

Delft University of Technology
Master's Thesis in Embedded Systems

A Robust Setpoint Based Heartbeat Solution for Unreliable IEEE 802.15.4 WSNs

Alain Noels



A Robust Setpoint Based Heartbeat Solution for Unreliable IEEE 802.15.4 WSA_Ns

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Alain Noels
a.noels@student.tudelft.nl

19th of November 2012

A Robust Setpoint Based Heartbeat Solution for Unreliable IEEE 802.15.4 WSA N s

Author

Alain Noels (a.noels@student.tudelft.nl)

MSc presentation

27th of November 2012

Abstract

Wireless sensor and actuator networks (WSANs) suffer from interference making them unfeasible for actuators that require reliability. This asks for an IEEE 802.15.4 behavior analysis for building automation actuators and a robust WSAN solution. This thesis uses the IEEE 802.15.4 based JenNet communication stack for experiments to define, measure, and create robustness. Failures are classified between soft and hard to identify the impact on the system. Equations are introduced to show the failure probabilities based on packet arrival probabilities. Experiments show the impact of interference on the failure rate with an increased failure rate during office hours, and a ratio between hard and soft failures ranging from 1:5 to 1:25 for single hops depending on the link quality. A setpoint based heartbeat solution is proposed that solves hard failures and copes with soft failures. Equations show the impact of different heartbeat properties on the performance of the heartbeat solution. The solution is implemented and experiments show that it meets the robust properties required by WSANs. To make a WSAN predictable and adaptive to its environment, future implementations could monitor the environment and reconsider timing properties, based on gathered data and hopcount.

Graduation Committee

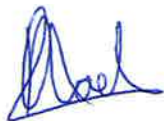
Chair: Prof. dr. K.G. Langendoen, TU Delft
Committee Member: M.A. Zuniga Zamalloa, PhD, TU Delft
Committee Member: H.G. Gross, PhD, TU Delft
Committee Member: N.P. Ray, Priva

Preface

The thesis project started as a quest towards a wireless sensor network stack fitting Priva's requirements, including connectivity with their existing wired network and possible deterministic behavior. Because of a lack of progress due to 6LoWPAN implementation issues, it became clear that an existing stack should be chosen. IEEE 802.15.4 based JenNet was chosen; because Priva's applications mainly use actuators that require reliability and robustness, the focus of the project changed towards wireless robustness.

This project would not have been possible without the aid and assistance of university supervisor prof. dr. Koen Langendoen and daily company supervisor Nick Ray. Both supported the project by placing the dot on the horizon and bringing new insights to face tough issues. Also, the Priva company and its R&D employees provided a lot of support during experiments and prototype development.

Delft, The Netherlands
19th of November 2012



Alain Noels

Contents

| | |
|--|------------|
| Preface | v |
| Contents | vii |
| 1 Introduction | 1 |
| 2 Related work | 3 |
| 2.1 Actuator applications | 3 |
| 2.2 Wired to wireless | 5 |
| 2.3 IEEE 802 wireless standards | 6 |
| 2.4 WSAN requirements | 7 |
| 2.5 Low-power wireless solutions | 9 |
| 2.5.1 DASH7 | 9 |
| 2.5.2 JenNet | 9 |
| 2.5.3 Wibree | 10 |
| 2.5.4 WirelessHART | 10 |
| 2.5.5 ZigBee | 10 |
| 2.5.6 Z-Wave | 11 |
| 2.5.7 Evaluation | 11 |
| 3 Categorization of wireless link failures | 13 |
| 3.1 Classification | 13 |
| 3.2 JenNet properties and limits | 14 |
| 3.3 Hypothesis for failures | 15 |
| 3.4 Failure analysis | 16 |
| 4 Experiments concerning wireless link failures | 21 |
| 4.1 Experiment setup | 21 |
| 4.2 Experiment environment map | 23 |
| 4.3 Office hours vs non-office hours | 23 |
| 4.4 Sun and radiation influences | 24 |
| 4.5 2.4 GHz interference | 25 |
| 4.6 Temperature effects | 27 |
| 4.7 Durations consecutive soft failures | 27 |

| | | |
|----------|--|-----------|
| 4.8 | Ratio soft and hard failures | 27 |
| 5 | Solution for wireless link robustness | 31 |
| 5.1 | Requirements | 31 |
| 5.2 | Existing solutions | 32 |
| 5.3 | Proposed solution | 32 |
| 5.3.1 | State machines | 32 |
| 5.3.2 | Heartbeats | 33 |
| 5.3.3 | Fail-safe mode | 33 |
| 5.3.4 | Recovery mode | 33 |
| 5.3.5 | Key points | 34 |
| 5.4 | Finite state machines | 34 |
| 5.5 | Transition diagrams | 36 |
| 5.6 | Robustness layer interfaces | 36 |
| 5.7 | Sequence diagrams | 38 |
| 5.8 | Heartbeat timing | 41 |
| 6 | Testing the proposed solution | 45 |
| 6.1 | Experiment setup | 45 |
| 6.2 | State ratios | 46 |
| 6.3 | State transitions | 47 |
| 6.4 | Solving hard failures | 47 |
| 6.5 | Hard failure durations | 49 |
| 7 | Conclusions and future work | 51 |
| 7.1 | Conclusions | 51 |
| 7.2 | Future work | 52 |
| | Bibliography | 53 |

Chapter 1

Introduction

Building automation mainly controls indoor environments by sensor and actuator systems. These systems include common known appliances like heating, ventilation, and lighting, which are currently connected by wires, and therefore require skilful installation technicians to place them. A wireless solution would save the costs of special technicians and the wire itself, as well as exploring the renovating possibilities.

Wireless networking techniques like Wi-Fi and Bluetooth are used daily by many people to connect their laptop or mobile phone to other devices. These users know that these devices deplete their battery in one day because of their high power consumption. The standard for low-power wireless personal area networks (802.15.4) solves this issue by standardizing techniques for low power and low data rate networks. Wireless sensor networks (WSNs) use this standard to gather sensor data.

Wireless sensor and actuator networks (WSANs) could be seen similar to WSNs, and could therefore be implemented in the same way. However, failures in WSANs are far more critical, because dangerous situations could occur when actuators malfunction. So, the major issue and concern of this thesis is how to maintain reliability when controlling the actuators wirelessly, as packets may be dropped at random. This leads to this thesis project in which a solution should be found and developed to fit the robustness requirements for the wireless actuator control applications. The solution should maintain compatibility throughout the existing wired network.

The solution should make it possible for nodes to communicate wirelessly, while having a reliable transmission, guaranteeing correct processing of for instance a new actuator setpoint. Reliable communication is not only a major issue when controlling actuators, but also when reconfiguring nodes remotely [5]. Another critical issue is the prevention of an unsynchronized situation between controller and actuator. Actuators are normally not battery powered making them less demanding with respect to energy constraints, although there is an ongoing research into battery powered or energy harvested actuators.

Many protocol stacks that focus on WSNs have been developed, each suiting dif-

ferent applications and environments. These existing WSN stacks and their design decisions will be investigated. A possible WSN solution could be an enhanced or hybrid combination of existing WSN solutions. If the wireless stack contains TCP, another solution could be a modified wireless TCP protocol [23, 25]. Another option is a heartbeat implementation, to ensure an alive receiver and usable medium.

The following main research questions of the thesis project are the starting point for the research part of the project:

How to make actuator applications wireless while maintaining reliability?

1. What are the general WSN requirements?
2. What are the wireless link failure properties?
3. How to make WSNs robust?

The WSN requirements are based on actuator applications at the Priva company, which initiated this thesis project. Several discussions led to requirements involving robustness, energy, bandwidth, processing, and memory capacity. The failure properties are firstly analyzed by the use of analytical models and secondly by carrying out experiments. These experiments compare failures with interference data, and analyze the link quality in relation to time. The robustness solution has been made by the design of state machines, and solution parameters have been tuned by the use of solution models. The solution has been implemented and been tested by experiments.

Related work is illustrated by the introduction of actuator applications, wireless properties, and the WSN requirements in Chapter 2. Chapter 3 categorizes wireless link failures of 802.15.4 by the use of models and theory. Experiments from Chapter 4 analyze the wireless link failures by comparing failure data with interference data. Chapter 5 introduces the robustness solution for WSNs by the use of heartbeats. Different models are used to illustrate the workings of the solution. An implemented solution is used in Chapter 6 to test the solution workings. Finally, Chapter 7 summarizes the conclusions and discusses future work.

Chapter 2

Related work

This chapter explains the actuator types and the requirements of typical WSNs. The project has been performed on behalf of the Priva company, which operates in two sectors for which it designs and produces control systems, which are horticulture, and the living and work environment. In both cases Priva delivers systems that sense and control environments, by the use of sensors and actuators. These systems currently are connected by wire to transport energy and data. The step towards wireless has been put in motion by this thesis project and the development of wireless module prototypes. The actuator applications and WSN requirements are based on Priva's products.

2.1 Actuator applications

Different actuator types are used by Priva that could be controlled wirelessly in the future. They differ in their requirement for robustness because of the impact of possible failures on the environment. They also differ in properties like jitter and control deadlines, which mean that wireless requirements not only concerns the risk of catastrophic failures, but also the risk of reduced life and increased maintenance because of long-term imperfect use. The following actuators are used by Priva from which the bold ones are most commonly used:

- **Window position actuators.** They are proportionally controlled to control the window position angle. Figure 2.1 illustrates a setpoint (SP) based actuator example where the open and close lines are controlled by a local controller.
- **Water supply valves.** The water supply is one of the key elements in a greenhouse. Valves control the flow of water to the crops in the greenhouse. The supplied water, which includes nutrients, has to be prepared in advance according to a custom recipe. A fail-safe system prevents the system from draining too much water from the water supply.

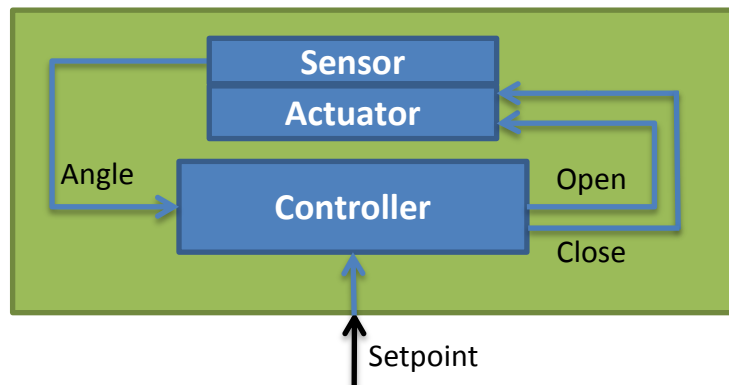


Figure 2.1: Actuator with a SP based dedicated local controller.

- Shading actuators. Sun light penetration into buildings is controlled by shade positioning controllers. Shade position relates to sun light intensity.
- **Relays.** These are used to control electrical actuators. These could be for example lighting installations or simple valves.
- Three way valves. An outgoing air flow changes an environment temperature by adjusting the ratio between the incoming hot and cold air flows. The valves are controlled by a PWM (Pulse-Width Modulation) signal.
- **Pumps.** Liquids and gasses in general are transported by pumps. These could relate to for example heat exchange or crop nutrition transportation.
- **Gas burners.** Proportionally controlled valves adjust the flow of gas towards the gas burner. The gas flow closely relates to the temperature generated by the gas burner. The valves have a slew rate of only a few seconds, which means a resolution of around 100 ms is required.
- **Air conditioners.** Temperature, humidity, and ventilations are the prime conditions controlled by the air conditioners. The ventilation improves air quality by decreasing CO_2 , smoke, dust, CO , O_3 , and moulds.
- Floor heaters. These heaters control the increase of temperature. These actuators are placed in the floor, while most actuators and sensors are located in the ceilings for installation flexibility.
- Lighting. These are controlled by simple relays or SP based controllers. Current lighting products only use SPs for light intensity, while future products could make use of color SPs.

2.2 Wired to wireless

WSANs can not use existing stacks designed for Ethernet because these stacks require too much memory, processing power, and energy. So, WSANs need a dedicated solution. Different algorithms and protocols have been developed for WSANs depending on their applications and environments [20].

Reliable data transmission is crucial for Priva, because their main applications contain the remote control of actuators. Another issue requiring a reliable connection is the need to reconfigure modules remotely. Incorrect or corrupted communication would lead to malfunctioning of the modules, which has to be addressed.

Priva planned to use multiple types of wireless network topologies, which are single/double-hop and multiple-hop networks. The single/double-hop networks could run on the ADRF of Analog Devices, while the multi-hop networks could use JenNet/ZigBee. Figure 2.2 illustrates examples of these networks.

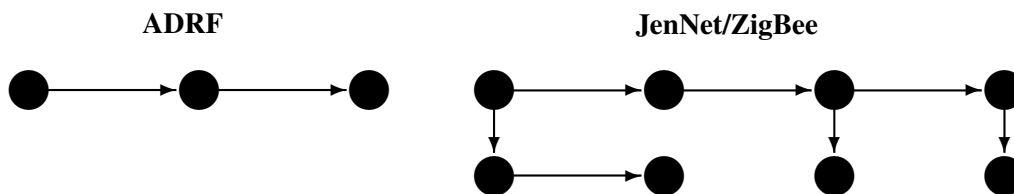


Figure 2.2: Topology examples of future wireless networks used by Priva.

The ADuCRF101 transceiver from Analog Devices is able to use two frequency ranges, 431 MHz to 464 MHz and 862 MHz to 926 MHz. The preferable frequency for horticulture is 433 MHz, making it most resistive to the absorption of energy in the environment. The transceiver uses an ARM Cortex™-M3 32-bit processor. It consumes 12.8 mA when receiving, 24.1 mA when transmitting, and only 1 mA at standby. Power consumption could be 180 nA or 330 nA during sleep depending on which parts remain powered.

Duty cycling is used for battery-powered wireless sensors to conserve energy [9]. Another solution examined at Priva is energy harvesting, which also extends the lifetime of a wireless sensor node. The research on the limits of the combination of these techniques is in progress by students on behalf of Priva at the time of writing.

Priva planned to use the real-time embedded operating system NuttX for its wireless module prototypes. It implements most standard POSIX (Portable Operating System Interface) OS interfaces and it has a non-restrictive BSD (Berkeley Software Distribution) license. The small footprint makes it ideal for microcontroller environments.

Wireless prototypes are developed by Priva, which are to be used when configuring and testing the WSAN communication architecture. The wireless modules use remote procedure calls to control the actuators.

2.3 IEEE 802 wireless standards

The IEEE 802 standards maintained by the IEEE 802 LAN/MAN Standards Committee define the lower two layers of the OSI-model for local area networks (LANs) and metropolitan area networks (MANs).

The 802.11 standard specifies the lower two layers of the OSI-model for wireless local area networks. It defines the basis for Wi-Fi networking products. There have been many amendments over the last years with extensions and enhancements.

