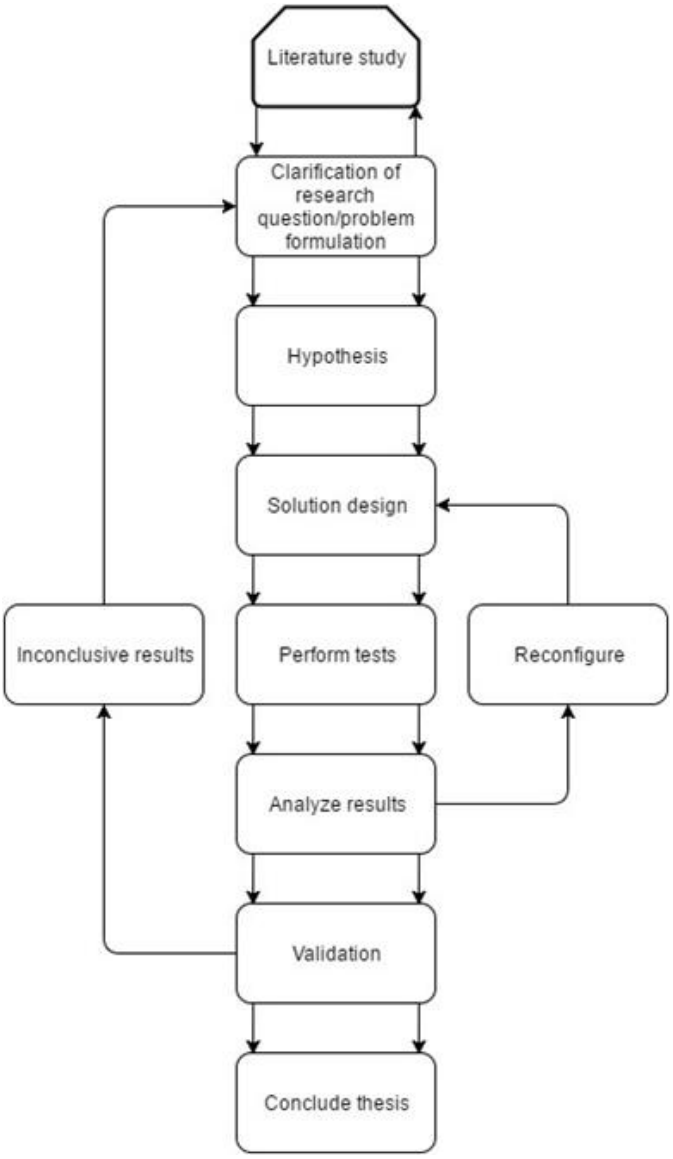


Figure 1 Research process steps

1.4 Thesis outline

The remainder of the thesis is structured as follows: Chapter 2 gives background information about embedded systems and the reliability and time constraints these systems demand, while Chapter 3 presents describes possibilities for wireless communication for embedded systems. The goal of Chapter 3 is to explain the difficulties posed by using wireless technologies with embedded systems. The thesis continues with Chapter 4 presenting related research and several different ideas for a lab assignment. Due to limitations set by hardware and cost issues a design centered around the use of CSMA/CA (Carrier sense multiple access with collision avoidance) is chosen and the implementation explained in Chapter 5. Next, Chapter 6 presents the results of several tests performed under different environmental conditions and analyzes the gathered data. The thesis finishes with Chapter 7 presenting conclusions derived from the thesis work. Appendix A contains detailed information about the setup done during implementation.

2. EMBEDDED SYSTEMS

The term embedded system (ES) in computing disciplines is used to refer to an electronic system that is designed for a specific function within a larger system. Traditionally an embedded system is composed and designed with a microprocessor running dedicated software, in contrary to general-purpose computing devices hardware. Devices like home computers and cellphones have their hardware and software usually designed separately and their role is to serve a variety of different functions e.g., web servers, game consoles and more [1, pp. 3-4]. Embedded systems are found all around us and integrated into basically all electronic devices in the market. A tour in an average person's home will highlight how widespread the usage of embedded systems has become. Some examples are dishwashers, garage doors openers and remote controllers. ESs are often parts of bigger systems and many bigger systems, e.g., cars and control systems, are dependent on several ESs to function properly. The complexity of ESs has increased dramatically due to the advancement of microprocessor technology and miniaturization of these chips.

2.1 Real time embedded systems

A larger system of ES units is not only dependent on correct computations but also on the physical time when the results are produced [2, p. 2]. In an ES the start of an event and the termination of an event have to be determined so that a larger system of ES units can work synchronized, e.g., an input like the push of the elevator button does not open the door until the elevator is on that floor. ESs depending on a specified time (i.e. deadline) are called Real-time embedded systems (RTES). Within RTESs the result of missing a deadline can have different outcomes. If missing the deadline causes system failure, then the RTES is called a hard real-time system. Alternatively, if a deadline miss causes system performance degradation, i.e. the information received is still operationally useful for the system, a system is called a soft real-time system [3, pp. 13-14]. An example of a hard RTES would be the stop button on a mill production line, which, when pushed, triggers all machines to stop. An example of a soft RTES would be the global positioning system used in cars to identify waypoints, where a delay of some seconds can result in the driver missing the waypoint. In a RTES the dependability of the system relies on the reliability of the system i.e., the probability that the system will provide the specified service until time *t*. Thus, in hard RTESs the requirements for reliability are much higher than in soft RTESs due to the consequences of a failure, e.g., loss of life compared to missing a waypoint.

2.2 Real-time communications

Several embedded systems can work in conjunction with one functionality being distributed among the different ESs. These nodes communicate through an interconnected network and this kind of an architecture is called a distributed embedded system. When dealing with real time requirements in an architecture of this kind, not only processing software within the nodes, but also message exchange between the nodes plays a crucial role. These temporal restraints require the communication channel to adhere to specific timing constraints and these communication types are known as real time communication [3, p. 14].

In a distributed RTES the internal trigger that elicits a response e.g., the action of transmitting messages through the communication channel (i.e., media access channel) can be divided into two formats, event driven and time-triggered [2, p. 16]. Event driven systems (EDS) are directed by significant events that initiate all processing and communications activities. When an event occurs, it evokes in the central processing unit of the ES a dynamically scheduled response. It does this by an interrupt mechanism [4] to force the CPU to deal with the event, e.g., an airbag should only open on impacts and with precise timing, as if it opens too early it might seriously injure the driver, too late might be too late to save the driver.

Time-triggered systems (TTS) are driven by the progression of time; all activities (e.g., processing or communication) are based on predetermined ticks of an internal clock. The only interrupt mechanism in the nodes of an TTS is the periodic real-time clock interrupt, meaning that each event is time driven and scheduled beforehand. The requirement for an TTS is that the interconnected nodes have the same global time, meaning the internal clocks of all ESs have to be synchronized. Events that occur in these systems are timestamped with this global time and the granularity of the global time must be chosen so that the observed time order of two events in an RTES can be established [2, p. 17].

The two methods of triggering a response in a RTES have different recommended areas of application. TTSs are more appropriate for control systems that collect periodic data e.g., periodic data about the temperature from a sensor, whereas ETSs are more suitable for systems with sporadic events e.g., an alarm that actuates a response. At the same time, there are several benefits of mixing the two methods to give an RTES the traits of both desired profiles. For example, in the case of sensor data being periodically collected, an alert could be generated if the temperature exceeds a certain threshold thus eliciting a response [3, p. 14].

Application of RTESs within industrial and home environments has increased specifically the use of RTESs that apply wireless technologies for sending and transmitting data. In industrial environments the benefits of reducing cabling expenses and the added mobility of small RTESs make them ideal for inexpensive data gatherers e.g., temperature, humidity, and smoke sensors. However, adoption of wireless communication in industrial environments characterized by the presence of electromagnetic noise, multiple moving objects and reflective surfaces, introduces challenges on how to maintain reliability levels required by industrial applications. The consequences of a delay in temperature readings can cause systems to go over hard deadlines which might incur high economic loss or even danger to human life. The next section deals with how wireless technologies function and how they can be applied to real time communications.

3. COMMUNICATION IN EMBEDDED SYSTEMS

The timing and reliability constraints dictated by embedded systems make reliable communication over wireless medium, with its inherent limitations, a problem to be solved when used in conjunction. This section will provide suitable background information about the wireless medium, its access methods and selected protocols to understand the difficulties posed by the collaboration of embedded systems and wireless communication.

3.1 Wired and wireless medium

To better understand the content provided it is important to know the fundamental difference between the wired and wireless medium. Signals in wired communication systems travel through either a copper wire or fiber optics. Signals traveling through copper wire use electrical signals to interpret ones and zeroes, while fiber optics use light pulses to do the same. Wireless communications use radio signals and different modulation techniques to differentiate the signals and interpret them as ones and zeroes, thereby creating a so called carrier signal. There are three components of a radio signal that can be modified to create an interpretable carrier signal.

- Amplitude - the height or power of the radio wave.
- Frequency - a parameter showing how fast a radio wave travels or how many waves are created per second.
- Phase - a relative term meaning the relationship between two waves with the same frequency.

Since wireless technologies use radio waves that travel through the air, they are more susceptible to outside interference compared to e.g., wired solutions where an electrical signal is running through a wire. Also, wireless communication mostly operates in half-duplex. This means that devices using the technology cannot send and transmit at the same time and must therefore have medium access control (MAC) protocols that keep this in mind [5, pp. 17-20].

3.2 Medium Access Control

Before we start going into the subject of wireless communication specifically for embedded systems, it is important to understand the fundamentals of medium access protocols. Medium Access Control protocols operate in the data link layer of the OSI model, right above the physical layer, and exist to solve two crucial tasks. Firstly, to assist with the construction of network infrastructure and secondly to control how several nodes can fairly access the medium in a shared network environment [6]. For MAC-protocols the most important performance requirements are traditionally: fairness stability, throughput, low access delay, low transmission delay and low overhead. Table 1 gives an overview of these concepts.

Table 1 Important performance requirements

Concept	Function
Fairness	Equal access to medium
Stability	Reliability and uptime
Throughput	Rate of successful message delivery
Access delay	Time between packet arrival and first attempt to transmit it
Transmission delay	Time between packet arrival and successful delivery
Overhead	IE. Control data, Error checking, Metadata

The performance of MAC-protocols is in turn heavily influenced by the underlying medium and as such inherit the problems that comes with it. In our case, this would include well known wireless problems as the time-variable error rates due to path loss, fading, shadowing, and man-made or thermal noise.

A vast number of wireless MAC protocols has been developed over the years and they can at large be categorized into two groups depending on which access scheme they use. These groups are those that are based on scheduling or resource allocation, also called contention free and those that are contention based. The next sections will describe the different schemes and relevant protocols developed using them [7, pp. 111-114].

3.2.1 Contention based access methods

Methods that are classified as contention based allow several nodes to share the same medium, but compete for the access to it. Nodes attempt to send whenever there is data, but instant access is not guaranteed. This section will present ALOHA and Carrier Sense Multiple Access (CSMA).