The 802.15 standard specifies the lower two layers of the OSI-model for WPANs (Wireless Personal Area Networks). The standard has groups in which group 4 (802.15.4 [14]) specifies low-rate WPANs (LoWPAN). The 802.15.4's features contain carrier sense multiple access with collision avoidance (CSMA-CA), automatic network establishment by the coordinator, power management to ensure low power consumption, and fully handshaked protocols for transfer reliability. Groups 1 (Bluetooth) and 3 (high-rate WPANs) are less interesting because of their increased energy consumption and high data rate. The upper layers could access the MAC-layer directly or use the logical link control (LLC) with the service specific convergence sublayer (SSCS).

The traditional channel frequencies of the 802.15.4 standard are 868 MHz, from 906 to 924 MHz in steps of 2 MHz, and from 2405 to 2480 MHz in 5 MHz steps. On February the 6th 2012 the IEEE Standards Association Board approved the IEEE 802.15.4f [18], which contains the frequency bands 433 MHz and 2.4 GHz. The Analog Devices transceiver targeted by Priva is able to use the channel frequencies from 906 to 924 MHz. However, the frequency of 433 MHz is more preferable for the horticulture because the lower frequencies penetrate better in the application environment. The 2.4 GHz frequency is more suitable for the living and work environment in which room controllers will be used. Further, 2.4 GHz does not penetrate walls well, making it less interfering with other rooms. Data rates of 200 kbps are reachable with 433 MHz. Figure 2.3 illustrates the 802.15.4 architecture.

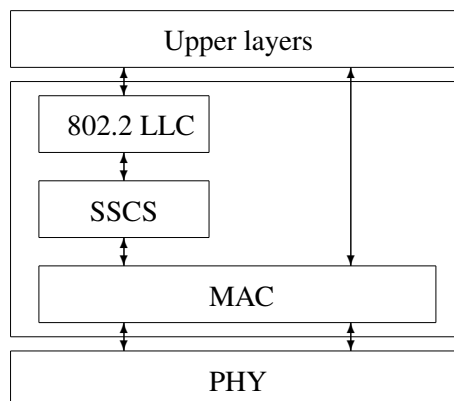


Figure 2.3: The 802.15.4 architecture. [11]

2.4 WSAN requirements

The constraints of Priva's WSAN communication architecture concern: reliability and robustness, energy consumption and power- and duty cycle, performance, memory footprint, multicast, mesh topologies, bandwidth and frame size, security, and network size.

- Reliability and robustness

An ideal medium would always allow a successful data transmission. However, transmission failures occur in reality leading to time-outs that signal errors to the transmitting application. These errors arise after a remote procedure call and a time-out, taking both time and energy.

A possible solution would be some form of heartbeat to let the transmitting side know that the receiver is alive and communication is possible. The transmitter would now be aware of an increased transmission failure probability before a remote call takes place, therefore saving time. A trade-off has to be made between robustness and energy consumption for battery powered devices.

The required reliability by Priva is expressed in a percentage of 99%, which means 99% uptime. The robustness is expressed as mean time between failures (MTBF). The requirement is stated as a MTBF of 10,000, so one hardware failure each 10,000 hours on average. Connection loss does not count as a failure.

- Energy consumption and power- and duty cycle

The sleep time of a device has a large influence on the energy consumption. The energy consumption of a receiving device is often as high as a transmitting one. Different power optimization techniques are available including strobed preamble [6] and low-power listening [13]. The battery powered sensor nodes of Priva's WSAN are supposed to work for 5 years without changing the battery. Different types of periods are planned to be used to report sensor data: 1 each second, 1 each minute, 1 each 5 minutes, and 1 each 10 minutes. Preliminary experiments show an uptime of around 7 ms for each message, this will lead to duty cycles of around 1:140; 1:8,600; 1:42,900; and 1:85,700.

- Network size

The size of a network is defined by the maximum number of nodes supported by the communication architecture. The future WSANs of Priva will contain at most 100 nodes.

- Performance

The performance of the WSANs is defined by the data rate, delivery percentage, and the duty cycle. These are all closely related to the other constraints.

Listing 2.1: A JSON example.

```
{ "TM" = "19.3"; "RH" = "32.9"; "CO" = "0.39"; "PR" = "1.325";  
  "FL" = "18.4"; "ID" = "64" }
```

The WSANs of Priva will contain up to 100 nodes, that each could have a maximum report frequency of 1 each second. This leads to 360,000 messages per hour, which are needed to be handled within the WSAN.

- Memory footprint

The memory footprint determines the amount of memory being used while running. It should be small enough to fit the dimensions of the wireless sensor nodes. Priva prepared to use 32 KB of memory for the code.

- Multicast

Priva requires support for multicast, because it needs to be able to request data from all the nodes simultaneously. Support for a reliable multicast introduces new challenges. This leads to an extra constraint.

- Mesh topologies

Support for mesh topology has advantages due to its ability to route around blocked and broken paths. The downside of it is the increased complexity of the protocol stack. Priva prefers a tree-like network, and does not require support for mesh networking.

- Bandwidth and frame size

Low-power radio links used in WSANs have small bandwidth and frame sizes, which could make it require support for fragmentation. A scenario of Priva's WSAN includes the use of messages containing up to 5 floating-point numbers and a sensor ID, which both are in JSON (JavaScript Object Notation) ASCII-format. Listing 2.1 illustrates a JSON example. It is assumed that on average 4 characters are needed for values and 2 for labels, which all add up to 73 characters in total. The total size is 64 or 73 bytes depending on the use of 7 or 8 bit characters. The payload could have sizes up to 96 bytes. A rate of 360,000 messages per hour of 73 bytes each concludes a maximum data rate of 210 megabit per hour and on average 58.4 kbps through the whole network. Multiple data sinks are used, so typical wireless links use much less than 58.4 kbps.

- Security

Protocol stacks use various techniques to improve security. The AES-128 encryption technique is most commonly used by WSAN protocol stacks for security. Authentication is possible by the use of a PMK (Pairwise Master Key) and PTK (Pairwise Transient Key), to increase security. Priva requires security, but allows it to be at the bottom of the stack or at application level.

2.5 Low-power wireless solutions

Different wireless solutions have been studied as a preliminary research for this thesis project, to find a suitable WSN for experiments and to learn from previous design decisions. These include software solutions like Chameleon [7], ESRT [21], and PSFQ [24]. The software stack to be used during experiments was originally μ IP [10] operating on NuttX OS, but 6LoWPAN [8, 19, 22] implementation issues forced the usage of JenNet as the experiment environment. The following sections introduce JenNet and the other wireless solutions considered during the initial phase of the project.

2.5.1 DASH7

DASH7 is an open source standard for WSNs, which could cover multiple kilometers and is able to run for multiple years on battery. The standard is maintained by the non-profit DASH7 Alliance. It uses the 433 MHz frequency (ISO/IEC 18000-7:2009-standard) that is available world-wide and allows to penetrate concrete and water. It supports IPv6 and also uses AES-128 encryption for security. The maximum number of hops, which is 2 is small compared to other solutions. The data rate (28 to 200 kbps), code size (20 KB) and frame size (256 bytes) fit within the requirements. Multicast is also supported, so DASH7 is a possible candidate because it fits the Priva requirements.

2.5.2 JenNet

JenNet targets ultra-low-power networking using an enhanced 6LoWPAN network layer. It is possible to access 500 nodes and its main applications are residential and industrial. JenNet uses a tree-configuration consisting out of a coordinator, routers and end-nodes. AES-128 encryption is used to ensure security. The edge-nodes are the only devices that could be battery powered because the other devices need to be powered permanently. Figure 2.4 illustrates a JenNet network configuration. The networks are highly robust and use the 'mesh-under' routing scheme from 6LoWPAN to optimize routing for low-power wireless links. Networks can heal from broken connections by reconfiguring routes as illustrated in Figure 2.4b and Figure 2.4c. The required memory depends on the size of the routing table and the type of node. An end-node only requires 16 KB, while the required size for the routers and the coordinator vary from 22 KB to 80 KB, depending on the routing table. A JenNet frame has a maximum of 127 bytes from which 116 bytes can be used as data from end node to the coordinator. Only 108 bytes can be used when sending from the coordinator to an end-node, because of communication overhead. The data rate of JenNet could be up to 250 kbps and multicast is supported. Most properties fit the Priva needs, but the code size of the coordinator and router do not because they could exceed the 32 KB code size limit. JenNet is also considered to be a candidate because it nearly fits all Priva requirements.

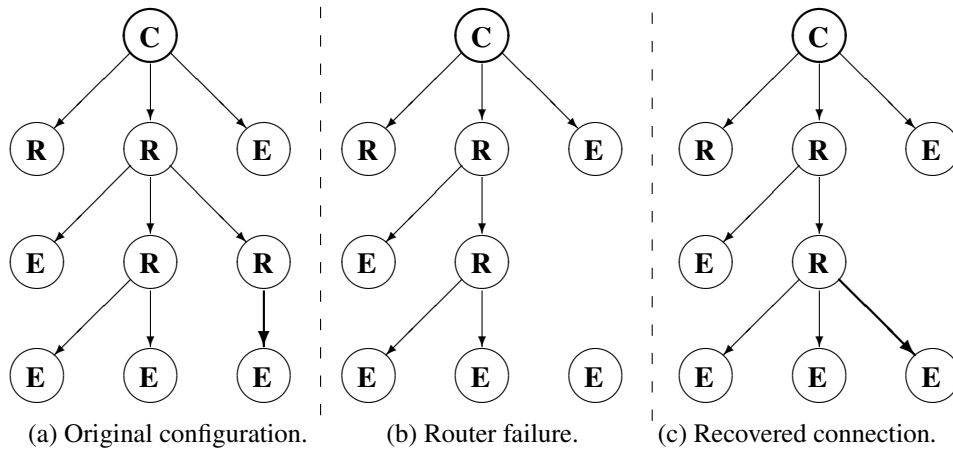


Figure 2.4: JenNet network recovery example.

2.5.3 Wibree

The Wibree [16] technology is closely related to Bluetooth and is mainly designed for watches and body sensors. It has the communication speed of Bluetooth, which is around 1 Mbps. The security is managed by using the AES-128 encryption. The downsides are its inability to use multicast and the small frame size of 31 bytes. Also the code size is too large to fit within the 32 KB. So, Wibree is not able to fit Priva's needs.

2.5.4 WirelessHART

The WirelessHART [17] standard is based on 802.15.4 and it uses the 2.4 GHz band to cover up to 250 meters line-of-sight. The application environment is mainly process plants and it focuses on reliability, security and effective power management. The data rate could be up to 250 kbps. Multicast and mesh-networking are supported. The code is small enough to fit within the required 32 KB. The frame used by 802.15.4 is 127 bytes, so the frames of WirelessHART are smaller than these 127 bytes. The header size is not expected to be larger than 31 bytes, so the frame size would fit the 96 bytes used by Priva. WirelessHART also uses AES-128 encryption, so it fits all the requirements of Priva. WirelessHART is also considered to be a candidate.

2.5.5 ZigBee

ZigBee [12] works on top of the layers defined by 802.15.4. It is used in industry as a standard for low data rate, low power, long battery life and secure networking. Its applications include light and access control for building automation. ZigBee has some similarities to Bluetooth while ZigBee supports multi-hops and Bluetooth does not. It supports multicast, CSMA-CA, mesh-networking, checksums and ac-

knowledgments, which increases reliability. Security is provided by the use of AES-128 encryption. ZigBee uses the same node types as JenNet: a coordinator, routers, and end devices. The data rate varies between 20 and 250 kbps and the code size between 32 and 100 KB, depending on the device type. It has frames of at least 111 bytes, because it also uses the 802.15.4 frames (127 bytes) and ZigBee has 8 to 16 bytes as header overhead. ZigBee fits all the requirements as long as the code size does not rise above the required 32 KB. ZigBee is also considered to be a candidate.

2.5.6 Z-Wave

Z-Wave focuses on home automation applications and competes directly with the ZigBee standard. It uses mesh networking and acknowledgments to confirm automatic or manual commands. It also monitors the present status of each device. The focus does not seem to be at energy consumption minimization, while it should be for WSNs. It focuses more on security by the use of the AES-128 encryption. The data rate varies between 9.6 and 200 kbps, the code size fits the 32 KB limit and multicast is supported. The only downside is the frame size, which is 54 bytes and does not fit the required 96 bytes. So, Z-Wave is not usable at Priva because of this size.

2.5.7 Evaluation

Table 2.1 provides an overview of the wireless solutions. There are many similarities, which are mainly because of the 802.15.4 standard.

Table 2.1: WSN protocol stacks overview.

| | data rate | memory | multicast | frame | security |
|--------------|------------------|---------------|-----------|------------------|----------|
| Priva REQ | ≥ 58.4 kbps | ≤ 32 KB | + | ≥ 96 bytes | + |
| DASH7 | 28-200 kbps | < 20 KB | + | 256 bytes | AES-128 |
| JenNet | ≤ 250 kbps | 16-80 KB | + | 115 bytes | AES-128 |
| Wibree | 1 Mbps | ≤ 128 KB | - | 31 bytes | AES-128 |
| WirelessHART | 20-250 kbps | ≤ 32 KB | + | < 127 bytes | AES-128 |
| ZigBee | 20-250 kbps | ≥ 32 KB | + | ≥ 111 bytes | AES-128 |
| Z-Wave | 9.6-200 kbps | ≤ 32 KB | + | 54 bytes | AES-128 |

The overview shows that some protocol stacks do not fit the requirements. These unsuited protocol stacks are Wibree and Z-Wave, because of their frame sizes. The protocol stacks that seem to be suited are DASH7, JenNet, WirelessHART,

and ZigBee, because they all fit the required properties. However, the DASH7 protocol stack development is still in an early stage at the time of writing. Priva tried to involve in this protocol stack in the past without success. For now, these experiences make the DASH7 protocol stack unusable for Priva.

JenNet has been chosen out of the last three (JenNet, wirelessHART, and ZigBee), because it allows a fast deployment by the use of a IEEE 802.15.4 JenNet evaluation kit. This kit provides several nodes and software development tools, which allow to get usable experiment data fast.

Chapter 3

Categorization of wireless link failures

The analysis phase of the thesis project is explained in this chapter. It first classifies the failure types, followed by an illustration of the environment and the failure properties. Finally, it introduces equations to calculate failure probabilities based on the packet delivery probability. Notice that the thesis project does not include module failures, and only focuses on link failures.

3.1 Classification

- **Actuator** JenNet module driving an LED (Light-Emitting Diode), simulating an actuator.
- **Transmitter** JenNet module sending SPs to the actuator. The SPs (ON/OFF) are initiated by a laptop during experiments.
- **Actuator status** The real status of the LED at the actuator.
- **Transmitter status** The status of the LED according to the transmitter.
- **Succeeded packet** The transmitter transmitted a packet and did receive an acknowledgement (ACK) from the actuator.
- **Failed packet** The transmitter transmitted a packet and did not receive an ACK from the actuator.
- **Soft failure** The transmitter transmitted a packet and this packet has not been received correctly by the actuator. This includes retries at the physical layer.
- **Hard failure** The transmitter status does not match the actuator status as a result of for instance a failed ACK. This will be shown in the next sections.