A. ALOHA

ALOHA is a method developed by the university of Hawaii and comes in two main forms, Pure ALOHA and Slotted ALOHA. A node using Pure ALOHA transmits its packets instantly when it is ready to send, without any coordination with the other nodes operating in the network. This means that pure ALOHA accepts the risk of collisions on the network. To detect these collisions a receiver is required to send an acknowledgment (ACK) frame when a packet is correctly received. The node interprets a lack of ACK as a collision has occurred on the network. At the point when the transmitter has interpreted the lack of ACK it initiates a random back-off value and then tries to transmit again, Figure 2 shows the basic process of ALOHA.

In slotted ALOHA, the time is divided into time-slots and packet transmission starts only at the beginning of one of these slots. One slot is big enough to hold a maximum-length packet. This means that only nodes transmitting in the same slot have a chance of causing a collision. This synchronization results in a reduced probability of collisions and as such slotted ALOHA has a higher throughput than pure ALOHA [7, pp. 115-116].

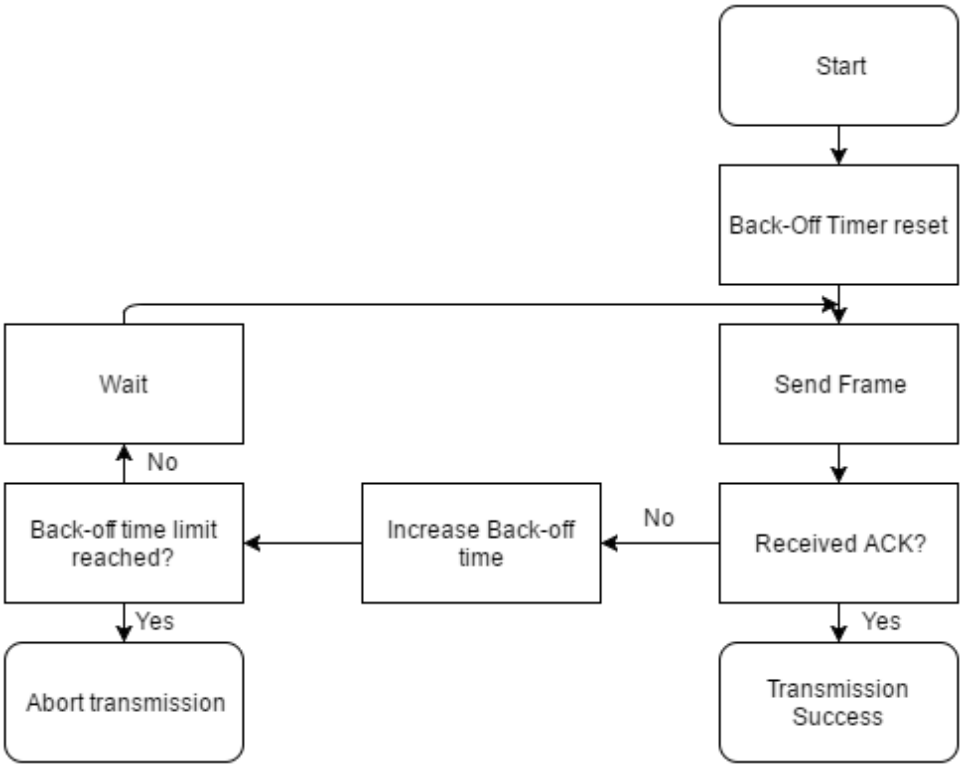


Figure 2 Basic operation of ALOHA

B. Carrier sense multiple access

CSMA is a contention based access method that comes in two forms, CSMA with Collision Detection (CD) [8, pp. 268-269] and CSMA with Collision Avoidance (CA) [8, pp. 303-309]. CSMA/CD is widely known and used in Ethernet networks, while CSMA/CA is primarily used in wireless systems. In both methods the devices first listen to the network (also known as carrier sensing) to make sure that the media is available, and if it is not - back-off. The difference between these two methods becomes clear when a node wants to transmit and no other nodes are currently transmitting. A node running CSMA/CD can begin its transmission immediately and if a collision happens, it can detect it and stop transmitting temporarily. CSMA/CA, due to the half-duplex nature of the wireless medium, cannot transmit and receive at the same time, which results in inability to detect a collision during transmitting. What 802.11 stations do instead, when ready to transmit, is initiation of a random back-off timer. The stations wait for this time before transmitting while continuing using carrier sense to listen to the medium. If during this time another node starts transmitting, another back-off timer is initiated and the process starts over. By doing this, the collisions are kept to a minimum as only one node should have access to the medium at the same time. Figure 3 shows the basic operation of the CSMA/CA mechanic.

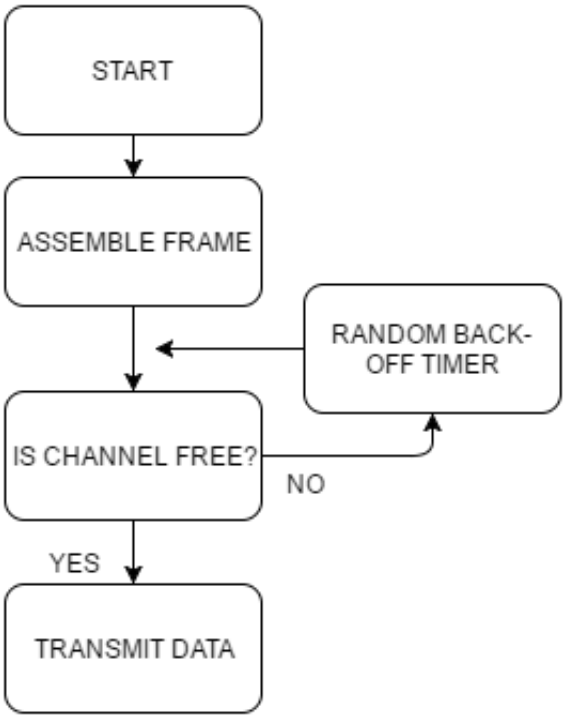


Figure 3 Basic CSMA/CA operation

3.2.2 Contention free access methods

Methods classified as contention free usually divide the medium in some way between the nodes to create an environment that is collision free and predicable. This division can come in different forms, such as time-division, frequency-division and code-division, show in Figure 4

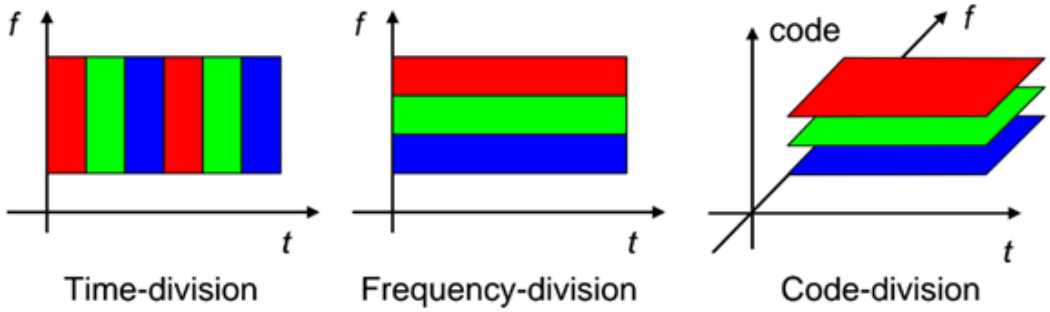


Figure 4 Visualization of contention free MAC **[9]**

A. Time Division Multiple Access

In time division multiple access (TDMA) schemes the time is divided into different time-slots and each node is given an appropriate amount of slots for transmitting and receiving its data. By doing this TDMA allows multiple nodes to use the same frequency channel, but not at the same time. TDMA requires strict time synchronization to make use of the time-slots, as every node must know when to transmit and receive its data. To inform and manage different nodes participating in the communication, a central control node is often present in the network. This node periodically sends beacon frames with the TDMA frame which contains the management information [2].

B. Frequency Division Multiple Access

This method uses another form of division to create a collision free environment. Frequency division multiple access (FDMA) divides the available frequency band into sub-channels and assigns each communications pair of nodes one of these channels. The nodes can then only transmit on the assigned channel and the total bandwidth is divided among them. FDMA requires frequency synchronization, bandpass filters and relatively expensive transceivers, making this method more complex than a TDMA-scheme [8, p. 258].

C. Code Division Multiple Access

Code division multiple access (CDMA) uses spread spectrum technology to cast signals over a much wider frequency band than other methods. The transmissions are then separated by using coding theory that identifies different data streams. The code must be recognized by the receiver and transmissions that are not recognized are treated as background noise. A very important aspect of CDMA is code management to make sure that all nodes recognize the relevant transmissions. The identification is made possible by the transmitter and receiver using the same code sequence. A simple analogy is to think about CDMA as dividing transmissions into different languages and only nodes that speak the same language are able to speak to each other and all other unrecognized speech is rejected. [10]

3.3 MAC-protocols for embedded systems

Due to the requirements of embedded systems some MAC-methods and protocols are better suited than others. CSMA, because of its random access nature, would not be well suited for embedded systems as medium access delays cannot be predicted in its original form. ALOHA would give instant access to the medium, but the risk of collisions and thus random back-offs results in unpredictability unsuited for embedded systems. TDMA [6], however, ticks a lot of the boxes for embedded systems.

TDMA-properties suited for embedded systems:

- Guaranteed access to the medium
- Predicable maximum access delay
- Collision avoidance

The next section will cover some of the protocols and standards used today having the qualities desired by embedded systems.

3.3.1 TDMA

TDMA is widely used in industrial standards, e.g. in ISA 100.11.a [11], WIA/PA [12] and WirelessHART [11], which is considered in more details in this thesis work. WirelessHART is a protocol that uses the IEEE 802.15.4 standard, Direct Sequence Spread Spectrum (DSSS) radios and a TDMA MAC-protocol on top [13]. It operates in the 2,4ghz ISM band and utilizes frequency hopping spread spectrum (FHSS) to hop between channels and avoid interference. All devices operating with WirelessHART are strictly synchronized and communicate using fixed length time-slots. The technology offers a high reliability factor in harsh industrial environments. This reliability is largely provided because of the channel hopping technology and the self-healing qualities provided in the mesh network built by WirelessHART. Due to its TDMA-based channel access and time-slots allowing exactly one transmission at a

time, WirelessHART can offer predictable transmission latencies desirable for embedded systems [14, pp. 3-6].

3.3.2 IsoMAC

IsoMAC [13] has its roots in the 802.11 standard which is CSMA/CA based but incorporate TDMA on top of this. A centralized manager assigns time-slots based on requests sent by each node, through CSMA. Each node receives, through the use of beacon frames, a time schedule and synchronization data from the manager. IsoMAC incorporates a contention phase as well as a scheduled phase. The contention phase is used for best effort traffic and, as previously mentioned, requests for time-slots. The scheduled phase consists of the time-slotted traffic flows.

3.3.3 SMACS

Self-Organizing Medium Access Control for Sensor Networks (SMACS) [2] is a protocol that implements the eavesdrop-and-register (EAR) algorithm when it starts up a new network and discovers nodes for neighbor association. SMACS utilizes superframes, each of which is divided into time-slots. New neighbor connections are slotted a time-slot in the superframe and during their allotted time-slot the established neighbors can send and receive traffic. Due to the lack of global synchronization, the SMACS protocol utilizes different randomly chosen frequencies for each link. This gives an upper limit on how many new connections a node can establish based on the available frequencies. Power consumption is something that can be managed through turning of nodes on time-slots when they are not scheduled to send or receive data.

3.5.4 PACT

PACT [6] stands for power aware clustered TDMA and uses a principle called passive clustering to preserve battery life and network lifetime. Nodes in a PACT protocol environment shift the responsibility of traffic between PACT cluster and the allocating of time-slots. These cluster heads are chosen based on their energy reserves and rotate to prolong the lifetime of the PACT network. This protocol shows through simulations an increase in network lifetime compared to 802.11 networks.

3.5.5 Traffic-Adaptive Medium Access Protocol

The TRAMA [15] protocol is traffic-adaptive and suitable for use in wireless sensor networks (WSN). With the use of TDMA the protocol aims to keep the collisions to a minimum and by doing this save power on the nodes (a requirement in WSN). To aid in the endeavor to save power the protocol also utilizes a low power usage mode on the nodes which are currently not scheduled to send or receive data. The TRAMA protocol consists of the mechanisms that are responsible for different aspects of the MAC. The neighbor protocol (responsible for exchange of neighbor information), the schedule exchange protocol (responsible for exchanging the appointed schedule throughout the network) and the Adaptive Election Algorithm (AEA). The AEA uses the information gathered and distributed by the previous mechanisms to prepare a schedule for the next time-slot.

4. DESIGN OF LAB ASSIGNMENT

For our lab design we propose to use two Raspberry Pis. Both Pis will wirelessly communicate with each other; several communication designs will be available and compared. The lab's architecture will consist of a sensor node that collects temperature readings and sends them to a controller that analyzes the data and depending on if the data is received, any loss packets are detected or if an alarm is triggered, lights up some LED lights for visual confirmation. Looking at Figure 5 the "sensor node" would then include a temperature sensor, one of the Pis and a wireless transceiver, while the control node would consist of the other Pi and LED lamps and a wireless transceiver connected to it.

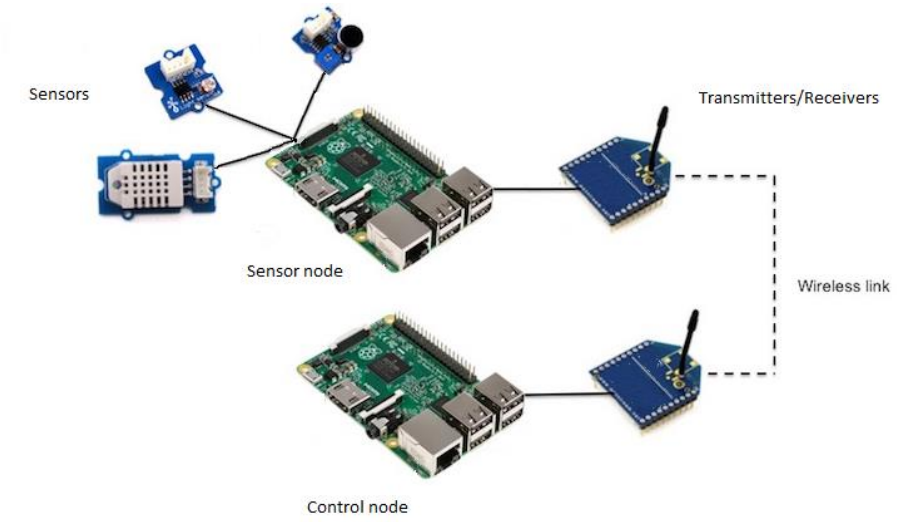


Figure 5 Initial mockup of lab design

With the design in place, several performance tests should be conducted to show as many aspects of the wireless design as possible. Delay and packet error rate are among the most relevant to evaluate the chosen architecture. The tests will include different set conditions of the sending and receiving of frames (e.g., without interference, with interference, with congestion and with congestion and interference), and the data collected will then be the basis for our final assessment of the design solution.

During the conducted literature study, it has become clear that most protocols used in systems with strict timing requirements, such as e.g. a control system consisting of a sensor node sending its readings to a controller, incorporate some TDMA-functions to guarantee deterministic channel access. The following section will therefore describe related research and different design possibilities for the lab assignment intended to show the qualities of different communication solutions.

4.1 Related research

Many studies have been done on how to apply wireless technologies for embedded systems in industry and home automation. The paper [16] shows the problem with the integration of wireless sensor networks (WSNs) and actuator networks i.e., WSANs. It suggests enhancements to the WirelessHART protocol to add downlink transmission from actuator networks through a logical programmable controller (PLC) to the sensor network. As it stands now, the WirelessHART is the first complete international and industrial standard for real-time processes. WSANs have some inherent problems when it comes to unpredictably delay, packet loss, and interference from other wireless technologies. The paper proposes a new service to the Hart protocol to enable a deterministic transmission from the gateway to actuators to solve or mitigate these problems that WSANs are subjects to. Further on the paper proposes a mechanism to enable actuators to enter a failsafe mode i.e., safety by design.

Several approaches of implementing TDMA schemes on top of CSMA hardware have been proposed and [13] highlighted the benefits of three proposed mac approaches for time constrained, rate constrained and best effort data. The first approach uses pre-scheduled time-slots, where TT and RC traffic have been allotted specific amounts of slots and the rest of the slots is left for BE traffic. The BE traffic is either assigned beforehand or alternatively the slots are handed out in a round robin fashion. The second approach is a combination of pre-scheduled and contention based time-slots. Like in the first scheme, TT and RC traffic have their allocated slots, but BE traffic contents for the remaining time-slots. In the third scheme, a division between pre-scheduled time-slots and contention based channel access is done. The TT and RC flows are grouped into a scheduled time-slot phase and BE traffic is allocated a continuous contention based phase.

Other research works focuses on the spread use of 802.11 networks and suggest using the 802.11e addendum [17], specifically the enhanced distributed channel access (EDCA) mechanism, which is part of the QoS and offers priority classes for different traffic. The approach in the paper is called Sched WiFi and proposes to use the EDCA mechanism for low and predictable end-to-end delay values for high priority traffic that is regularly scheduled over 802.11 networks.

A new approach on how to change used MAC protocol depending on the environment where a system is deployed is being studied in the literature. One proposed idea is to define the MAC protocol in software and not hardware, which is the standard approach for vendors. The authors of [18] propose a way of designing and implementing a software defined radio architecture capable of supporting core MAC function that can implement high performance MAC protocols like CSMA. The implications of this approach is huge due to the flexibility it would offer in deployment for different wireless systems and applications.

4.2 Design propositions

After the conducted literature study three different design possibilities were proposed to give the foundation for the lab environment in which to evaluate different strategies for communication.

4.2.1 Proposition 1

A paper on implementing a software based TDMA scheduling (Soft-TDMA) [19] gave us the idea of implementing their solution onto our problem as one of three mac-schemes to compare on 802.11 hardware. The paper describes a method of implementing global time with a minimal synchronization offset of microseconds between nodes. This makes it possible to time-slot the channel access and frames can be queued waiting for their scheduled transmitting window. The protocol uses commodity 802.11 hardware that has the QOS service parameters turned on, which in turn disables CSMA/CA. This solution solves the problem of collision and back-off times and gives the nodes predictable transmission times where the propagation delay over the medium can be calculated. The network consists then of nodes who synchronize to a gateway i.e., a base station and use it as a timing reference.

The second method would be to use the Point-Coordinated Function (PCF) that the 802.11 standards describe [20, p. 64]. This extension to the 802.11 standard describes a mechanism, where a point coordinator, usually an access point (AP), performs a connection oriented and contention free frame transfer. The AP polls each PCF enabled client and grants them an access to transmit without contending. The third method chosen for the comparison was the default behavior of the 802.11 standard, i.e. basic CSMA/CA.

4.2.2 Proposition 2

Second design proposal would be to manipulate the network stack of an 801.11 compliant network interface card and implement different mac-schemes. As mentioned in Section 4.1 there exist several approaches of overlying different mac schemes on commercial 802.11 hardware. We propose to compare the methods in [13] and [17] with standard 802.11 CSMA/CA.

4.2.3 Proposition 3

Looking at the hardware from different vendors is might be possible to buy dedicated hardware and utilizes different MAC-schemes for sending and receiving data. Specifically, a comparison of commercially off the shelf available wireless solutions. We propose to compare Bluetooth [21] as a polling mechanism, WiFi [5, pp. 221-241] as a contention based mechanism and Zigbee [22] as a TDMA mechanism.

Bluetooth utilizes a master slave model to control when and to whom the data is transferred. A master node can have up to seven slaves and comprise a piconet network; each slave in a piconet network can be a master of its own piconet. Clusters of these treelike networks are called scatternets. Bluetooth supports two channel modes for sending data between the master and slave(s): asynchronous and synchronous. Both modes are TDMA based, but asynchronous is a poll based channel mechanism where the master polls each slave for information and the slave responds in the next time-slot. Synchronous channel mode is set up before hand and communication between master and slave occurs periodically, in i.e. in time-triggered manner, in contrary to the more event driven asynchronous mode.

ZigBee is a wireless specification that conforms to the 802.15.4 -2003 Low-Rate Wireless Personal Area Network (LR-WPAN) standard and uses a CSMA/CA mechanism for channel access. It differs from pure 802.11 CSMA/CA as it can operate either in slotted CSMA/CA or unslotted CSMA/channel access. Furthermore, nodes in a ZigBee network can be allocated guaranteed time-slots, thus implementing a TDMA like scheme on top of CSMA. A Bluetooth adapter and a ZigBee adapter would then be compared to standard WiFi adapter and put through a performance tests.

4.3 Selection of the communication design

All three of the researched designs should be able to provide a solid foundation for our tests. However, budget constraints and the fact that the lab assignment must be applicable for a class of students makes the design proposition described above too expensive and some of the technical aspects of the design propositions too difficult to implement. Technical aspects of the design propositions 1 and 2 have a steep learning curve and thus, course students would not be able to finish the assignment within the allocated time.

Given the limitations above, we suggest to use off-the-shelf WiFi devices with CSMA/CA MAC. These transceivers are easy to use with the Raspberry pi system and thus the students will be able to quickly set up all the lab equipment. Thus, we suggest to implement different communication designs on top of existing CSMA MAC. The main task of the lab would be then to evaluate the work of time-triggered and event–driven communication schemes operating in an affordable CSMA/CA environment. The idea is that each Raspberry Pi communicates with a WiFi dongle and a specially written program structures how information should be sent to the network card. Three possible communication scenarios are to be coded: time-triggered, event-triggered and a combination of the two.

5. Implementation of final design

The following section will cover the implementation of the proposed lab design. In this chapter each component will be listed, the setup - shown and explained, while the programming code and more detailed configurations will be included in Appendix A

5.1 Components

- Raspberry Pi 2b x2
- Breadbord 400 holes
- T-cobbler pack
- Jumper wires
- Led lights
- Edimax 150Mbps Wireless nano USB adapter
- DIGITAL THERMOMETER, DS1820+

Each component has been selected on the criteria to be compatible with the Raspberry Pi system.

5.2 Setup

The two Raspberry Pis are divided into specific roles. One acts as the control node and, as such, receives data and reacts to it. The other Pi is our sensor node that connects to the control node and sends its sensor data over the network. Also, there are three LED-lights connected to the control Pi and used to indicate certain states, i.e. alarm, packet loss and correct data reception. The sensor node consists of the Pi connected to a sensor via a breadboard and cobbler cable.

5.2.1 Control node

The three LED-lights are connected to GPIO pins 18, 23 and 24 on the Raspberry through the breadboard. A 330ohm resistor is placed between each light and power source. Figure 6 shows an overview of the control node with connected LED-lights.