3.2 JenNet properties and limits

JenNet allows an easy and quick WSN deployment based on the 802.15.4 standard operating in the 2.4 GHz band. Preliminary outdoor experiments show a maximum communicating distance in best case conditions over 375 meters for the power type used in this thesis. These outdoor experiments are executed in an open field like environment at a height of around 1 meter. Indoor distances vary depending on many environment variables like line-of-sight and interference. Distances between 30 and 60 meters within a typical office environment, without walls between nodes, are mainly observed for the experiments.

JenNet uses a tree architecture consisting of three node types. The central node is called the communicator. It will be the initiating node in the experiments and is connected to a laptop to acquire experiment data. The second node type is called a router and always has one parent node to which it belongs. Routers and coordinators cannot use a sleep mode, so they have to be powered permanently. The last type is called end node, which is not used in the experiments because the only advantageous is the possibility to sleep. Robustness is required for actuators that are externally powered and in that sense do not need sleep mode. Figure 3.1 illustrates a JenNet tree configuration example.

JenNet uses three retransmissions in the physical layer for each hop to increase reliability. The packets are acknowledged on each single hop, while end-to-end ACKs are optional for each packet. The end-to-end ACKs contain all the data from the acknowledged packet, while the single hop ACKs do not.

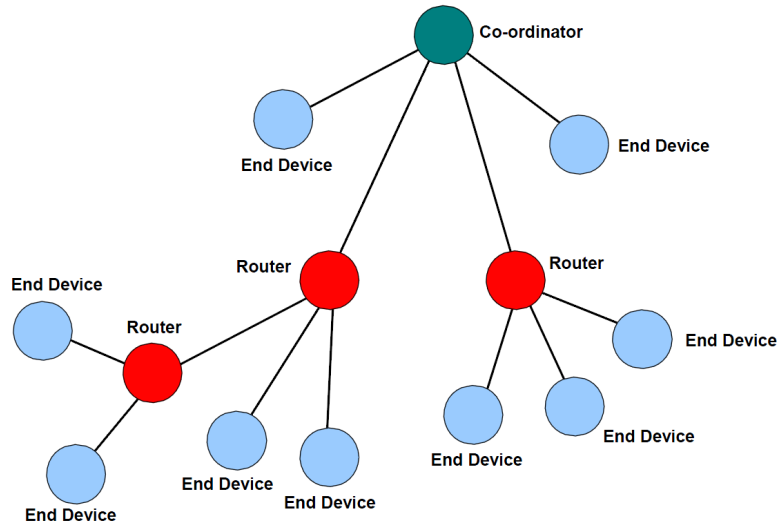


Figure 3.1: JenNet tree configuration example. [1]

3.3 Hypothesis for failures

A Priva WSAW working environment was designed and deployed in order to define, measure, and create robustness. Assume an actuator controlled by the use of a wireless link. Actuators like engines could become dangerous when control would be lost. Loss of control could be the result of one of the following or both:

- Loss of the wireless connection (soft failure).
- Unsynchronized status between actuator and transmitter (hard failure).

Example 1, soft failure. A control system sends a SP wirelessly to an engine to start running. The control system could want to stop the engine to prevent possible damage. Because of a loss of connection, the engine could not be stopped. In this case, the control system assumes that the engine still runs and could not be stopped wirelessly. This assumption corresponds with the reality. The control system will now retry to stop the engine.

Example 2, hard failure. A control system sends a SP wirelessly to an engine to start running. The control system does not receive an ACK, while in reality the packet has been received correctly at the actuator. The transmitter would retransmit the packet, but imagine that the control system decides to cancel retransmissions due to new data inputs. The engine will then keep running, while the control system assumes that the engine never even started to run. This assumption is false. The system will not try to stop the engine, which could lead to dangerous situations.

Figure 3.2 illustrates the situations for successes and failures.

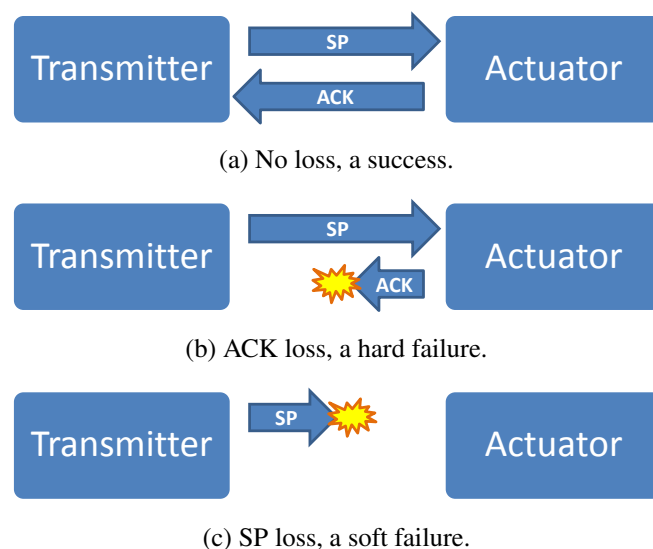


Figure 3.2: End-to-end success and failure types.

Table 3.1 illustrates what could happen when sending a SP from the controller to the actuator. These situations compare the ACK response with the packet success. Success from Table 3.1 is indicated by Figure 3.2a without losses. A hard failure occurs when a SP has been received by the actuator while the ACK has been lost. Figure 3.2b illustrates this, representing the hard failure from Table 3.1. A soft failure occurs when a SP has been lost, illustrated by Figure 3.2c and corresponding with the soft failure from Table 3.1. The models illustrated in Figure 3.2 are based on end-to-end data and end-to-end ACKs.

Table 3.1: Situations when sending a SP.

| | received SP | no SP |
|--------------|--------------|--------------|
| received ACK | Success | - |
| no ACK | Hard failure | Soft failure |

Figure 3.3 illustrates the success and failures when observing the three retransmissions at the lowest layer. It shows that only one success is needed to guarantee a success. A success lets the controller stop retransmitting, showed by grey blocks. A soft failure only occurs when all transmissions fail to send the SP, while at least one failed ACK would conclude a hard failure.

| | First transmission | Retransmission 1 | Retransmission 2 | Retransmission 3 |
|--------------|--------------------|------------------|------------------|------------------|
| Success | Success | | | |
| Success | Failed SP | Failed ACK | Success | |
| Soft failure | Failed SP | Failed SP | Failed SP | Failed SP |
| Hard failure | Failed SP | Failed SP | Failed SP | Failed ACK |
| Hard failure | Failed ACK | Failed SP | Failed ACK | Failed SP |

Figure 3.3: Slots explained.

3.4 Failure analysis

The ratios of successes and failures largely depend on the number of hops between the controller and the actuator. When assuming a single hop situation, which is a common case for a room controller targeted by Priva, only the three retransmissions at the lowest layer have to be taken into account.

Assume packet delivery probability P_{pd} and packet loss probability $1 - P_{pd}$. This holds for data (SPs) and ACKs. Assume data packet loss $f_d : 1 - P_{pd}$. A soft

failure can now be calculated by assuming loss of each data packet. The probability of a single-hop soft failure P_{ss} is shown in equations (3.1).

$$P_{ss} : (f_d)^4 \quad (3.1a)$$

$$P_{ss} : (1 - P_{pd})^4 \quad (3.1b)$$

A hard failure has different cases in which it could occur. Imagine the four transmissions illustrated in Figure 3.3 as four slots. No slot should contain a success ($(P_{pd})^2$), as this would be classified as a successful transmission. Each slot has a failed data packet f_d or a failed ACK ($f_a : P_{pd}(1 - P_{pd})$). There has to be at least one failed ACK in one of the slots, else it would be a soft failure. These rules conclude to the probability of a single-hop hard failure P_{sh} calculated in equations (3.2) in which the probability of a soft failure is subtracted from the probability of four consecutive failed data packets or ACKs.

$$P_{sh} : (f_d + f_a)^4 - (f_d)^4 \quad (3.2a)$$

$$P_{sh} : ((1 - P_{pd}) + P_{pd}(1 - P_{pd}))^4 - (1 - P_{pd})^4 \quad (3.2b)$$

$$P_{sh} : (1 - P_{pd}^2)^4 - (1 - P_{pd})^4 \quad (3.2c)$$

The ratio between single-hop hard and soft failures r_s can be calculated by dividing P_{sh} by P_{ss} , which is illustrated in equations (3.3).

$$r_s : \frac{P_{sh}}{P_{ss}} \quad (3.3a)$$

$$r_s : \frac{(1 - P_{pd}^2)^4 - (1 - P_{pd})^4}{(1 - P_{pd})^4} \quad (3.3b)$$

$$r_s : \left(\frac{1 - P_{pd}^2}{1 - P_{pd}} \right)^4 - 1 \quad (3.3c)$$

The probabilities of single hop failures and successes are illustrated in Figure 3.4. It should be noticed that the packet delivery probability will mostly be near 0% or 100%. This makes hard failures occur less often than soft failures, and this difference will increase when the link quality worsens. The multi-hop situation also differs because of an end-to-end ACK, which behaves like a data packet to acknowledge a successful transmission. So, even multi-hop with only one hop differs from single hop because of the end-to-end ACK.

Assume h to be the number of hops from the controller to the actuator. A soft failure can be modelled as a situation in which at least one single-hop soft failure occurs. The multi-hop soft failure P_{ms} is shown in equations (3.4).

$$P_{ms} : 1 - (1 - P_{ss})^h \quad (3.4a)$$

$$P_{ms} : 1 - (1 - (1 - P_{pd})^4)^h \quad (3.4b)$$

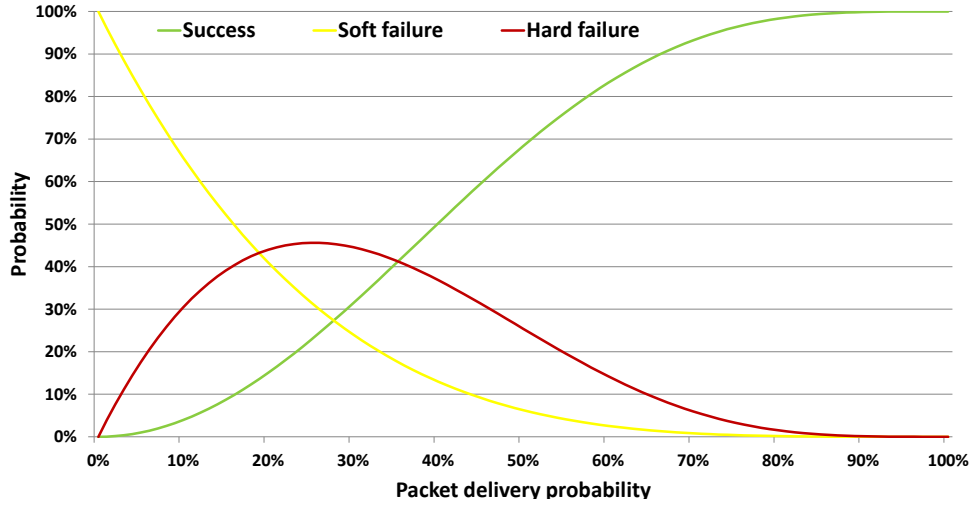


Figure 3.4: Single-hop probabilities of success, soft failure, and hard failure.

A hard failure can be modelled as a situation in which all hops towards the actuator do not contain a single-hop soft failure, while towards the controller at least one single-hop soft failure occurs. The multi-hop hard failure P_{mh} is shown in equations (3.5).

$$P_{mh} : (1 - P_{ss})^h (1 - (1 - P_{ss})^h) \quad (3.5a)$$

$$P_{mh} : (1 - (1 - P_{pd})^4)^h (1 - (1 - (1 - P_{pd})^4)^h) \quad (3.5b)$$

Figures 3.5, 3.6, and 3.7 illustrate the success and failure probabilities when using multi-hop ranging from 1 to 10 hops.

The multi-hop equations in this section assume the multi-hop system to ignore the first single-hop ACK, while waiting for the end-to-end ACK. The equations would be slightly different when it would not ignore this single-hop ACK. This difference would become smaller when the number of hops increases.

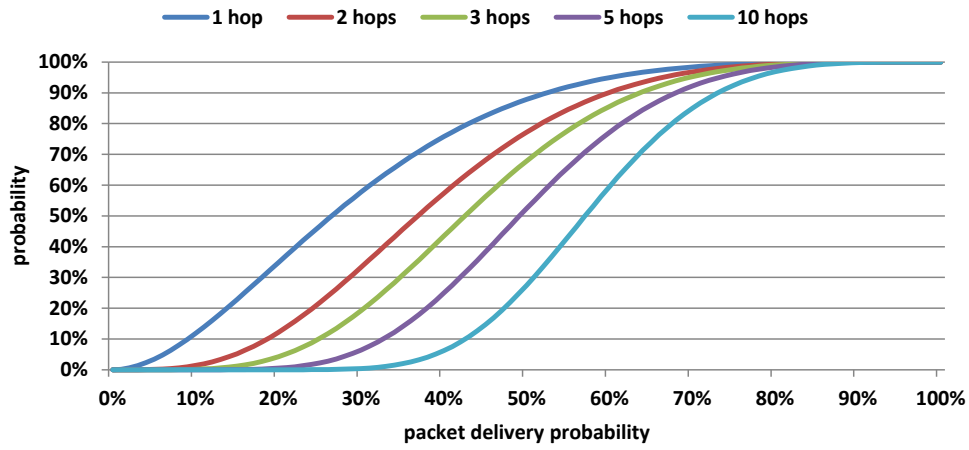


Figure 3.5: Multi-hop success probability.

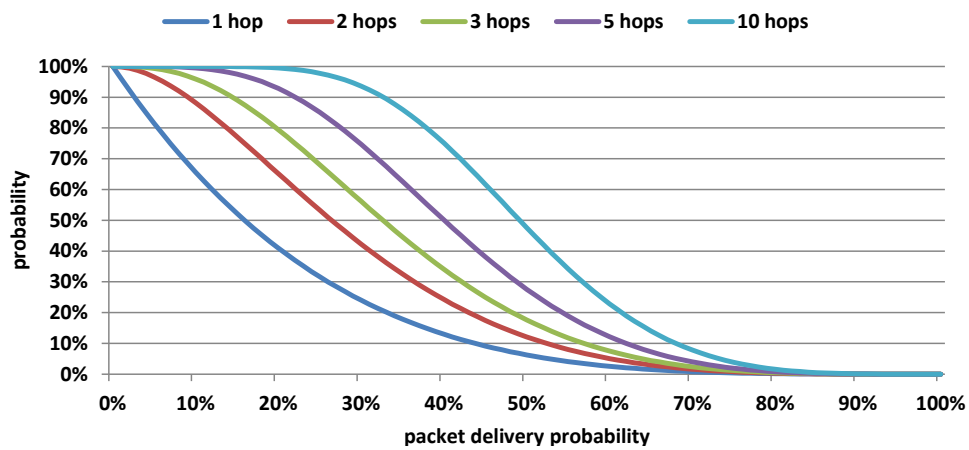


Figure 3.6: Multi-hop soft failure probability.

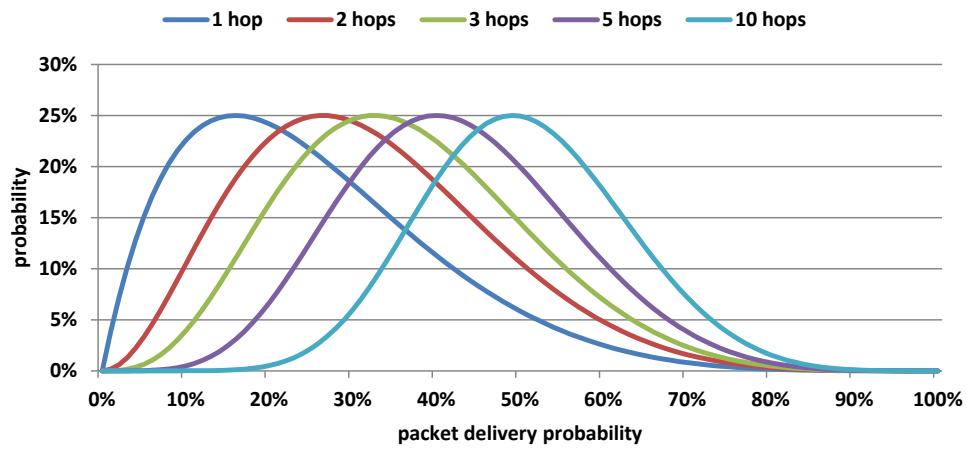


Figure 3.7: Multi-hop hard failure probability.

Chapter 4

Experiments concerning wireless link failures

The experiments discussed in this chapter took place in an office environment with most activity from 8 AM to 7 PM. Failures occurred more frequently during these office hours due to different factors, which are to be investigated. Interferences include reflection on people and movable objects, Wi-Fi/Bluetooth frequency interference, and temperature.

Interferences are not easy to separate, because some are strongly correlated. However, it is possible to measure for example the Wi-Fi signal strength or temperature while running the experiments, to see whether the fluctuations in failures match the Wi-Fi or temperature data. The first experiments show the difference between failures during office hours and non-office hours. Secondly, sun and radiation influences are analyzed to check for a relation with failure data. Then, 2.4 GHz interference is scanned to check for fluctuations able to trigger failures. Finally, indoor temperature is compared with failure data. The last sections show the duration of soft failures and the ratio between hard and soft failures.

4.1 Experiment setup

The experiments focus on the amount and duration of both kind of failures in relation to time and environment. The setup consists of a transmitter (JenNet coordinator) and an actuator (JenNet router). The transmitter sends SPs to the actuator. During experiments, these SPs will be a simple ON/OFF SP. The actuator drives an LED, producing light. The actuator responds with an ACK when a SP has been received. The transmitter receives the ACK and knows that the transmission has been completed successfully.

A wired UDP feedback signal is used to validate the actuator status. A Priva embedded module prototype checks the actuator status by using an LDR (Light Dependent Resistor) that responds to the light controlled by the actuator. The em-

bedded module and the actuator are placed in a box to distinguish ON and OFF more easily. The embedded module sends an UDP packet containing this data.

A laptop is the only "smart object" during experiments. It communicates with the transmitter using serial communication and receives the wired UDP feedback information. It initiates all the SPs (ON/OFF). The transmitter informs the laptop about successful and unsuccessful transmissions, based on received ACKs. The laptop checks if the transmitter and actuator statuses match. A hard failure will be concluded when they do not. When the transmitter informs the laptop about a failed packet and the laptop confirmed a matching status, then the laptop concludes a soft failure. Figure 4.1 and Table 4.1 both illustrate an overview of the experiment setup.

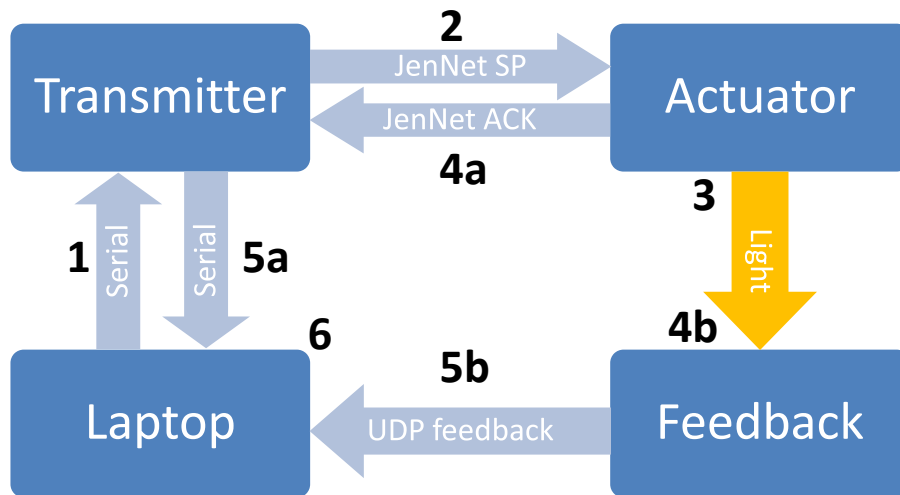


Figure 4.1: Experiment setup overview.

Table 4.1: Experiment flow overview.

| | | | |
|----|-----------------------------------|----|--------------------------|
| 1 | Laptop sends SP to transmitter | | |
| 2 | Transmitter sends SP to actuator | | |
| 3 | Actuator drives led | | |
| 4a | Actuator sends ACK to transmitter | 4b | Feedback monitors light |
| 5a | Transmitter notifies laptop | 5b | Feedback notifies laptop |
| 6 | Laptop compares values | | |
| 7 | Return to 1 with inverted SP | | |

4.2 Experiment environment map

An overview of the floor including obstacles is shown in Figure 4.2. It also gives an impression of the possible reflection paths and the non-ideal shape of the experiment environment. The light grey parts are obstacles with a height of 1 meter, while dark grey represents an obstacle of 1.5 meters. The dark blue desk at the bottom left shows the coordinator location. Line of sight at the outer right part of the office has been marked by the light blue beam. The map shows some locations where the wireless link quality has been determined based on success/failure rate. The experiments need a location in which the link quality is critical. This will generate usable data for environment influences evaluation.

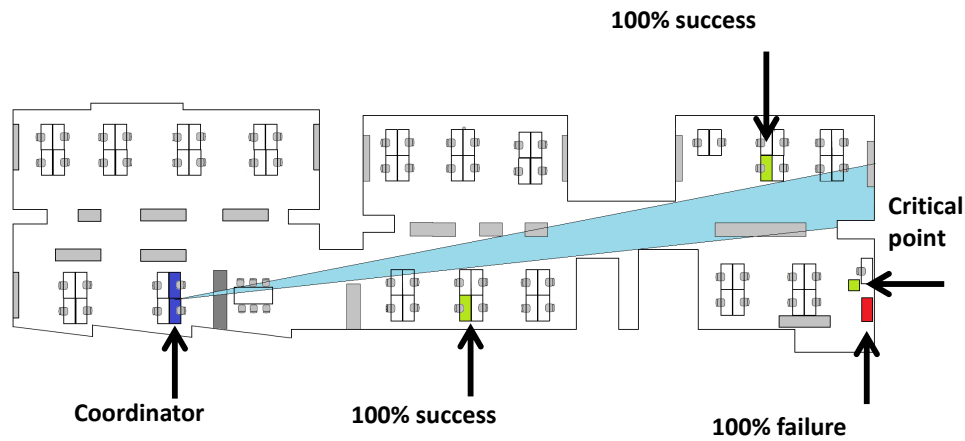


Figure 4.2: Office map of experiment environment.

4.3 Office hours vs non-office hours

Table 4.2 illustrates soft and hard failures during a weekend experiment between Friday afternoon and Monday morning. The data points out the major differences between office hours, before Friday 19:00 and after Monday 08:00, and non-office hours. It also shows that failures are rare in general, indicating a good link quality.

The wireless link quality is sensitive to the antenna angle and antenna location. It is hard therefore to exactly reproduce different ratios. Even so, repeated experiments show similar differences between office and non-office hours. Table 4.3 shows similar results during other experiments.

These experiments show that failures increase significantly during office hours. These differences increase even further when the link quality gets more critical. Observations during experiments indicate that reflections and obstructions of line of sight influence link quality the most during office hours.

Table 4.2: Failures, office vs. non-office, weekend experiment.

| | Time slot | Hard failures | Soft failures |
|------------|---------------------|---------------|---------------|
| Office | Fri 15:51-Fri 19:00 | 0.0877% | 0.6135% |
| Non-office | Fri 19:00-Mon 08:00 | 0.0005% | 0.0009% |
| Office | Mon 08:00-Mon 09:43 | 0.0082% | 0.2035% |

Table 4.3: Failures, office vs. non-office, night experiment.

| | Time slot | Hard failures | Soft failures |
|------------|---------------------|---------------|---------------|
| Office | Wed 09:40-Wed 19:00 | 0.009% | 0.024% |
| Non-office | Wed 19:00-Thu 07:00 | 0.002% | 0.007% |
| Office | Thu 07:00-Thu 10:08 | 0.027% | 0.018% |

Table 4.3 shows something remarkable regarding the ratio of hard and soft failures. The hard failures occur more often during the last time slot (Thu 07:00-Thu 10:08), which is assumed to be because of the small number of failures in general.

4.4 Sun and radiation influences

When the location of the antenna is changed such that the link quality decreases, then something else can be seen from experiment data. Figure 4.3 illustrates an experiment from Thursday 10:00 to Monday 09:00, where failures occur on Saturday and Sunday while no people were present in the experiment environment. So, the interference during the weekend has to be due to an external factor. Failures seem to be almost absent during night and concentrate from 09:00 to 15:00.

There has not been any data recorded regarding interferences during the weekend at the experiment site. Meteorological institutes however, did log data that is used for evaluation. Weather data show a similar ratio between Saturday and Sunday for sun activity and soft failures. Data indicating hours of sun during the experiment show a peak on Saturday; Thu 6 hrs, Fri 5 hrs, Sat 9 hrs, Sun 4 hrs. Sun activity including radiation could possibly have an effect on reflection characteristics in the building and in that way influence the wireless link quality. Figure 4.4 disproves the relation between sun/radiation and soft failures by showing a counterexample. It shows many failures over night and almost none during the day. Other kinds of interference will be targeted in next sections by measuring 2.4 GHz interference and temperature effects to find an explanation for these failures.

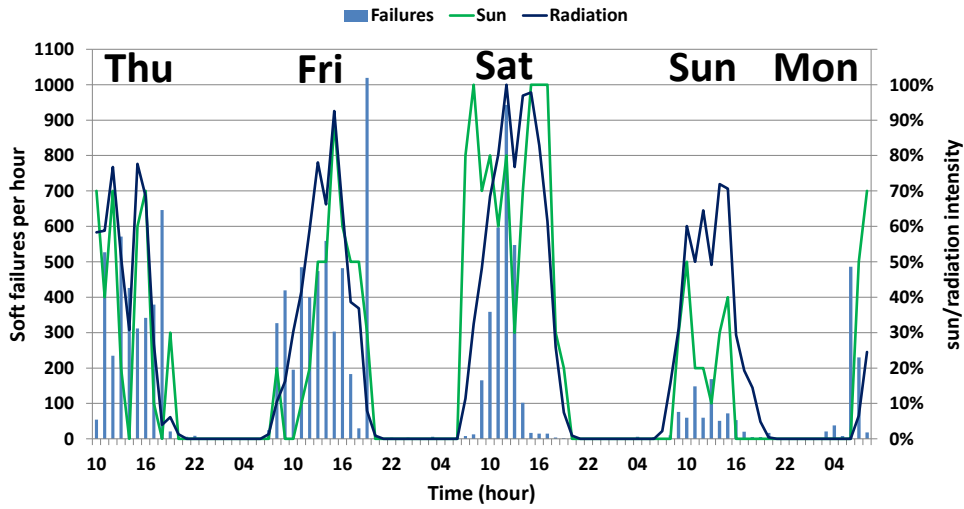


Figure 4.3: Failures vs. sun and radiation, during weekend, matching values.

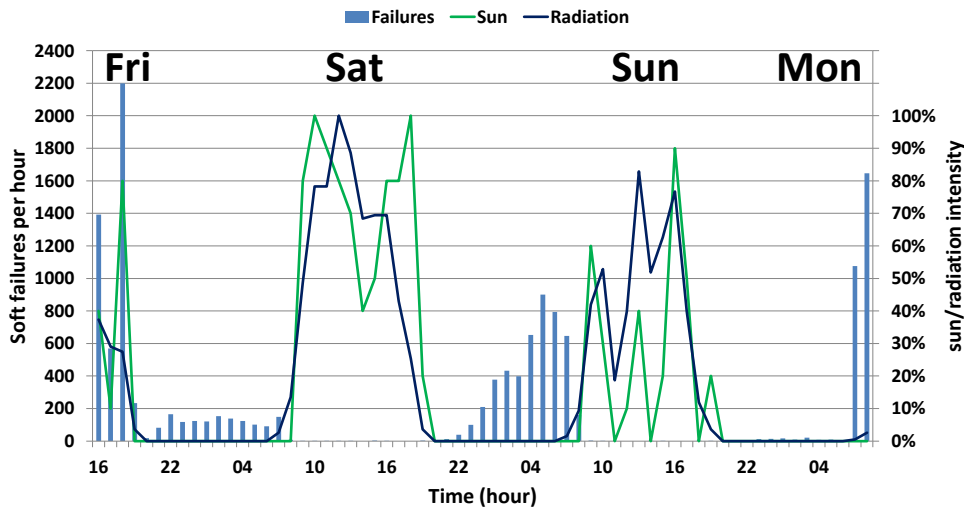


Figure 4.4: Failures vs. sun and radiation, during weekend, counterexample.