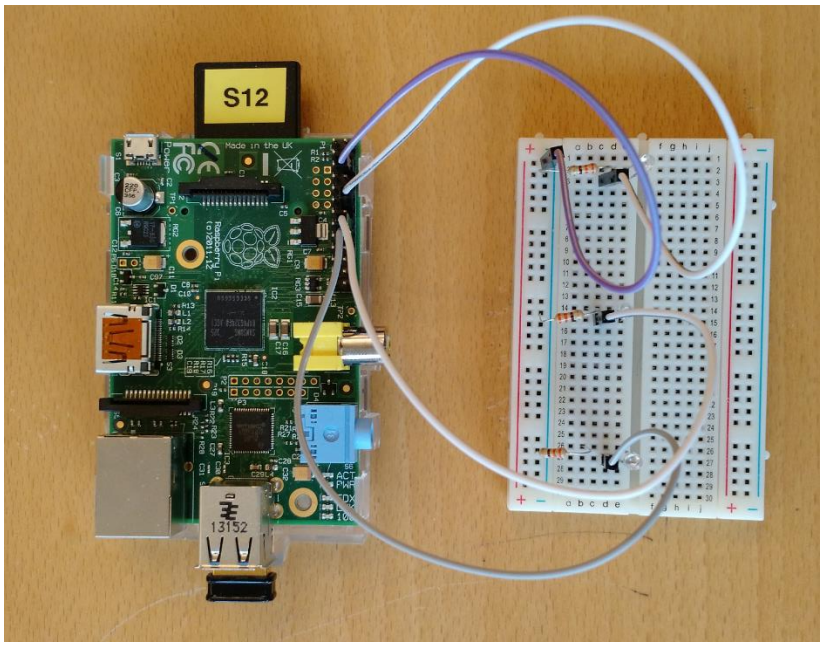


Figure 6 Overview of Raspberry Pi with LED-lights connected

5.2.2 Sensor node

A cobbler cable connects the GPIO pins on the Raspberry to the breadboard and the sensor is placed with the curved edge facing right (as seen in Figure 7). The breadboard facilitates connections as each numbered row is a circuit. The plus and minus columns are also each one a circuit and can be used to connect several devices to one ground connection or one power source. The T-Cobbler extends the GPIO pins on the Raspberry onto the breadboard. This means that the 3,3 volts GPIO pin for example can be accessed by all holes in row 30 on the left side. The sensor has three pins and when looking from above and the curved edge facing outwards makes the middle pin the data pin, the right one the power pin and the left pin the ground pin. Using jumper cables GPIO pin 3,3 is coupled to the power pin on the sensor, GPIO pin 4 is coupled to the data pin sensor and a GPIO GND pin is coupled to the ground pin on the sensor. A 4,7K resistor is placed between the data and power connections for stable data transfers.

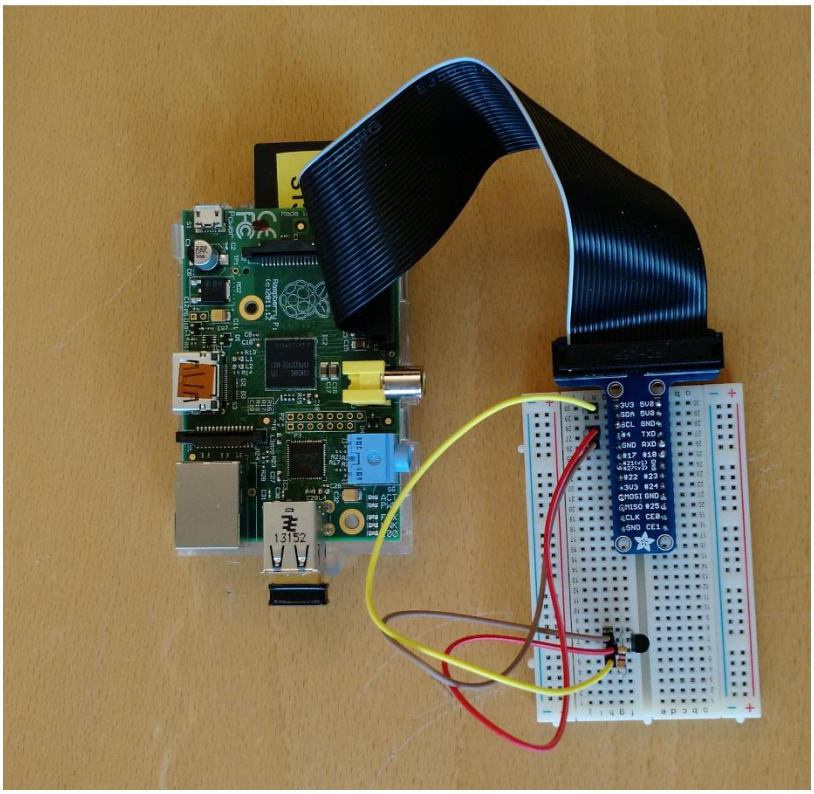


Figure 7 Overview of Raspberry Pi and attached sensor

To read the sensor data of the thermometer a small python program, named Temp.py was created to collect and return the value of the sensor. The sensor itself converts the readings into temperature values every 750ms (max) [23].

5.3 Communication

To get the Pis to communicate with each other over the WiFi, we had the option of making an ad-hoc network (decentralized network without managed infrastructure) or configure one of the Pis as an access point (Network with a centralized managing node in the form of an access point). Both methods were tested and the ad-hoc option was proven to be too unreliable. Therefore, the latter option was used. To make a Raspberry pi function as an access point the following steps has to be made.

- Install and configure wireless access point daemon (hostapd used)
- Install and configure a Dynamic Host Configuration Protocol (DHCP) server

This will allow the Pi to announce an SSID and devices to connect to its network.

5.4 DHCP

DHCP is a standardized network protocol used to dynamically distribute network communication configuration to hosts connecting to the network. This enables the connecting host to automatically enroll in the network with the correct IP-address. The control Pi has isc-dhcp-server installed and configured to fulfill this purpose.

5.5 Network Time Protocol

Network time protocol (NTP) [24] is a protocol designed to synchronize the local clocks on devices. By installing and configuring NTP on a device it is possible to poll a NTP-server and receive back clock synchronization data with the correct time. A Raspberry pi does not have a real time clock in its kit and therefore NTP is used to keep the clocks of the as synchronized as possible. Since the devices should not need an internet connection to work correctly, the control pi is configured as a NTP server which updates the clients with its own local system time. However, while the network is under heavy load and due to the inaccuracy of the Raspberry Pis internal software clock, the NTP-clients get a large off-set in time compared to the server. A decision was therefore made to run the NTP over a wired network during our tests to keep the devices synchronized.

5.6 Programs

Due to low complexity and high availability of various libraries and support communities Python was chosen as our programming language. To work on top of the CSMA/CA based WiFi, three programs were created to showcase aspects of time- and event-based communication scenarios. These programs are used to alter the way the sensor data is sent to the control node. In the first two programs the data will be sent in a purely time-triggered or event-triggered fashion. The third program will showcase a hybrid of the two and the benefits this combination brings to the performance of the system. The following sections explain each program created.

5.6.2 Time-triggered

This program uses a time-triggered scheme to send sensor data over the network, i.e. measured temperature data is sent to the controller every second. The sensor node's code uses TEMP.py to collect the sensor data and then sends it over a UDP-socket to the control node. To keep track of the sensor data over the network, a time-stamp (using datetime module) and sequence number is sent with each packet. When data is received at the control node the values are stripped and examined. The sensor data is used to trigger an alarm if the temperature exceeds a certain value, the sequence number is used to indicate packet loss and the timestamp to calculate the delay over the medium. Each LED-light connected to the control pi corresponds to an event-triggered by the received data. The green light is on as long as the control node receives data at a certain interval. The yellow light blinks when the sequence number indicates packet loss and the red light turns on when the temperature is at an alarm value level.

5.6.3 Event-triggered

Much like the program described above, this program uses UDP-socket code to send its sensor data, packets have time-stamps and sequence numbers. The main difference in in the way the data is sent. This program sends collected sensor readings only when a certain threshold value is reached, i.e., when reading the Temp.py return value. This means that the node is completely idle on the network until an event triggers the node to send its data. The control node strips the data and reacts to the alarm by lighting the red LED-light. If the sequence number does not match, the yellow light flashes. However, the control node has no way of knowing if the sensor node is still alive during its idle state or if an alarm message has been lost in transit, until the next alarm happens and the sequence numbers are compared.

5.6.3 Hybrid

This program is a continuation of the event-triggered program described above and incorporates some time-triggered aspects and reliability improvements. While still waiting for the temperature to reach its threshold value before sending the payload, this program incorporates a keep-alive mechanism that enables the control node to know that the sensor is still active. This mechanism is built using the schedule module in python that enables the program to send a keep-alive message every 8 seconds. Another improvement over the event-triggered program is an acknowledgement system between the sensor and control node when sending its data. Given the proper outer stimuli, the sensor node sends its data over the network and keeps sending until the control node sends an acknowledgement message back. At that time the sensor stops sending and keeps checking for new alarm values. On the control node, the LED-lights are used in the following way: green LED-light indicates that the sensor is still alive, red LED-light signals an alarm message, while the yellow light is idle in this program.

5.7 Test environment

Each program has undergone the same three states of environmental conditions during testing. In all three cases the sensor and control nodes are located 10 meter apart from each other.