4.5 2.4 GHz interference

Signals in the 2.4 GHz frequency range, like Wi-Fi and Bluetooth, could interfere with JenNet communication because they both use the 2.4 GHz ISM frequency band. Figure 4.5 shows the results of a Wi-Fi channel scan within the experiment environment. The channels in use within the experiment environment are 1, 4, and 6. The channels have different power envelopes due to the use of different modulation standards (802.11b and 802.11n). Wi-Fi channels are not identical to

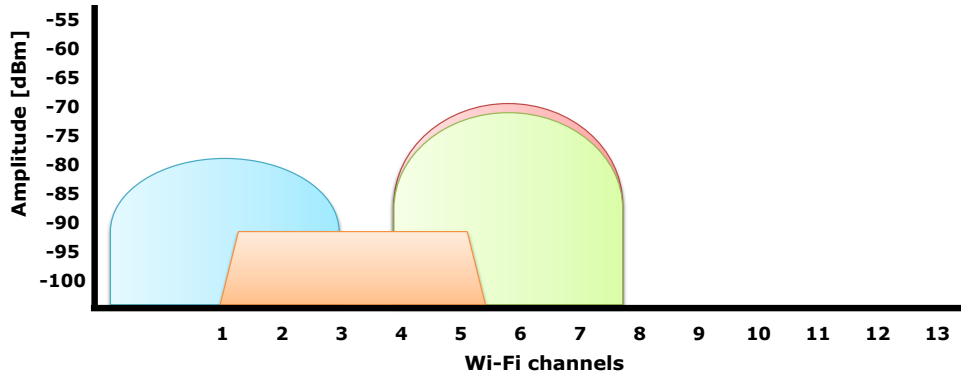


Figure 4.5: Wi-Fi channels used in 2.4 GHz range at experiment environment.

those of JenNet (802.15.4). The Wi-Fi channel bands are wider and overlap while JenNet's do not. Figure 4.6 illustrates both types of channel bands.

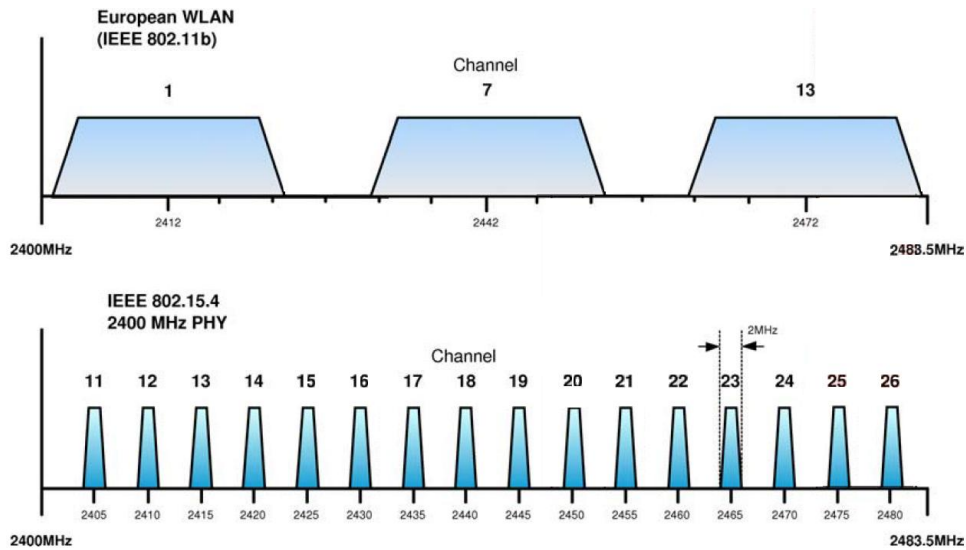


Figure 4.6: Wi-Fi channels mapped onto JenNet (802.15.4) channels.[1]

The Wi-Fi activity has been monitored over the weekend to check for fluctuations and these were not present. This indicates that the Wi-Fi channels do not initiate the fluctuations over the experiment weekends. Channel 12 has been used in previous experiments while its frequency band could be interfered by used Wi-Fi channels. JenNet channel 26 is used from this point to evade the used Wi-Fi channels, and in that way improve the link quality. Bluetooth interference has not been taken into account, because Bluetooth devices are assumed to be unused during the weekend while no persons are present in the experiment environment.

4.6 Temperature effects

The authors of [26] analyzed the transitional region of low power wireless links and concluded that in static environments, variations over time are mainly due to fluctuations in the thermal noise of radios. Figure 4.7 illustrates the temperature during a weekend related to the failures and link quality. It shows that the temperature fluctuates while the number of failures and link quality remain constant. This makes it hard to believe that the temperature fluctuations that are typical for the Priva WSA environment are large enough to be able to initiate failure fluctuations from Figure 4.4.

4.7 Durations consecutive soft failures

Consecutive soft failures take less than 500 ms in most cases, relating to wireless link quality. Figure 4.8 illustrates short (≤ 4 seconds) consecutive soft failure durations. Keep in mind that some could last longer, depending on the environment. Short occurrences could be ignored in most cases, while long ones call for a response depending on the actuator and the application. Antenna diversity could change the shape of Figure 4.8 in a positive way.

4.8 Ratio soft and hard failures

Ratio r_s , introduced in Section 3.4, show the ratio between hard and soft failures according to theory. Figure 4.9 illustrates the real ratio r_s compared to the soft failures per hour during experiments. Figure 4.9 uses failures per hour instead of packet delivery probability P_{pd} , because failure data from the lowest layers has not been available. P_{pd} mainly has values close to 0% or 100%. Probabilities close to 0% lead to ratios around 0.1, while probabilities close to 100% lead to ratios between 10 and infinity. Probabilities close to 100% make it very rare for failures to occur. The graph shows that the scattering of points decreases when failures per hour decreases. This is due to the increased resolution, as a result of more failure data. From the data can be concluded that ratio r_s behaves as expected by the theory.

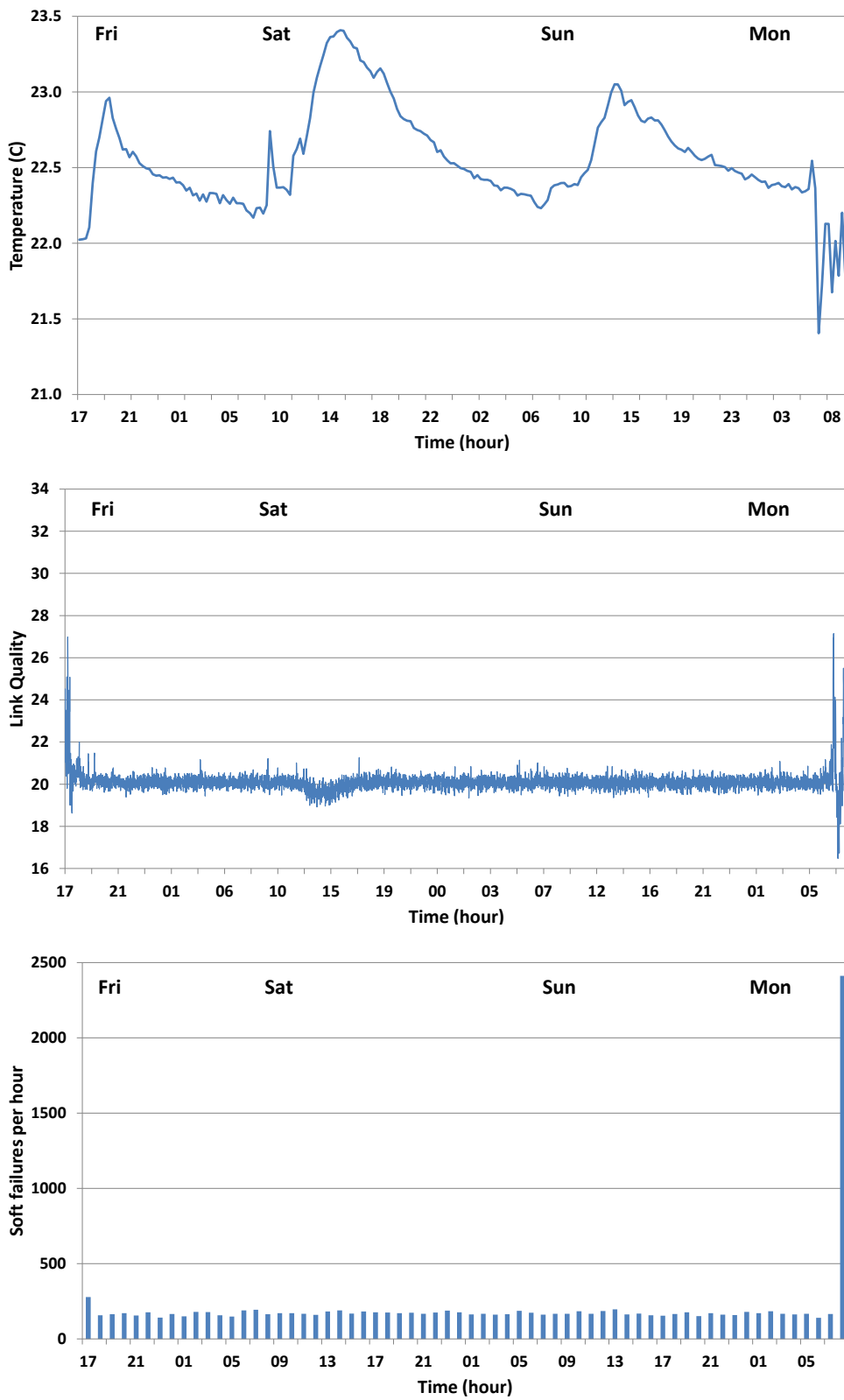


Figure 4.7: Indoor temperature, LQI, and soft failures during weekend experiment.

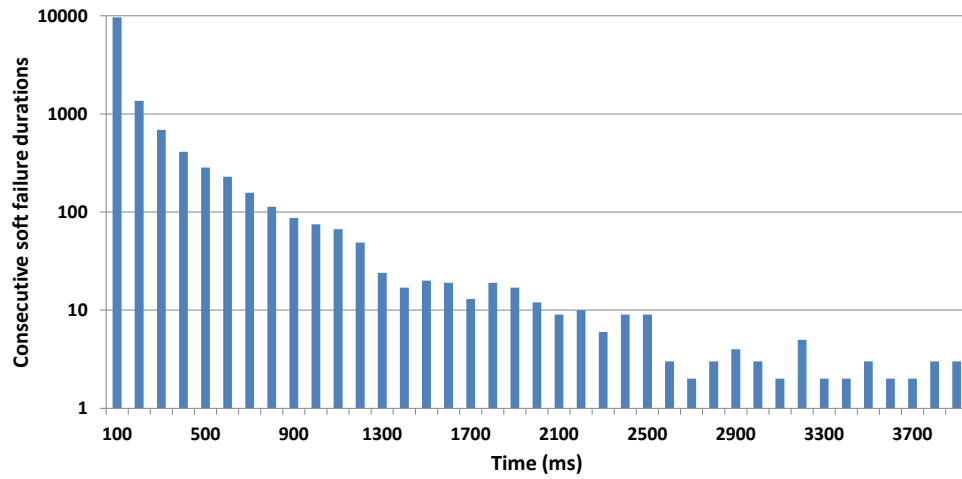
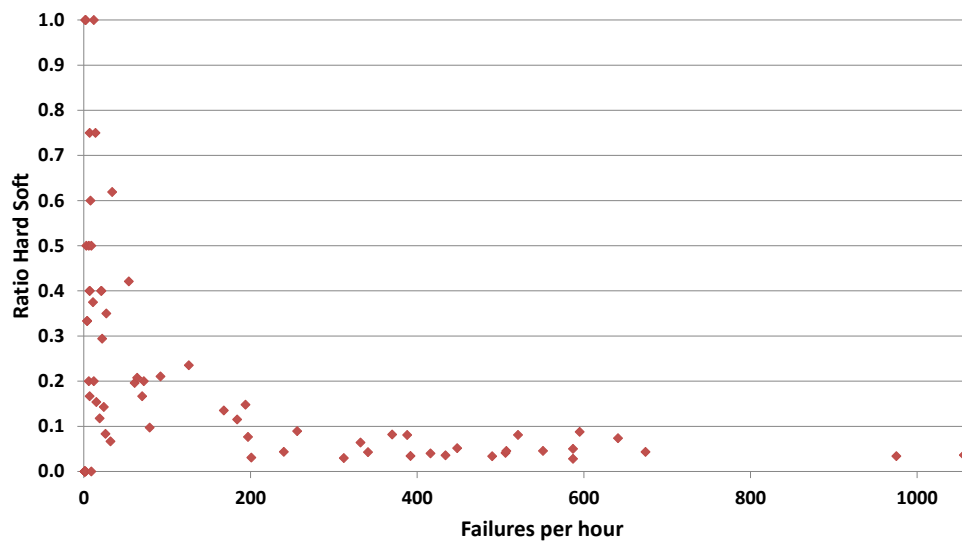


Figure 4.8: Histogram of the duration of consecutive soft failures.

Figure 4.9: Single-hop r_s related to failures per hour.

Chapter 5

Solution for wireless link robustness

Initially, this chapter presents the robustness requirements and the existing solutions. Then, a new solution is proposed suiting these requirements, and the details of the solution are presented by the use of finite state machines, transition diagrams, and sequence diagrams. Finally, the effect of different heartbeat timings on the solution performance is shown by the use of equations.

5.1 Requirements

The robustness solution would need to find and log both kinds of failures. It should solve the hard failures, and need to prevent dangers to the system and its surroundings when soft failures occur. Finally, there should be an interface between the robustness solution, the applications, and the lower layers. The solution should be portable, which means that it could easily be used with a different type of (wireless) communication. So, the main requirements are:

- Find and log both type of failures.
- Solve hard failures.
- Cope with soft failures.
- Interface with applications and lower layers.

Priva has its own vision on the demanded robustness layer, which is the central point of interest. Applications can subscribe to the robustness layer resulting in an error response when a connection fails. It could also manage the duty cycle and sleep of the physical layer. Figure 5.1 illustrates the Priva vision on the robustness layer and its interaction with the other layers.

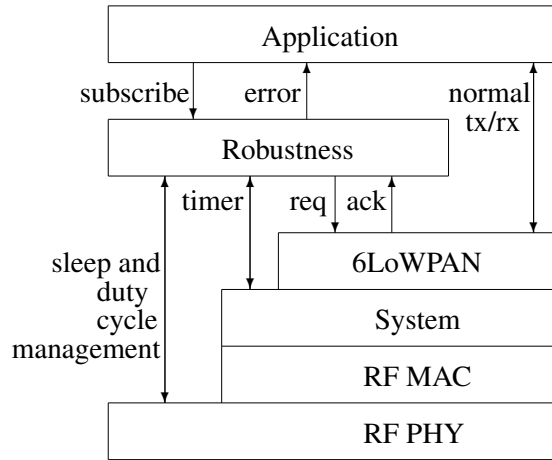


Figure 5.1: The requested Priva robustness layer example.

5.2 Existing solutions

Within the field of low-power wireless communication, different approaches involving robustness and reliability exist. Many solutions focus on failure prevention by the use of for example mesh-networking, channel hopping, and time slotted transmissions. Other solutions increase delivery ratios by tuning MAC parameters [4]. Also, many solutions for low-power wireless communication assume a non-critical data-collection scenario, while this scenario is critical and focuses on actuator control. A heartbeat is used in multiple previous works like [2], but none contains an actuator SP. Other kinds of robustness like [3] use redundant deployment, where all nodes have been placed at least twice to cope with node and link failures.

5.3 Proposed solution

The proposed solution acknowledges the existence of communication failures and works around these failures at a higher level. This solution contains state machines on both sides that look similar but are slightly different. These state machines are being synchronized by the use of a bidirectional heartbeat (HB). HBs also check the link quality and initiate a fail-safe mode whenever the connection has been lost. A recovery mode is used as a transition state between fail-safe and idle. These elements will be explained thoroughly in the following subsections.

5.3.1 State machines

The actuator side of the wireless connection has a similar state machine as the one maintained at the controller side. It includes idle operation mode, fail-safe

mode, and recovery mode. The state machines on both sides should always be synchronous, otherwise a hard failure has occurred. The fail-safe mode is the initial state in the state machine. The state machines contain no exit state [15], because the system will keep on running continuously.

5.3.2 Heartbeats

Link quality is monitored by the usage of HBs, which are used to find soft failures. If there are x consecutive missed HBs, then the local end (controller or actuator) will conclude a connection loss. The number x is predetermined and dependent on the actuator and application. In this way, the controller will stay synchronized with the actuator because they both act in the same manner. The HB contains the status (idle/fail-safe/recovery) of the HB-originator and its SP. If the status from the received HB differs from the local status, then the local status will be changed to match the status from the received HB. In this way, both state machines will recover from a hard failure with every HB. This also holds for different SP values. HBs solve a situation in which a wireless link works asymmetrically, because of the state information inside the HBs. The state machines will show some exceptions.

A system without HBs would retry a failed packet and in that way increase traffic when the link quality is low. This has a large failure probability and a small chance of solving the problem. The HBs use static timings that do not lead to traffic peaks. The HB timings differ from each other in order to prevent a system dead lock. These static timings let the controller resend SPs when HBs succeed and in that way decreases failure probability and increases chances to solve problems. Finally, the HBs make it possible to monitor the link quality, making the system able to act when the failure probability increases.

5.3.3 Fail-safe mode

The fail-safe mode prevents the system from damaging itself or its environment. The actuator executes a fail-safe mode routine when in fail-safe mode. This routine differs for each type of actuator and for each application. The routine for a window position actuator could be to close the window. The routine should be placed on the actuator side before the system starts its idle routine.

5.3.4 Recovery mode

The recovery mode will be entered when both sides are in fail-safe mode and the controller receives a HB containing the fail-safe mode. Recovery mode also contains a routine for the actuator side that is dependent on the actuator type and application. Such a routine for a gas burner for example would be to adjust the flow of gas in such way that the gas burner is ready to handle SPs again. The routine has to be present on the actuator side before the system starts the idle routine. Both state machines return to idle mode by a new SP initiated by the controller side. This SP would most likely be the last used SP.

5.3.5 Key points

The key points of the robustness solution are:

- State machines (Figure 5.2) located on both sides (controller and actuator).
- State machines can be out of sync (hard failure) due to lost ACKs.
- A bidirectional HB will be used to be able to detect communication losses (leading to soft failures).
- The HBs will contain the current state to find and solve hard failures.
- When a predetermined number of consecutive soft failures occur, then both state machines will go into fail-safe mode.
- A single-side observed communication failure leads to a hard failure.
- When communication recovers (successful HBs), then both state machines will recover and return to idle mode.
- A single-side observed communication recovery leads to a hard failure.
- The robustness layer informs the application layer when communication can not be guaranteed.
- A robustness-index can be expressed in percentages of successes, soft failures, and hard failures.

5.4 Finite state machines

Figures 5.2 and 5.3 illustrate the finite state machines (FSMs) located on the controller and actuator. Instead of multiple SPs, only two states are chosen now for simplicity reasons. The threshold for the number of consecutive HBs, which is x in figures 5.2 and 5.3, is determined by the application on top of the robustness layer and by the actuator. The state machines differ because the SPs are sent by the controller and received by the actuator. The controller only changes its SP when an ACK has been received, while the actuator does not need to wait for an ACK.

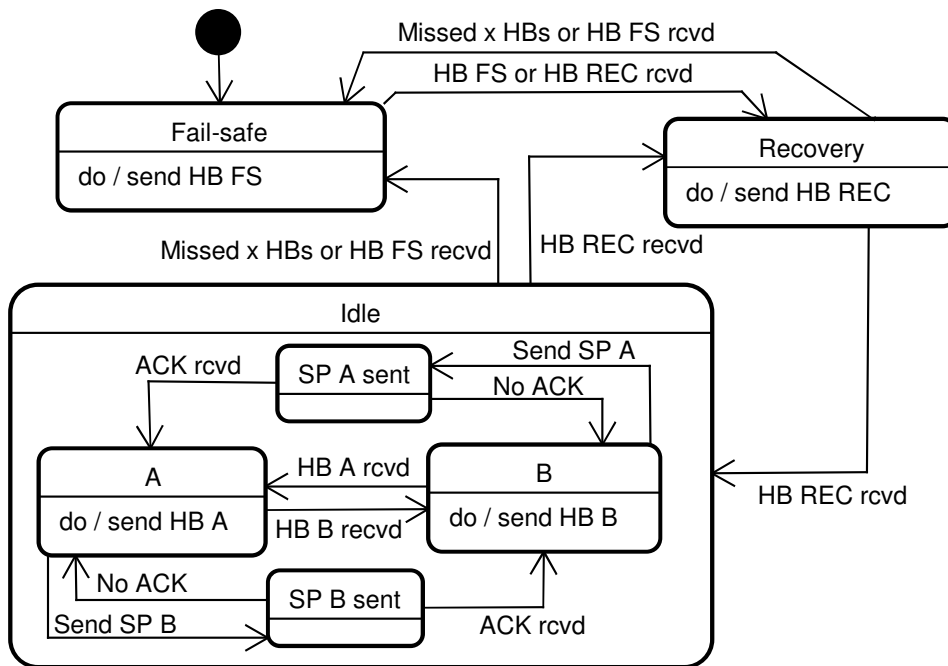


Figure 5.2: Robustness finite state machine at the controller, where FS represents Fail-safe and REC Recovery.

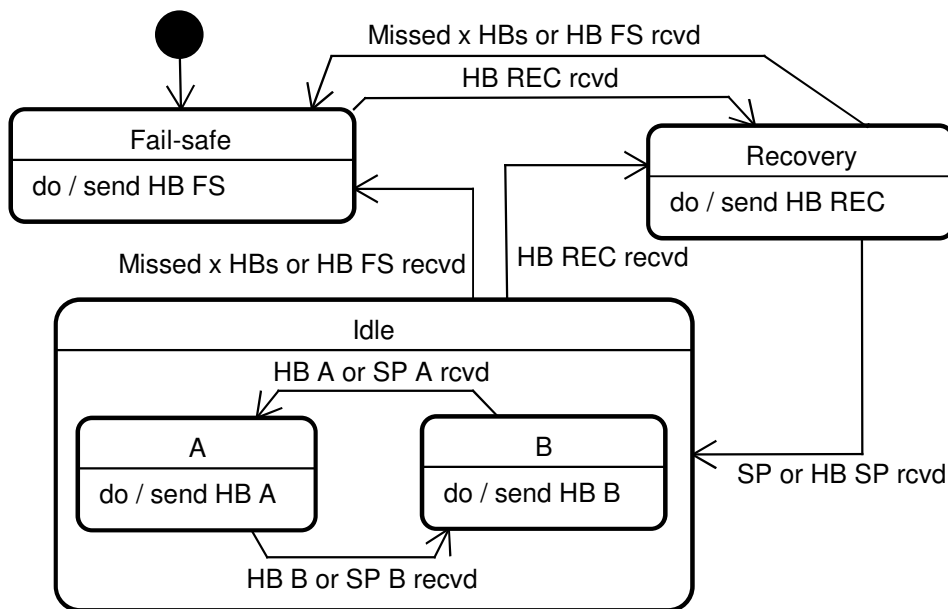


Figure 5.3: Robustness finite state machine at the actuator, where FS represents Fail-safe and REC Recovery.

5.5 Transition diagrams

Tables 5.1 and 5.2 illustrate the transition diagrams for the controller and actuator. The horizontal axis shows current states, while the vertical represents state transition initiators. Table 5.1 contains empty cells because the SPs are not being sent while in fail-safe or recovery state, and when the new SP does not differ from the previous.

Table 5.1: Transition diagram at controller.

| | Idle A | Idle B | Fail-safe | Recovery |
|------------------------------|---------------|---------------|------------------|-----------------|
| Sent SP A | - | Idle A | - | - |
| Sent SP B | Idle B | - | - | - |
| Received HB Idle A | Idle A | Idle A | Fail-safe | - |
| Received HB Idle B | Idle B | Idle B | Fail-safe | - |
| Missed x HBs | Fail-safe | Fail-safe | Fail-safe | Fail-safe |
| Received HB Fail-safe | Fail-safe | Fail-safe | <i>Recovery</i> | Fail-safe |
| Received HB Recovery | Recovery | Recovery | Recovery | <i>Idle...</i> |

Table 5.2: Transition diagram at actuator.

| | Idle A | Idle B | Fail-safe | Recovery |
|------------------------------|---------------|---------------|------------------|-----------------|
| Received SP A | Idle A | Idle A | Fail-safe | Idle A |
| Received SP B | Idle B | Idle B | Fail-safe | Idle B |
| Received HB Idle A | Idle A | Idle A | Fail-safe | Idle A |
| Received HB Idle B | Idle B | Idle B | Fail-safe | Idle B |
| Missed x HBs | Fail-safe | Fail-safe | Fail-safe | Fail-safe |
| Received HB Fail-safe | Fail-safe | Fail-safe | <i>Fail-safe</i> | Fail-safe |
| Received HB Recovery | Recovery | Recovery | Recovery | <i>Recovery</i> |

5.6 Robustness layer interfaces

The robustness layer has multiple interfaces as illustrated in Figure 5.1. Its task is to inform the application layer about the wireless link, while controlling and listening to the lower layers.

The robustness API (Application Programming Interface) mainly includes:

- Subscription, informing the robustness layer about application properties.
- Set setpoint, a command to set the actuator SP.
- Error feedback signal, feedback for the application about the wireless link status, only when relevant.

Applications with different properties can subscribe to the robustness layer. They inform the robustness layer about their properties at subscription, which are:

- Deadline, the deadline for a soft failure (related to missed HBs).
- Jitter, the jitter of the actuator (related to typical consecutive soft failures).
- Fail-safe mode routine, a routine executed at entry of fail-safe mode.
- Recovery routine, a routine executed at entry of the recovery mode.

The robustness layer could cancel application requests when the wireless link quality drops below the one required by the application. This could be for example when expecting a soft failure.

The robustness layer sets parameters, that have been regarded to be variable at previous chapters, like for example the threshold for consecutive lost HBs. These parameters are related to the applications that subscribe to the robustness layer. Figure 5.4 illustrates the use case diagram for the robustness layer at the controller side, while Figure 5.5 illustrates the use case diagram for the robustness layer at the actuator side.

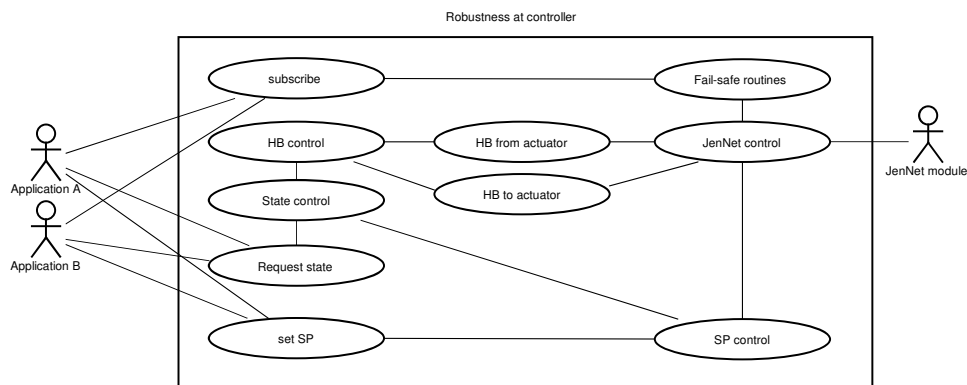


Figure 5.4: Use case diagram for the robustness layer at controller.

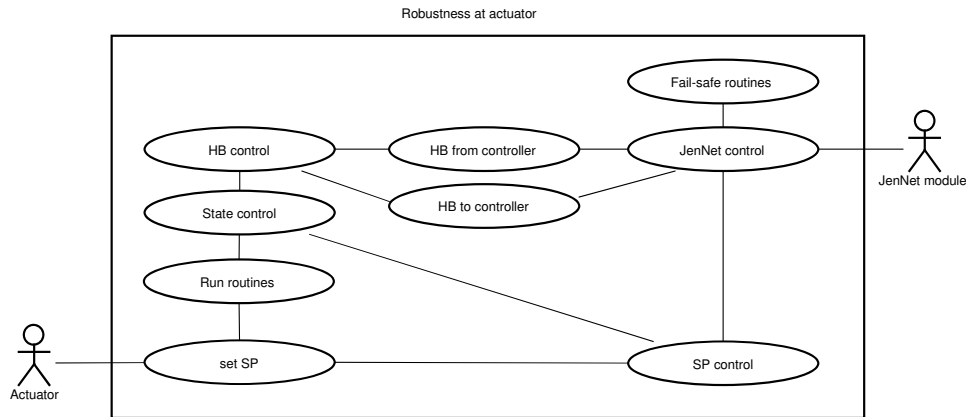


Figure 5.5: Use case diagram for the robustness layer at actuator.

5.7 Sequence diagrams

Sequence diagrams show the interfaces between the application layer, robustness layer, JenNet controller, JenNet actuator, and the actuator. Figure 5.6 illustrates the initialization sequence in which the application subscribes to the robustness layer and confirms the deadlines, jitter, fail-safe mode routine, and recovery routine.

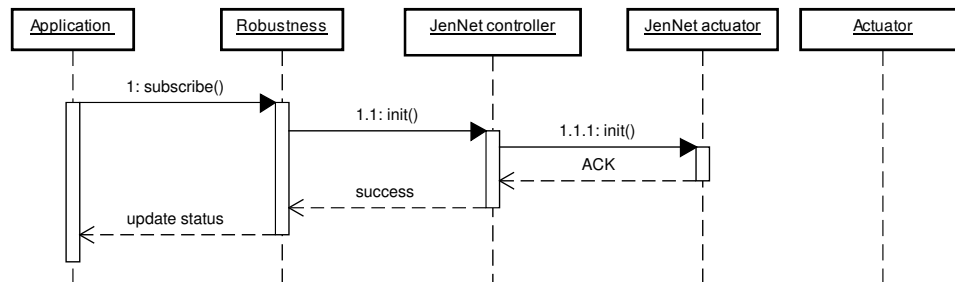


Figure 5.6: Sequence diagram of initialization.