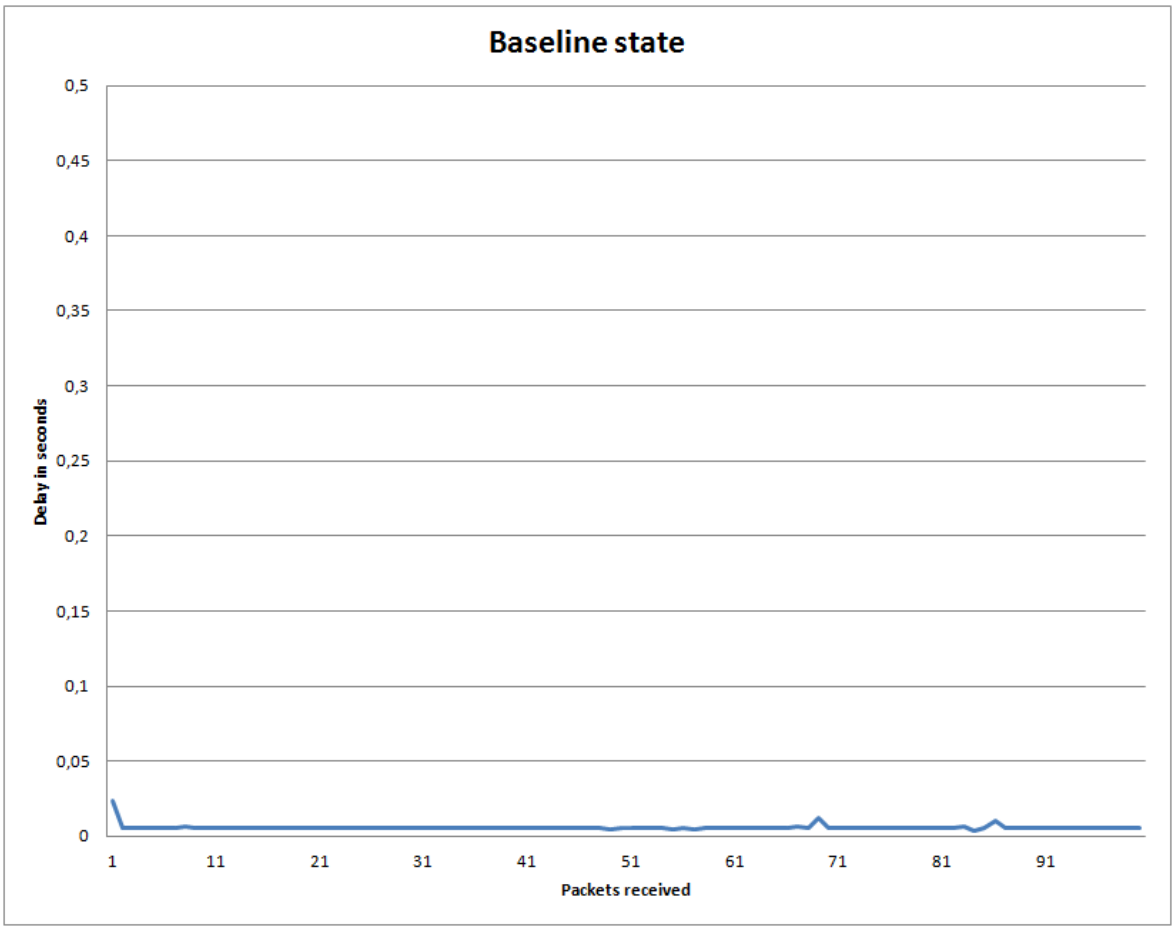
- Baseline state - the sensor node sends its data unhindered.
- Congested state - the sensor node sends its data under heavy traffic load. The congested state is achieved through 4 computers connecting to the control node's network and sending 1gb dummy files between each other. This results in a large amount of traffic going through the media air disrupting the sensor data.
- Congested and interference state - the sensor node sends its data under heavy traffic load and with added interference. This state introduces the added hindrance of a wall and metal container between the sensor and control node. Additionally, there is interference from a device sending a dirty signal on the 2,45 GHz band (close to the operating channel of our network).

These tests are designed to showcase the performance of the programs and to some extent the applicability of the underlying MAC-protocol. Due to the underlying CSMA/CA protocol used in our lab environment the expectation is that the test results will differ quite substantially. Introducing a heavy traffic load on our network should force the carrier sense to initiate the random back-off timer in CSMA often and the access delays should increase thereafter. The added interference should hopefully result in some packet loss and showcase the benefits and limitation of our different programs used in the tests.

6. RESULTS

Here follow the results of the different tests that the sensor systems have undergone. Each test consists of a 100 packets being sent under the described conditions. During the tests the status of the time synchronization of NTP is continually checked to make sure that the values are as accurate as possible. For each program the results are presented in three graphs representing the different states. The y-axis of all the graphs represents the delay and the x-axis shows how many packets were received on the control node.

6.1 Time-Triggered Program

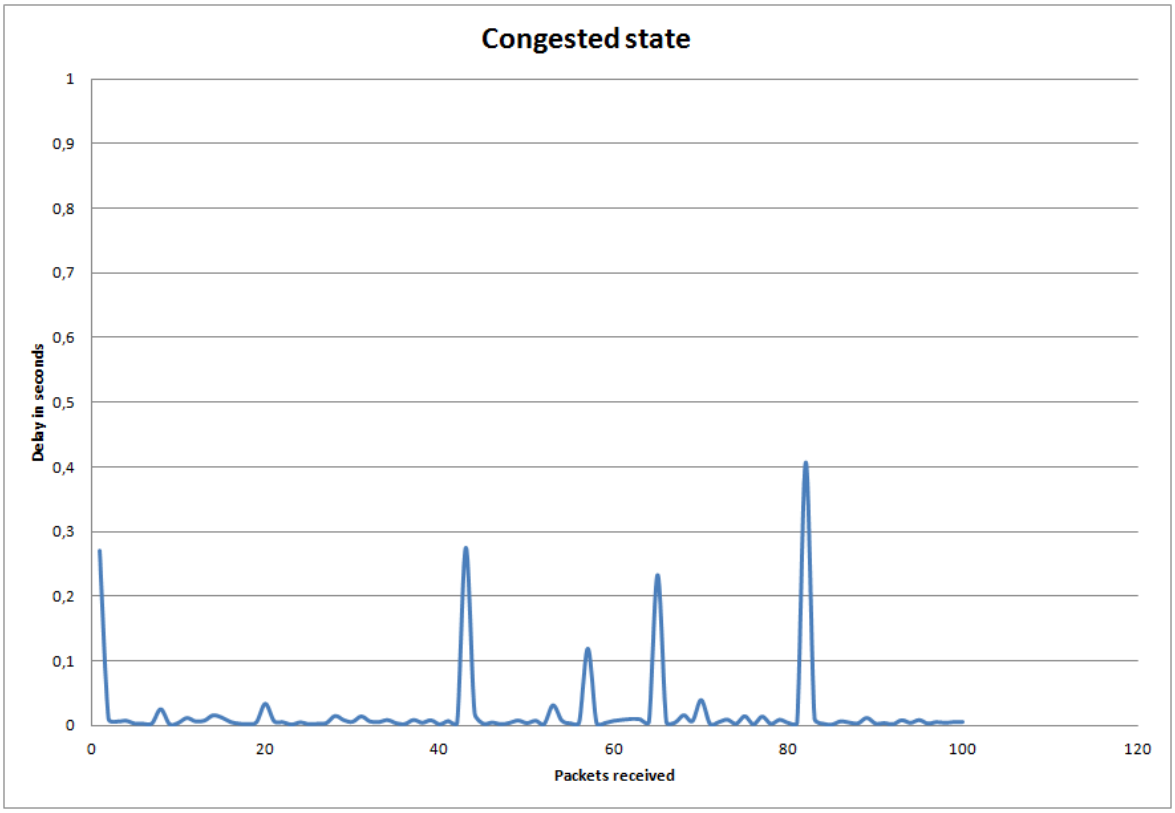


Max delay in seconds	0,02378
Min delay in seconds	0,003439
Average delay in seconds	0,00573474
drop rate per 100 packets	0

Figure 8 Graph representation of baseline state

Table 2 NTP run during base-line state

pi@Raspberrypi	delay/ms	offset/ms	jitter/ms
Max-value	0,921	-0,72	0,162
Min-value	0,897	-1,064	0,06
Average	0,9066667	-0,874333	0,128

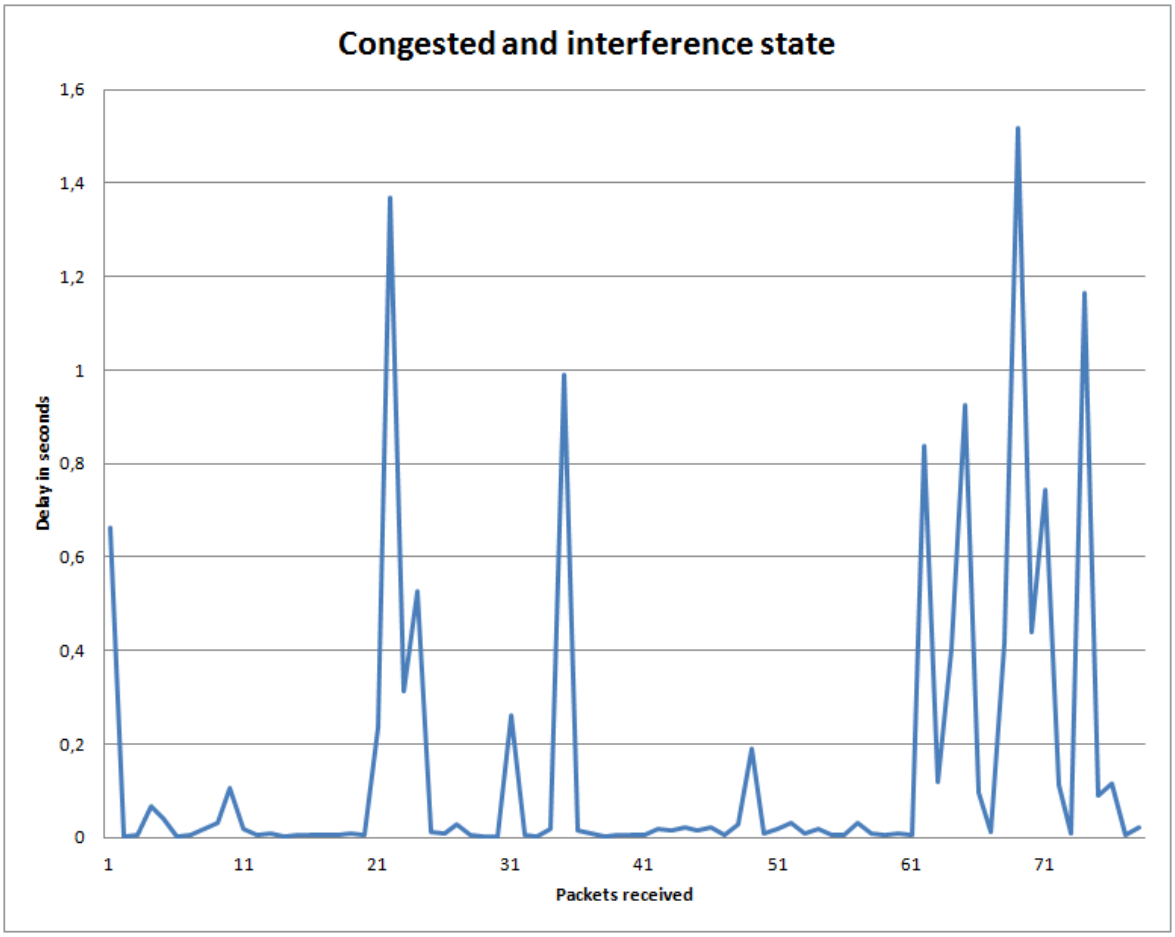


Max delay in seconds	0,40658
Min delay in seconds	0,00152
Average delay in seconds	0,01991
drop rate per 100 packets	0

Figure 9 Graph representation of congested state

Table 3 NTP run during congested state

pi@Raspberrypi	delay/ms	offset/ms	jitter/ms
Max-value	0,916	-0,44	0,24
Min-value	0,916	-0,44	0,24
Average	0,916	-0,44	0,24



Max delay in seconds	1,51939
Min delay in seconds	0,001739
Average delay in seconds	0,158547
drop rate per 100 packets	22

Figure 10 Graph representation of congested and interference state

Table 4 NTP run during congested and interference state

pi@Raspberrypi	delay/ms	offset/ms	jitter/ms
Max-value	0,862	-0,78	0,413
Min-value	0,862	-0,78	0,389
Average	0,862	-0,78	0,401