Figure 5.7 illustrates the sequence in which the application sets a new SP at the robustness layer. The robustness layer issues the SP to the JenNet controller, which sends the SP over JenNet towards the JenNet actuator. The JenNet actuator controls the real actuator SP, while acknowledging it. The JenNet controller confirms the new SP to the robustness layer, which will update the new status to the application.

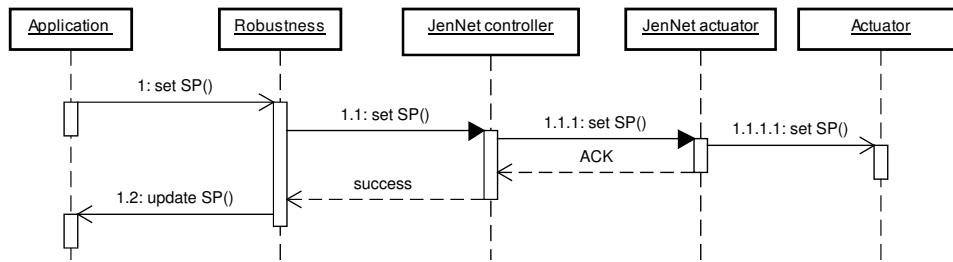


Figure 5.7: Sequence diagram of a successful setting of the SP.

Figure 5.8 illustrates the HB communication that solves hard failures and finds soft failures. The robustness layer orders the JenNet controller to send a HB to the JenNet actuator containing the current SP. Both sides sends the HBs periodically at fixed times, but differ from each other to prevent system deadlocks. The HB information is passed on to the robustness layer to compare it with the current state and SP. A timer will trigger the fail-safe mode when x consecutive HBs are missed.

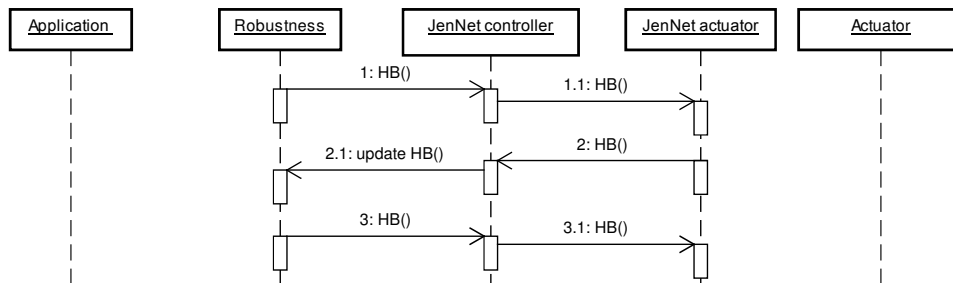


Figure 5.8: Sequence diagram of some HBs.

Figure 5.9 illustrates a soft failure. The application sets a new SP at the robustness layer, which fails to send the data over JenNet. The application will not be informed about this issue because the system has not changed its mode to fail-safe. The robustness layer will retry as long as the configured deadline and jitter allow it to do so.

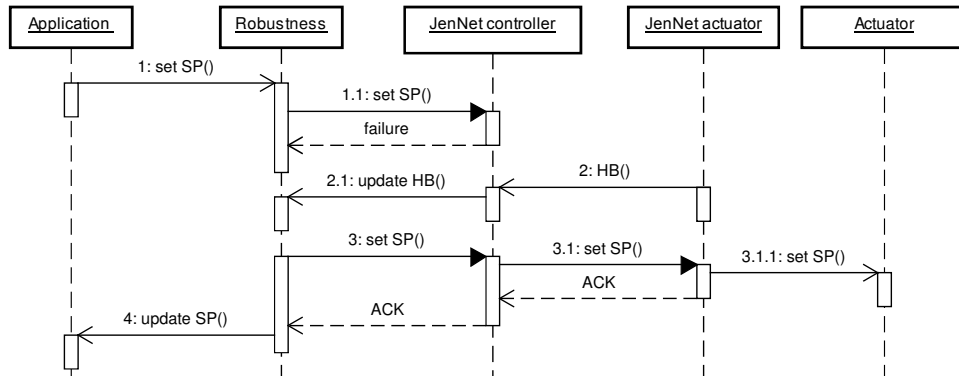


Figure 5.9: Sequence diagram of a soft failure.

Figure 5.10 illustrates a few consecutive soft failures taking long enough to trigger the fail-safe mode. HBs could fail multiple times without informing the application. The robustness layer informs a status update when the configured threshold has been exceeded. The system status changes to recovery mode when HBs start to be successful. The robustness layer orders the JenNet controller to send the latest SP and will update the application when successful.

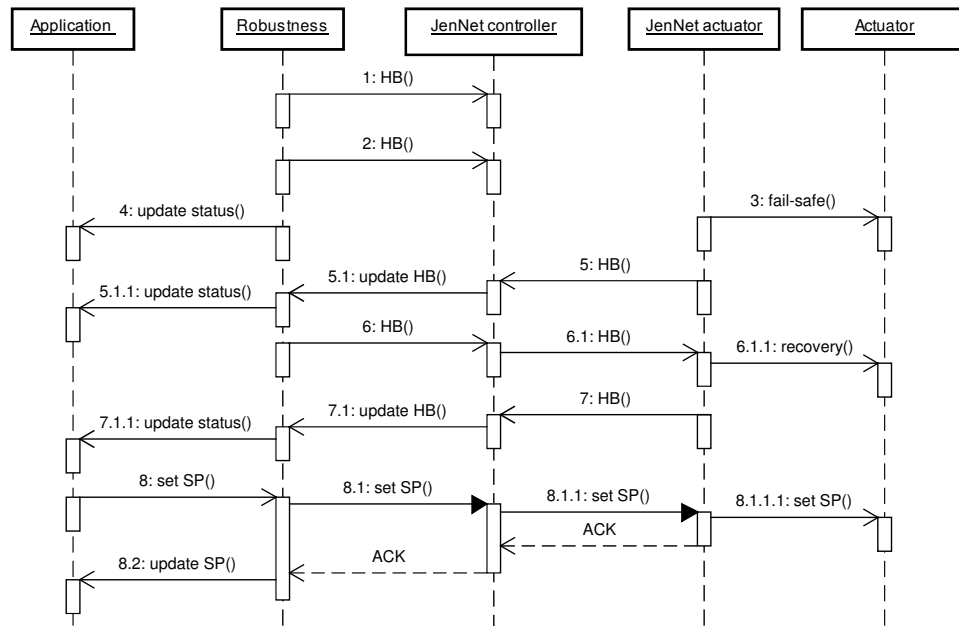


Figure 5.10: Sequence diagram of a few consecutive soft failures.

Figure 5.11 illustrates a hard failure. The JenNet controller sends the SP without receiving the ACK, while the JenNet actuator received the SP correctly. The ac-

tuator now issues the actuator with the new SP. The robustness layer receives a failure, which triggers a retransmission. This retransmission fails in the example, while the HB containing the actuator status succeeds. The HB informs the robustness layer triggering a SP update. The robustness layer now informs the application about the actuator status.

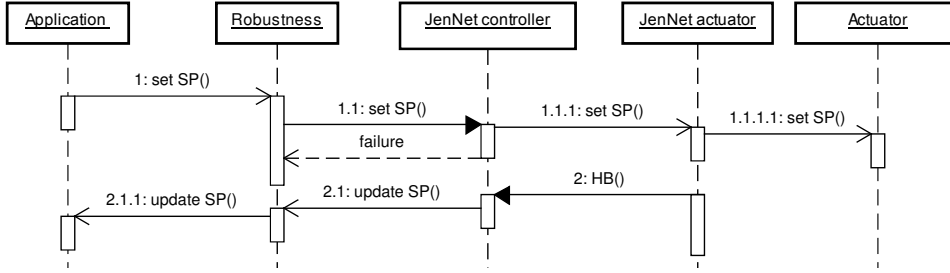


Figure 5.11: Sequence diagram of a hard failure.

5.8 Heartbeat timing

The time that HBs need to solve hard failures depends on the HB frequency. The frequencies of the HBs (towards and from actuator) differ to prevent HBs from continuously sending at the same time. Performance changes as a result of different HB timings could be calculated by some probability equations. The equations introduced in this section of the thesis assume the time of flight of HBs to be zero, while in reality this is not. Time of flight varies because of retransmissions and hopcount. It means that in reality HBs could be travelling at the same time and create a hard failure directly after solving one. Time of flight is assumed not to be long enough to create multiple consecutive occurrences in which HBs travel at the same time, so the first HB after the occurrence would solve it permanently. The equations also assume the HB originated by the controller to be random. In reality, this could be different because the controller knows when it sends its SP.

Assume HB A and HB B being sent every t_A and t_B ms respectively, where H_A and H_B are the times left until the next HB. Thus, for H_A this would range between 0 and t_A . Assume $t_A < t_B$. The variable X denotes the time it takes to solve a hard failure. This means that $X \leq t_A$, because $t_A < t_B$ and the equations in this section assume no HB losses. The variables are visualized in Figure 5.12. A hard failure is solved within X ms when one of the HBs reaches the other side. HB A could arrive first with probability $P_A(x)$ as shown in equation (5.1a), and HB B with $P_B(x)$ shown in equation (5.1b). The probability for both HBs to arrive exactly at the same time approaches zero in theory and is in that way negligible.

$$P_A(x) = P[H_A = x]P[H_B > x] \quad (5.1a)$$

$$P_B(x) = P[H_B = x]P[H_A > x] \quad (5.1b)$$

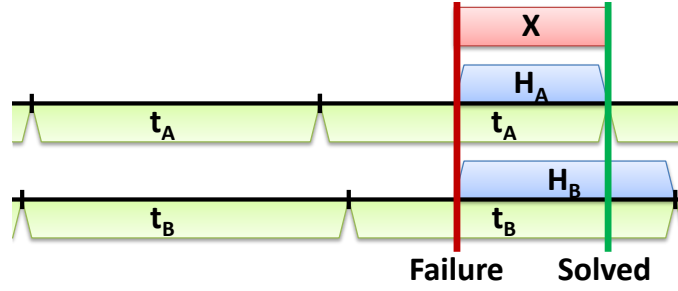


Figure 5.12: Variables t_A , t_B , H_A , H_B , and X explained.

The probability to solve a hard failure $P_X(x)$ at ms X is calculated by adding both previous equations, as illustrated in equations (5.2).

$$P_X(x) = P[X = x] = P_A(x) + P_B(x) \quad (5.2a)$$

$$P_X(x) = P[H_A = x]P[H_B > x] + P[H_B = x]P[H_A > x] \quad (5.2b)$$

The probability as a function of X , H_A , and H_B between zero and t_A is shown in equations (5.3). The probability beyond t_A is equal to zero.

$$P_X(x) = \frac{1}{H_A} \left(1 - \frac{x}{H_B}\right) + \frac{1}{H_B} \left(1 - \frac{x}{H_A}\right) \quad (5.3a)$$

$$P_X(x) = \frac{1}{H_A} - \frac{x}{H_A H_B} + \frac{1}{H_B} - \frac{x}{H_A H_B} \quad (5.3b)$$

$$P_X(x) = \frac{1}{H_A} + \frac{1}{H_B} - \frac{2x}{H_A H_B} \quad (5.3c)$$

The cumulative probability is calculated by integrating the probability distribution as shown in equations (5.4). The cumulative probability is equal to 1 beyond t_A .

$$F_X(x) = P(X \leq x) = \int \left(\frac{1}{H_A} + \frac{1}{H_B} - \frac{2x}{H_A H_B} \right) dx \quad (5.4a)$$

$$F_X(x) = \frac{x}{H_A} + \frac{x}{H_B} - \frac{x^2}{H_A H_B} \quad (5.4b)$$

Figure 5.13 shows a graph where t_A and t_B are changed from 100/200 ms to 500/600 ms, while keeping the difference in time δt at 100 ms. It shows significant differences in performances due to these changes. Figure 5.14 shows what happens when the δt is changed from 250 to 50 ms, while keeping the average value of t_A and t_B constant. It shows that the differences are small, but noticeable. Differences increase above the 90% cumulative probability. Finally, Figure 5.15 shows the behavior of the probability when changing the t_B from 400 to 600 ms, while keeping the t_A static at 300 ms. The differences in this graph are minor, so it can be concluded that t_B only has a small influence on the performance.

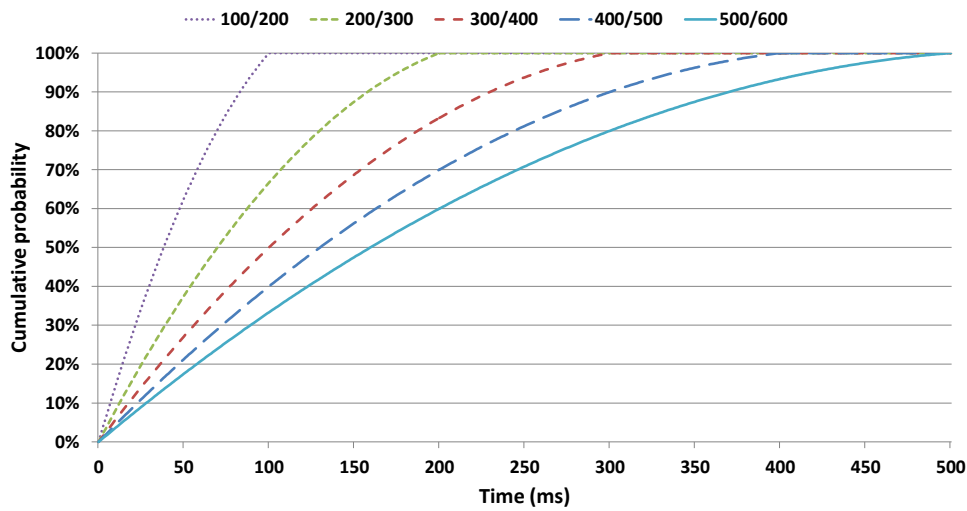


Figure 5.13: Cumulative probability of solving a hard failure, when varying t_A and t_B with constant δt . The times (ms) in the legend denote t_A/t_B .

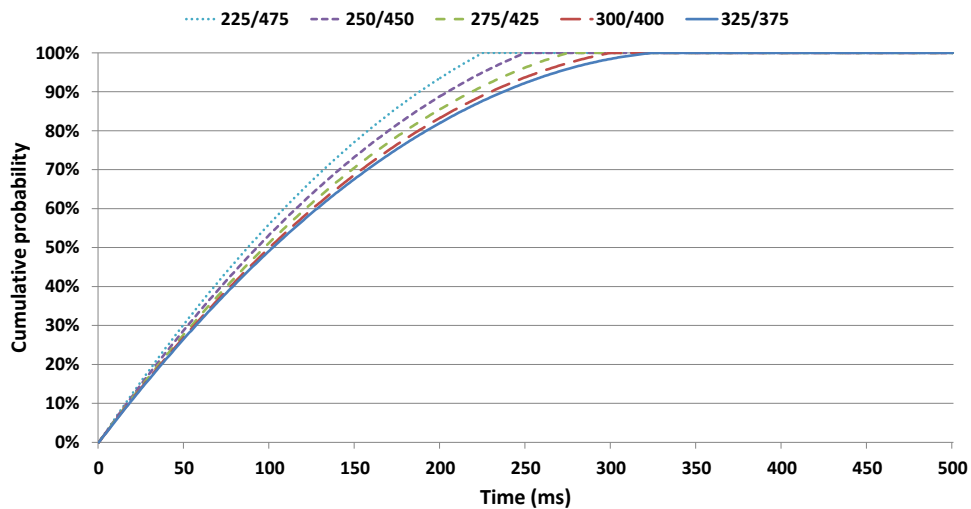


Figure 5.14: Cumulative probability of solving a hard failure, when varying t_A and t_B with variable δt . The times (ms) in the legend denote t_A/t_B .