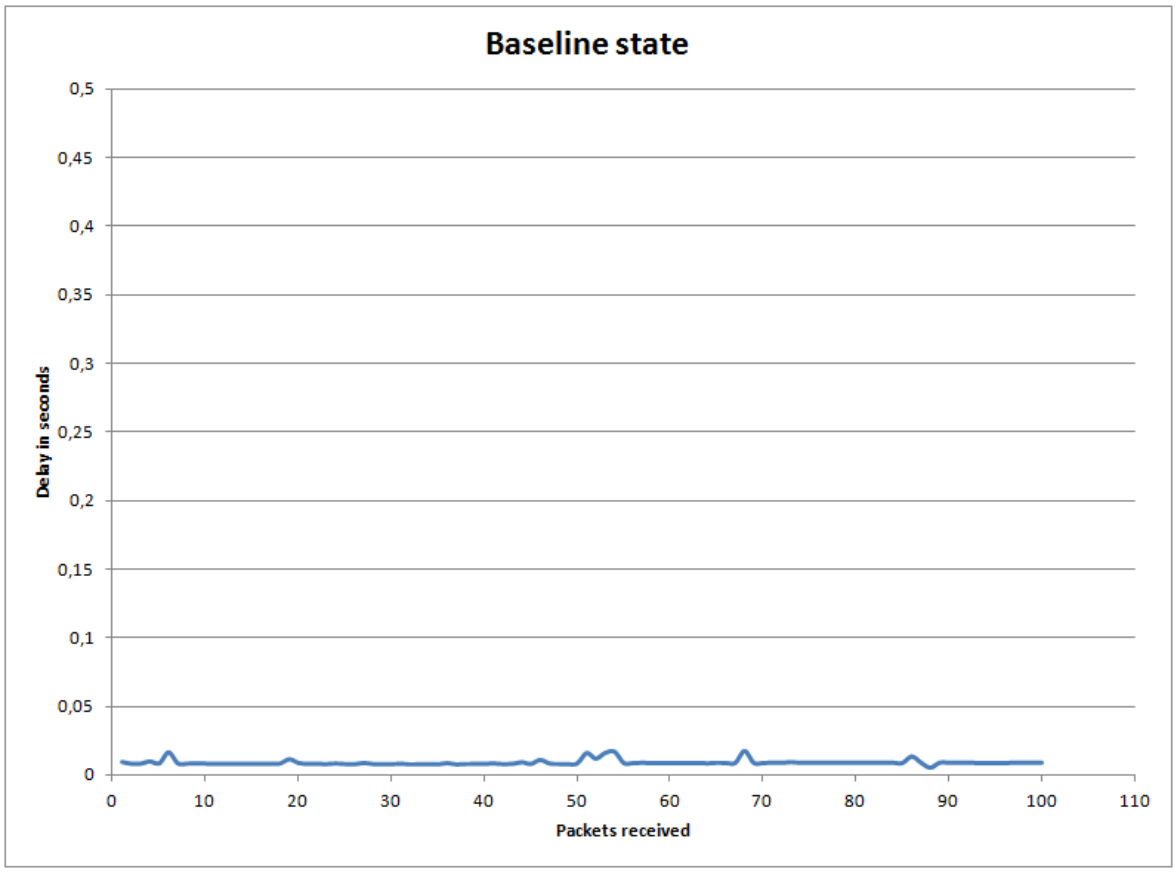
6.1.1 Analysis of time-triggered runs

The Baseline state (Figure 8) shows a steady end-to-end delay over the network with a 5,7 milliseconds an average NTP off-set of 0,9 milliseconds (Table 2). These results are expected due to the little to no other traffic in the network, which allows fast access to the medium through CSMA/CA. As expected, no packets were dropped, but the inherent non deterministic nature of the CSMA/CA protocols has given some off values that differ in delay, e.g., the values shown in Figure 8. These could be accounted to the back-off mechanism of CSMA/CA.

When the network was introduced to a heavy traffic load the results of the tests changed, as showed in Figure 9. The congested state shows a large increase in delay with an average of 19,9 millisecond delay and an average NTP off-set of -0,44 milliseconds (Table 3). When analyzing the maximum and minimum delays, we can see a big difference between the maximum value of 400 milliseconds and the minimum of 1,5 milliseconds, which is a prime example of the random access nature of CSMA/CA as the random back-off timers can delay the access to the medium considerably.

The addition of interference and obstruction of line of sight (LOS) on top of heavy traffic load caused the following changes (Figure 10). The average delay jumped up to 150 milliseconds and packet drop rate to 22 packets out of 100 sent packets. NTP offset during the run was at a steady average of -0,78 ms (Table 4). An average of 160 milliseconds with a minimum value of 1,7 ms and a maximum of 1,5 seconds demonstrates again the back-off algorithm of CSMA/CA. The packet drop rate highlights the issue of this architecture of not being designed for reliability i.e, not suitable in time constrained and safety critical environments. Every packet lost in this system could be an alarm triggering packet and there is no way of confirming that the critical packed arrived or got lost in transit.

6.2 Event-triggered program

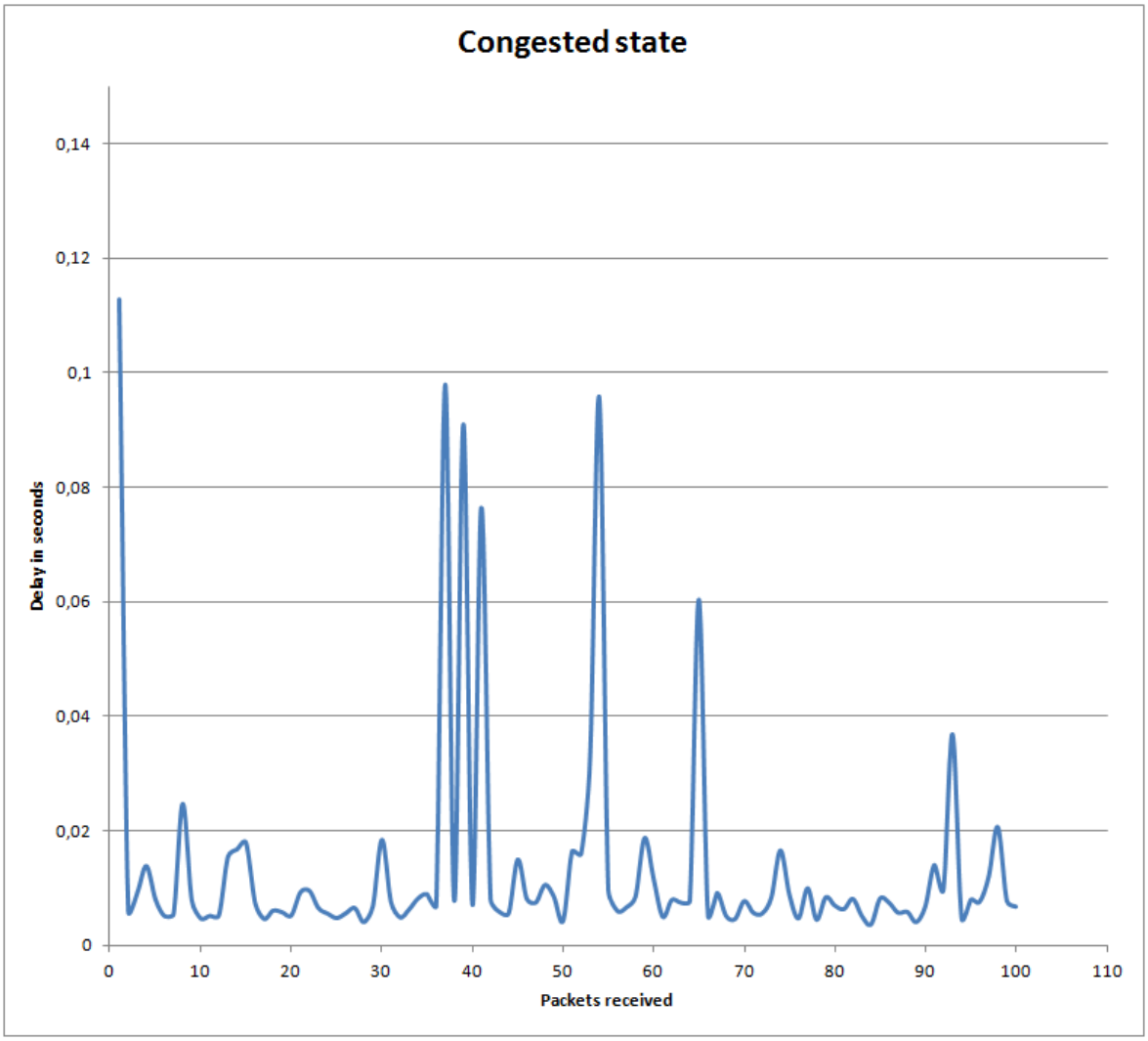


Max delay in seconds	0,0175
Min delay in seconds	0,00494
Average delay in seconds	0,00881
drop rate per 100 packets	0

Figure 11 Graph representation of baseline state + event-triggered program

Table 5 NTP run during base-line state + event-triggered program

pi@Raspberrypi	delay	offset	jitter
Max value	1,022	0,299	0,38
Min value	0,833	-0,824	0,04
Average	0,897132	-0,18303	0,114

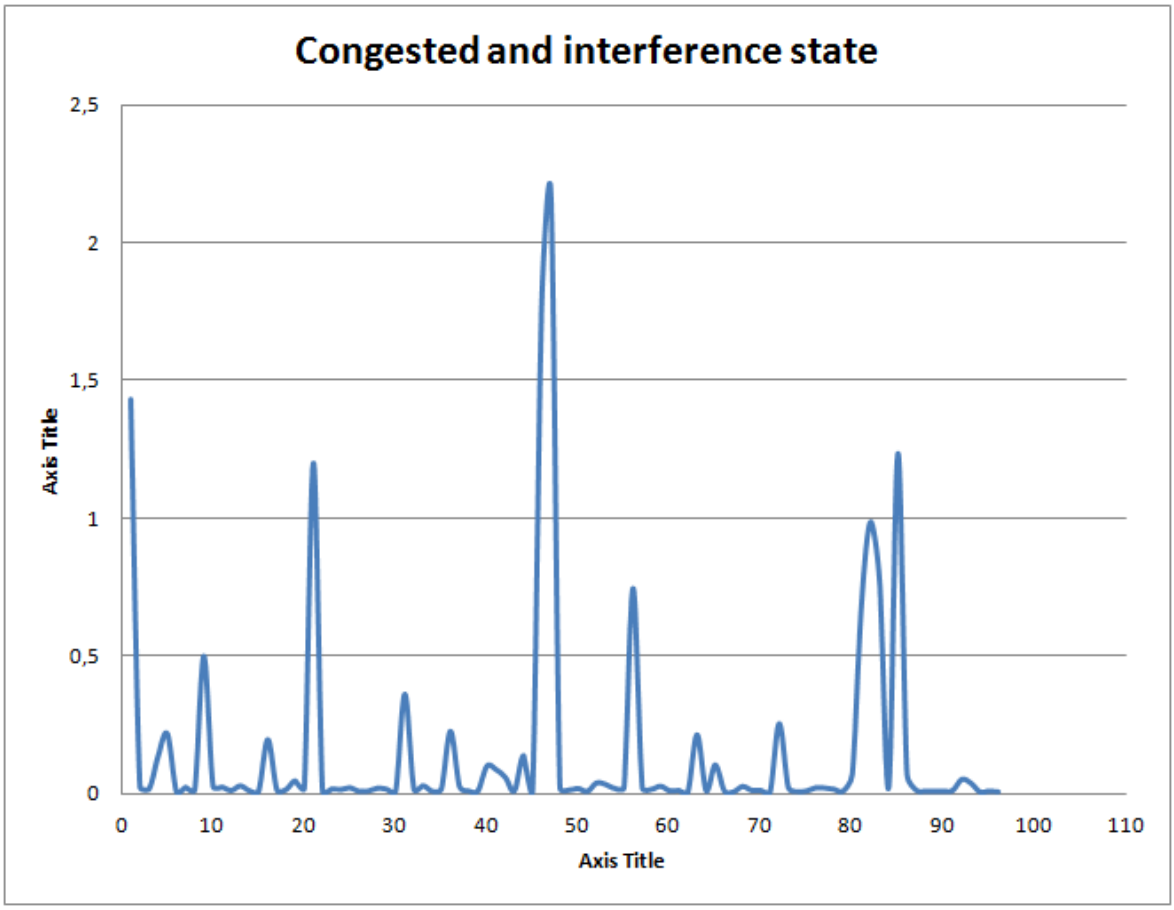


Max delay in seconds	0,11268
Min delay in seconds	0,00378
Average delay in seconds	0,01382
drop rate per 100 packets	0

Figure 12 Graph representation of congested state + event-triggered program

Table 6 NTP run during congested state + event-triggered program

pi@Raspberrypi	delay/ms	offset/ms	jitter/ms
Max value	1,793	-0,042	0,537
Min value	0,874	-0,637	0,157
Average	1,144133	-0,32447	0,403267



Max delay in seconds	2,17434
Min delay in seconds	0,00421
Average delay in seconds	0,15398
drop rate per 100 packets	4

Figure 13 Graph representation of congested and interference state + event-triggered program

Table 7 NTP run during congested and interference state + event-triggered program

pi@Raspberrypi	delay	offset	jitter
Max value	1,793	-0,042	0,467
Min value	1,016	-0,45	0,058
Average	1,721889	-0,32539	0,168111

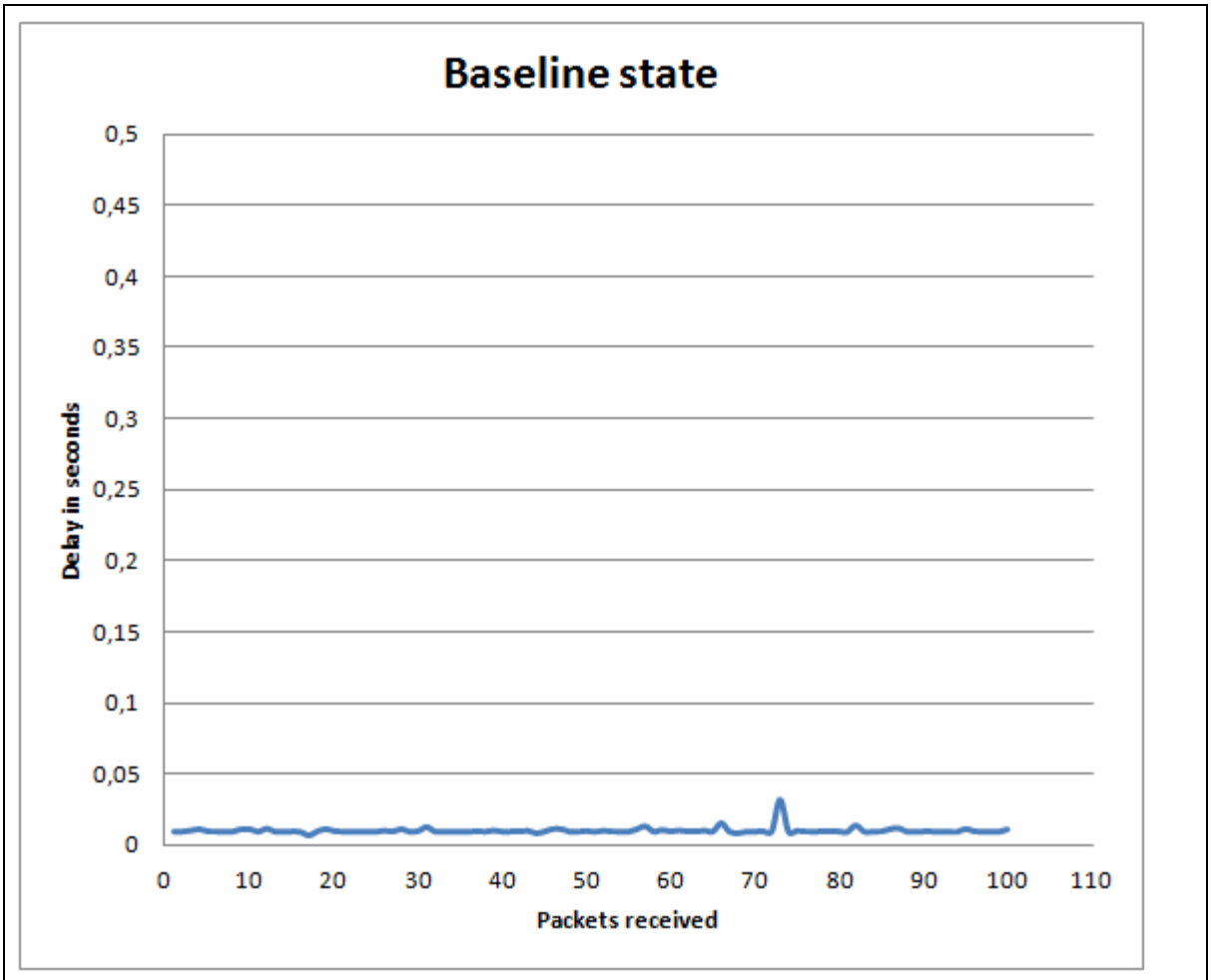
6.2.1 Analysis of event-triggered runs

The baseline in the event-triggered run, as expected, shows (Figure 11) a relative steady values of delay with an average value of 8.8 ms, the minimum of 4,9 ms and the maximum of 17,5 ms. Like the time-triggered baseline this run shows also some off values that could be accounted to the back-off mechanism. NTP offset (Table 5) during the run vas at an average of -0,18 ms. Like in the time-triggered run, these values were expected as the channel medium is not under any traffic load or duress.

Introducing traffic load caused an increase in the delay as the sensor node now has to contend for channel access, as demonstrated in Figure 12. The average delay increased to 13,8 ms with the minimum value of 3,8 ms and the maximum of 113 ms. As mentioned in the TT-congested state, this corresponds to CSMA/CA protocol behavior. The NTP shows an average offset of -0,32447ms (Table 6).

The adding interference created a packet loss of 4% (Figure 13). The average delay was at a 154 ms with the maximum value of 2,17 seconds and the minimum of 4,2 ms. NTP offset during the test was at -0.33 ms (Table 7). Each packet sent within the system contains an alarm temperature value and therefore every dropped packet is an alarm gone without notifying the control node. Possible reason for the low drop rate compared to TT congested with interference state, is that the sensor node only has to deal with sending one packet at different times i.e., less contention for channel access and internal buffer queues in the WiFi network card of the sensor nodes.

6.3 Hybrid program

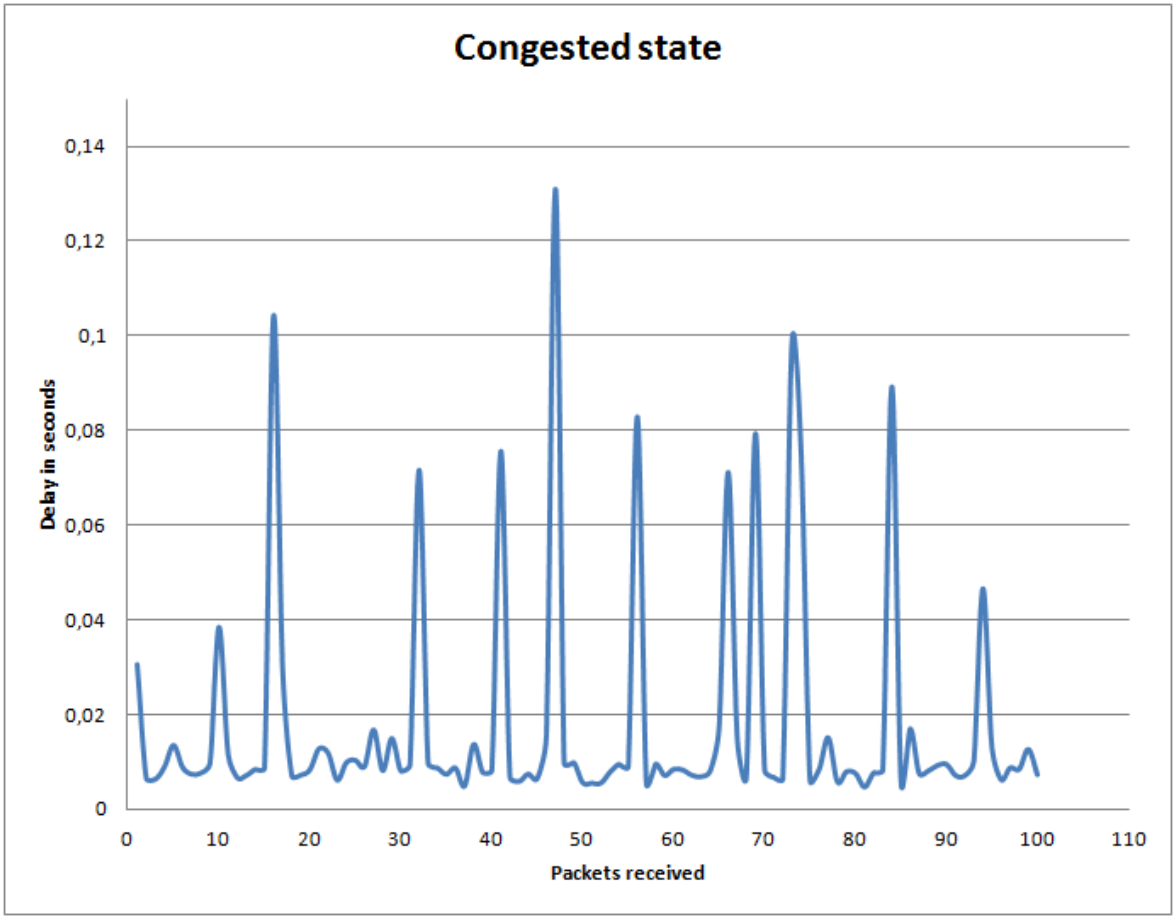


Max	0,031658
Min	0,006346
Average	0,0096754
drop rate	0

Figure 14 Graph representation of base-line state + hybrid program

Table 8 NTP run during base-line state + hybrid program

pi@Raspberrypi	delay	offset	jitter
Max value	0,936	0,306	0,317
Min value	0,867	0,112	0,096
Average	0,9158	0,2186	0,2554

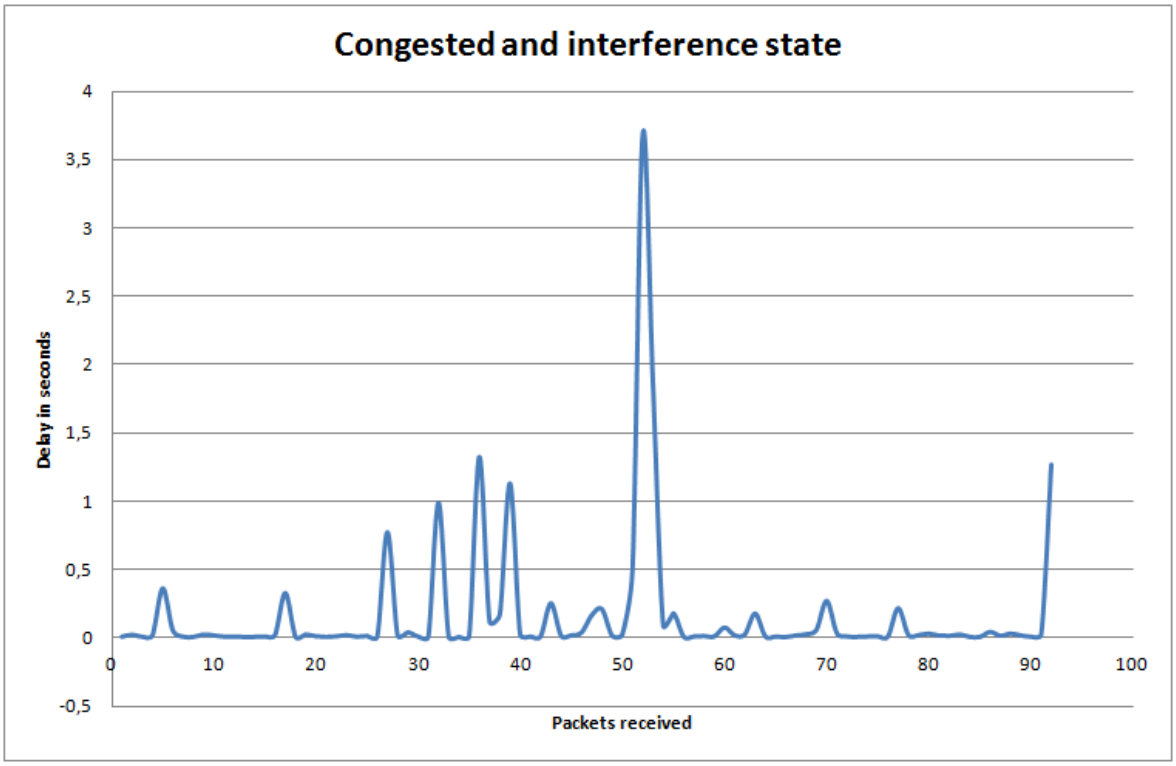


Max	0,131024
Min	0,00475
Average	0,0178558
drop rate	0

Figure 15 Graph representation of congested state + hybrid program

Table 9 NTP run during congested state + hybrid program

pi@Raspberrypi	delay	offset	jitter
Max value	1,867	0,424	0,432
Min value	0,997	0,036	0,155
Average	1,457455	0,259455	0,243636



Max	3,692763
Min	0,006267
Average	0,1673831
*drop rate	8

Figure 16 Graph representation of congested and interference state + hybrid program

Table 10 NTP run during of congested and interference state + hybrid program

pi@Raspberrypi	delay	offset	jitter
Max value	1,052	0,243	0,131
Min value	0,853	-0,069	0,001
Average	0,943053	0,126579	0,050895

6.3.1 Analysis of hybrid runs

The hybrid scheme run in baseline scenario shows, in Figure 14, the same characteristics as the other two schemes but with an increase in the average delay of packets transmitted. The average delay in the hybrid run is 9,7ms, the minimum and maximum values are 6,3ms and 31,7ms respectively. The NTP offset (Table 8) average during the run is 0,2ms. The hybrid program adds some more functions to the server node, this could add a delay from that the tagging of the data string to its transmission in the network stack of the WiFi adapter.

When congested, the hybrid run shows (Figure 15) an increase in average delay, compared to its baseline, resulting in the average delay of 17,9ms, the maximum and minimum values of 131ms and 4,8ms. The results conform to the time and event-triggered scenarios run in the same channel conditions. The NTP offset (Table 9) during this state run was at an average of 0,3ms.

The state with added interference and traffic load (Figure 16) measured an average delay of 167ms with the maximum value being 3,7 seconds and the minimum value being 6,3ms. The NTP offset (Table 10) was at an average of 1ms and the packet drop rate was 8 packets per 100 sent. The added mechanics of the keep-alive message and acknowledgement response gives this hybrid system added reliability when it comes to getting the alarm message at the control node. While a packet loss of 8% was detected during testing, every alarm got through to the control node due to its added continuous sending mechanic while no acknowledgement is received. The duration from the event-triggered alarm on the sensor node to the control node receiving this is however delayed by one second each retransmission while the system waits for an acknowledgement message. The longest delay of an end-to-end alarm message in our run was 1,995 seconds, which constitutes one extra transmission of an alarm level value.

7. CONCLUSIONS

The goal of this thesis work was to design a lab assignment for a university level course focused on embedded systems. After finishing the lab assignment, the students should have a greater understanding of the criteria that are crucial for embedded systems used in different applications and also gain greater knowledge about time- and event-triggered systems, timely communication over wireless medium.

This thesis work started with a literature study about different classes of embedded systems, depending on their applications and timing requirements. Next, the work continued with a theoretical overview of the principal communication mechanisms for accessing the wireless media, some core standards and protocols designed for systems that depend on deterministic communication, e.g. ESs with time driven or event driven communication. After that, different possibilities for a lab design were considered and a setup with a sensor node sending its reading to the controller was chosen. To achieve the goals of the course future students are suggested to implement the setup using a temperature sensor and two Raspberry Pis. After that, various possibilities for the communication between the sensor and the controller should be considered by the students.

From the literature study performed in this thesis work, several different communication designs were evaluated, such as: implementing a software based TDMA, manipulating the network stack, or comparing the vendor hardware with different mac-schemes. While these designs would serve its intended purpose they could not be chosen due to cost, hardware and time limitations. Instead, a new solution suggesting implementation of several communication strategies on top of existing CSMA MAC was proposed. The designed lab is a suitable assignment for students wanting to learn about Raspberry Pi as a system, basic Python programming and different applications with timing requirements. Three programs were developed to work over the MAC and alter the communication between the devices. The programs use communication that is either time-triggered, event-triggered or a combination of the two.

During the development of the time-triggered and event-triggered communication strategies students are forced to consider different aspects that are essential for use in embedded systems, such as time-synchronization and reliability, and how these can be incorporated in the used systems. Under the performed tests the network was filled with continuous traffic and therefore the packets sent from both systems were treated equally under the heavy traffic load, but that would not be the case if all traffic on the network was either time-triggered or event-triggered. If used in the same manner as our sensor node, the time-triggered system would send considerably more data over the network than our event-driven one. This would in theory give the event-driven system a higher chance to reach its destination as the network would be less busy overall. On the other hand, a situation when a node gets disconnected from the network can be immediately detected in a system communicating in time-triggered fashion, which is not the case for event-driven communication. The hybrid system should, during the development phase, spark thoughts about providing reliability when designing the system. With its keep-alive mechanic and acknowledgement messages it guaranties that the control node receives an alarm even if it takes several transmissions.

At the same time, test results show the limitations of the CSMA/CA-protocol i.e., CSMA is not a good mechanism for time-triggered or event-driven traffic in embedded systems with strict timing requirements due to the inherent variation in access delay to the medium. The protocol gives no guaranties for the timeliness and makes the system considered unreliable. Further confirmation is given in our literature study where strong evidence is derived from the related research were qualities found in TDMA-based solutions are favorable for embedded systems with time constraints.

REFERENCES

[1] X. Fan, Real-Time Embedded Systems, Oxford: Elsevier Inc., 2015.

[2] H. Kopetz, Real-Time Systems, New York Dordrecht Heidelberg London: Springer, 2011.

[3] M. Ashjaei, "Performance Analysis of Master-Slave Switched Ethernet Network," *Multi-Hop Real-Time Communication over Switched Ethernet Technology*, pp. 1-26, 2013.

[4] scriptoriumdesigns, "Introduction to Embedded programming," [Online]. Available: http://www.scriptoriumdesigns.com/embedded/interrupts.php. [Accessed 10 04 2016].

[5] D. Coleman och D. Westcott, CWNA Certified Wireless Network Administrator, Indianapolis: John Wiley & Sons , 2012.

[6] K. Kunert, ”TDMA-based MAC Protocols for Wireless Sensor Networks: State of the Art and Important Research Issues,” Halmstad Univeristy, Halmstad.

[7] H. Karl och A. Willig, Protocols and Architectures for Wireless Sensor Networks, West Sussex: John Wiley & Sons, 2005.

[8] A. Tanenbaum och D. Wetherall, Computer Networks, United States: Pearson Education Inc, 2011.

[9] E. Uhleman, *Power Point for Embedded systems course,* Västerås: Mälardalen University, 2014.

[10] F. Fitzek, ”Code Division Multiple Access,” Universita di Ferrara, 2003.

[11] M. Nixon, ”A Comparison of WirelessHART™ and ISA100.11a,” Emerson Process Management, 2012.

[12] T. Zhong och M. Zhan , ”Industrial Wireless Communication Protocol WIA-PA and Its Interoperation with Foundation Fieldbus,” International Conference On Computer Design And Appliations , China, 2010.

[13] P. Gutiérrez, E. Uhlemann, W. Steiner och M. Björkman, ”A Wireless MAC Method with Support for Heterogeneous Data Traffic,” Mälardalen University, Västerås.

[14] A. Saifullah, Y. Xu, L. Chenyang och Y. Chen, ”End-to-End Communication Delay Analysis in WirelessHART Networks,” Washington University, St.Louis, 2011.

[15] V. Rajendran, K. Obraczka och J. Garcia-Luna-Aceves, ”Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks,” University of California, Santa Cruz.

[16] J. Åkerberg, M. Gidlund, J. Neander, T. Lennvall och M. Björkman, ”Deterministic Downlink Transmission in WirelessHART Networks enabling Wireless ControlApplications,” Mälardalens Univeristy, Västerås.

[17] G. Patti, G. Alderisi och L. Lo Bello, ”SchedWiFi: An Innovative Approach to support Scheduled Traffic in Ad-hoc Industrial IEEE 802.11networks,” University of Catania, Catania.

[18] G. Nychis, T. Hottelier, Z. Yang, S. Seshan och P. Steenkiste, ”Enabling MAC Protocol Implementations on Software-Defined Radios,” Carnegie Mellon University, Pittsburgh, 2009.

[19] P. Djukic och P. Mohapatra, ”Soft-TDMAC: A Software TDMA-based MAC over Commodity 802.11 hardware,” Carleton University, Ottawa.

[20] L. Hernaandez, Wireless LAN Standards and Applications, Boston: Artech House, 2001.

[21] C. Bisdikian, ”IBM Research Report - An Overview of the Bluetooth Wireless Technology,” Thomas J. Watson Research Center, Yorktown Heights, NY, 2001.

[22] AMC LLC , ”ZigBee White Paper,” AMC, Texas, 2006.

[23] r-pi, ”r.pi.se,” [Online]. Available: http://www.r-pi.se/labb-och-experiment/sensorer/digital-thermometer-ds18s20.html. [Använd 25 05 2016].

[24] NTP.org, ”NTP: The Network Time Protocol,” [Online]. Available: www.ntp.org. [Använd 25 05 2016].

APPENDIX A

Consists of configurations and code created during the thesis work. Submitted in a separate document.