Analysis shows the impact of HB timings on the behavior of the solution in theory. The analysis focussed on how well hard failures are solved, while HBs also monitor the link quality and initiate the fail-safe mode. The smaller HB timings will find soft failures better, because smaller timings let the system monitor the link quality with a higher frequency. An advantage is more data to work with, but the downside is the increased required data rate. The models used in this chapter

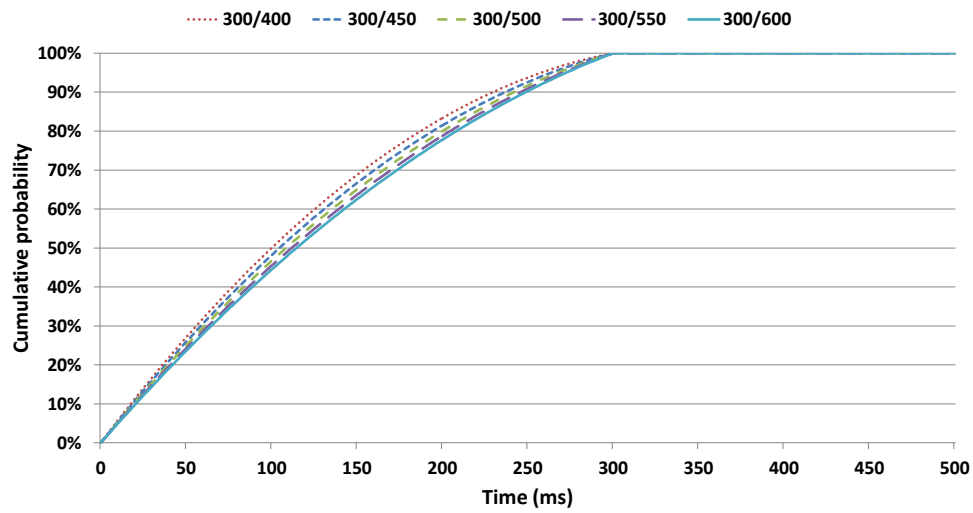


Figure 5.15: Cumulative probability of solving a hard failure, when varying t_B and δt with constant t_A . The times (ms) in the legend denote t_A/t_B .

do not consider packet delivery failures, while the next chapter uses experiments where packet delivery failures do occur.

Testing the proposed solution

The purpose of the following experiments is to prove the workings of the robustness solution and to show the results, regarding the requirements. Chapter 4 showed results regarding the link properties and behaviors, so it is not needed to go into details on such issues in this chapter. Because of this, the solution experiments presented in this chapter can suffice by simulating hard failures and in that way increase experiment intensity. It should be noticed that simulated hard failures occur during better link quality conditions as real hard failures. Data differences as a result of the simulated failures are assumed to be negligible.

6.1 Experiment setup

Figure 6.1 illustrates the setup for the experiments. The laptop again is the centre of

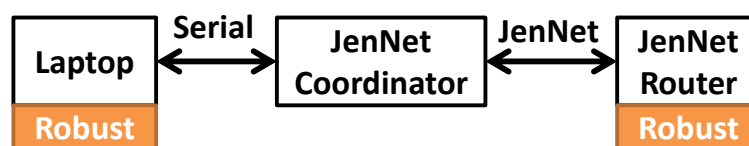


Figure 6.1: Experiment setup overview.

the experiments. It initiates the SPs and HBs from the controller side. It also logs the data and system states. The JenNet router, which is the actuator, also initiates activity. It does not initiate SPs, but it sends HBs periodically. The JenNet router does not log data, so system states transitions and solved hard failures can only be logged at the actuator side. The implementation does not include a complete fail-safe mode or recovery routine. The system will turn the led off during the fail-safe state to simulate a simple implementation. In reality, this could take longer depending on the actuator and application type.

6.2 State ratios

Figure 6.2 illustrates the solution's states during a weekend experiment.

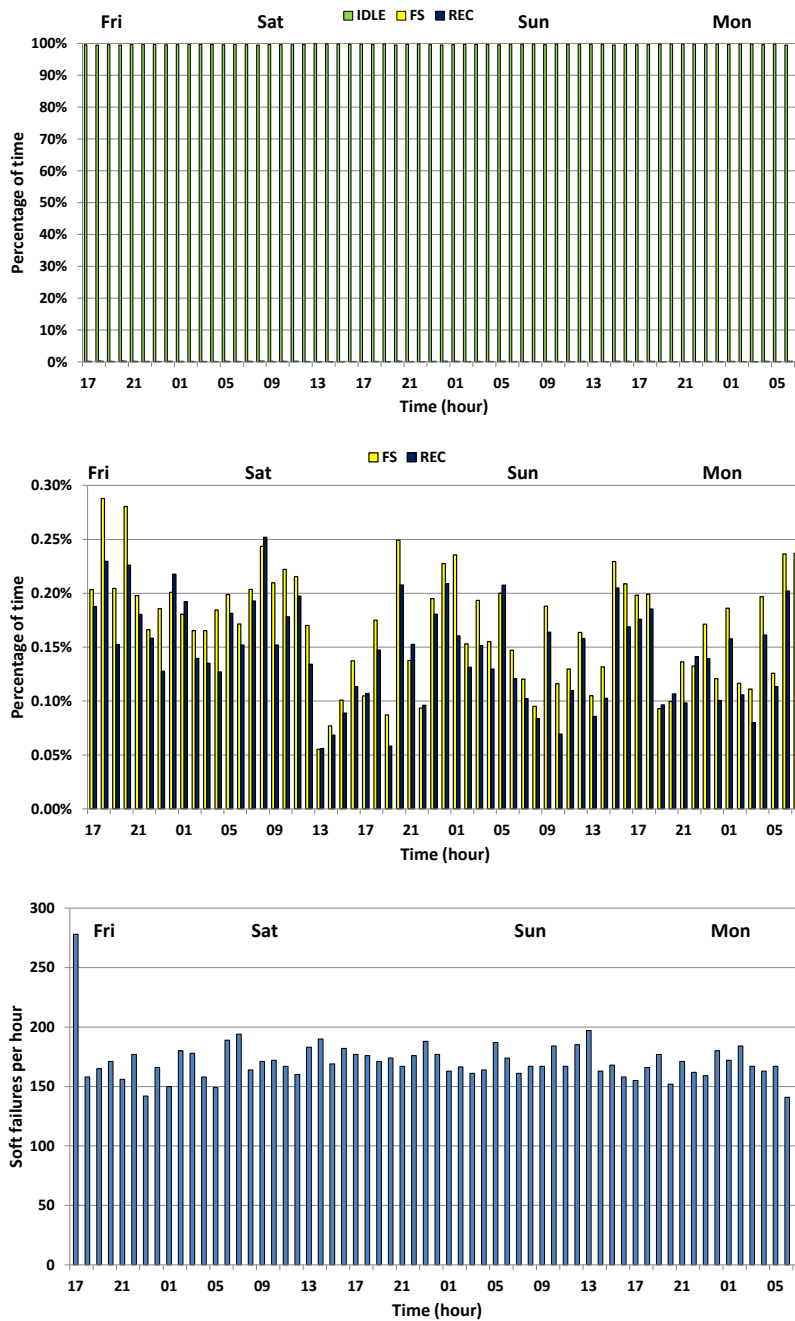


Figure 6.2: Percentages of states, and number of failures during weekend.

It shows percentages in which states have been active. Over the weekend, only around 250 failures per hour occur, making the idle-state most dominant in the top subfigure. The fail-safe and recovery modes are better visualized in the middle subfigure, showing that fail-safe mode takes longer in general than recovery mode. This is fairly reasonable, because the recovery mode starts when link quality improves, while fail-safe mode could take as long as the link quality is bad. The recovery mode is a transition state, which is not supposed to be active for long.

6.3 State transitions

The state machine contains only 3 states, which are fail-safe, recovery, and idle. Table 6.1 shows data from two experiments, Exp1 and Exp2, which have different failure properties. It shows the ratios between state transitions to fail-safe and from recovery, which are equal because of the state machine. The ratios depend on the

Table 6.1: State transitions ratios at actuator.

| | | |
|-------------------|-------|-------|
| Towards fail-safe | Exp 1 | Exp 2 |
| From idle | 85% | 58% |
| From recovery | 15% | 42% |
| From recovery | Exp 1 | Exp 2 |
| Towards idle | 85% | 58% |
| Towards fail-safe | 15% | 42% |

link quality, because a bad link quality would increase the possibility for a transition from recovery to fail-safe instead of a transition from recovery to idle. Table 6.2 shows the origin causing the fail-safe mode. The soft failure is significantly more likely to cause a fail-safe mode than an incoming HB.

6.4 Solving hard failures

Hard failures are not likely to occur in general, so the hard failures are initiated randomly to increase experiment intensity. The simulated failure rate could be increased to for instance 90%, but a failure percentage of 10% has been chosen to keep the environment properties reasonable.

Example: The experiment only uses ON and OFF as SPs. Assume a succeeded ON SP delivery during the experiment. The laptop will now assume the packet to be failed (OFF) in 10% of the cases, to simulate a hard failure. The actuator (ON)

Table 6.2: Origin fail-safe at actuator.

| Fail-safe because of | Exp 1 | Exp 2 |
|----------------------|-------|-------|
| soft failure | 99.9% | 97.4% |
| incoming HB | 0.1% | 2.6% |

| Fail-safe because of | From idle | From recovery |
|----------------------|-----------|---------------|
| soft failure | 95.5% | 100% |
| incoming HB | 4.5% | 0% |

and controller (OFF) are now out of sync. The system should recognize the hard failures during the experiment and solve it. There are three cases in which a hard failure is solved, which are:

- The HB of the laptop will reach the actuator, and in that way change the status of the actuator. This can not be logged directly, but derived from a successful HB.
- The HB of the actuator will reach the laptop, and in that way change the status of the laptop. This can be logged directly from the HB information.
- The laptop sends a HB to the actuator, while the actuator sends a HB at the same time. The wireless link can only be controlled by one side because of CSMA, but the simultaneous HBs can still occur because the JenNet coordinator acts as a small buffer as could be seen in Figure 6.3. Experiments show that this only happens around 1% of the cases.

Table 6.3 shows data from an experiment during weekend taking around 65 hours. It has a simulated hard failure rate of 10%. The data shows that hard failures are mostly solved at the controller side and double HBs are rare. The HB timings used for this data are 400 ms for the actuator and 500 ms for the controller.

Table 6.3: Ratios in solving hard failures.

| | |
|----------------------|-------|
| Solved at actuator | 14.5% |
| Solved at controller | 84.8% |
| Double HBs | 0.7% |

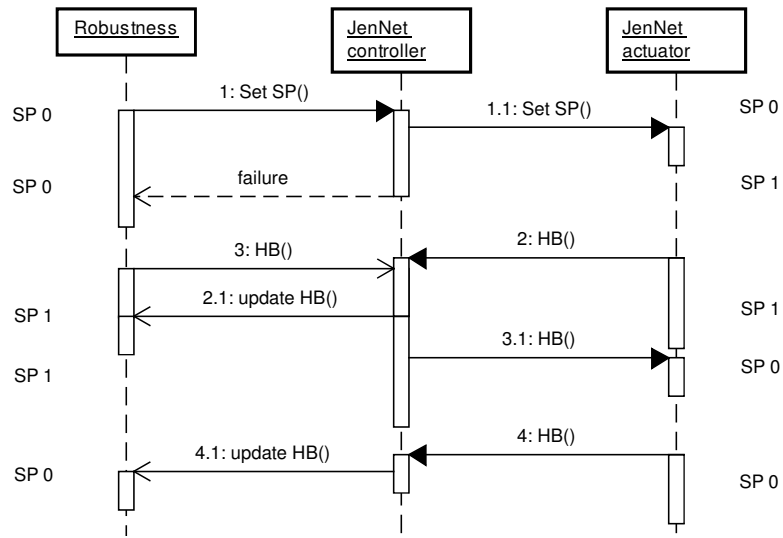


Figure 6.3: Problems occurring because of JenNet controller buffer effect.

6.5 Hard failure durations

Figure 6.4 illustrates the time it takes to solve a hard failure. The time to solve is related to the time between HBs, and could be configured to match different application requirements. The HB timings used for this data are 400 ms for the actuator and 500 ms for the controller. The graph shows that the cumulative probability from experiment data is slightly smaller as the probability from the theory. This is most probably due to packet failures.

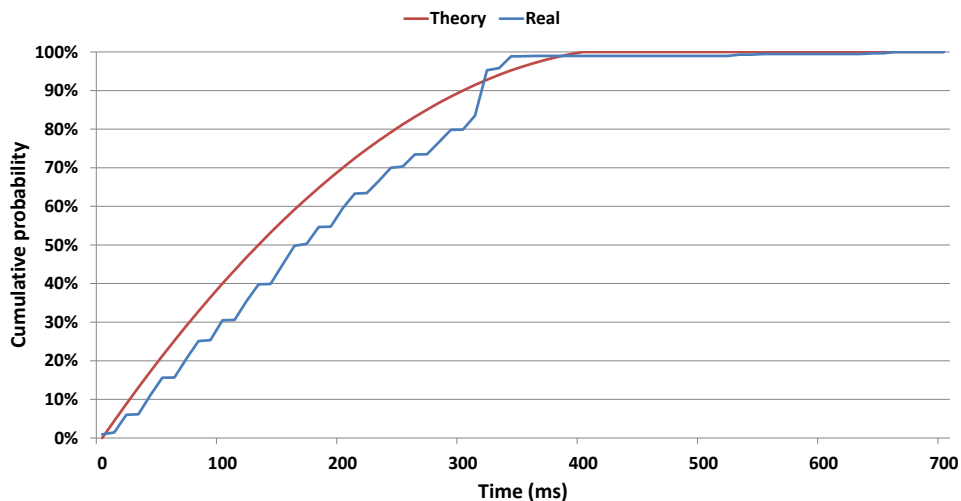


Figure 6.4: Solving hard failures duration, theory vs. experiment data.

Figure 6.5 illustrates the time it takes to solve hard failures when the frequency of the HBs is changed.

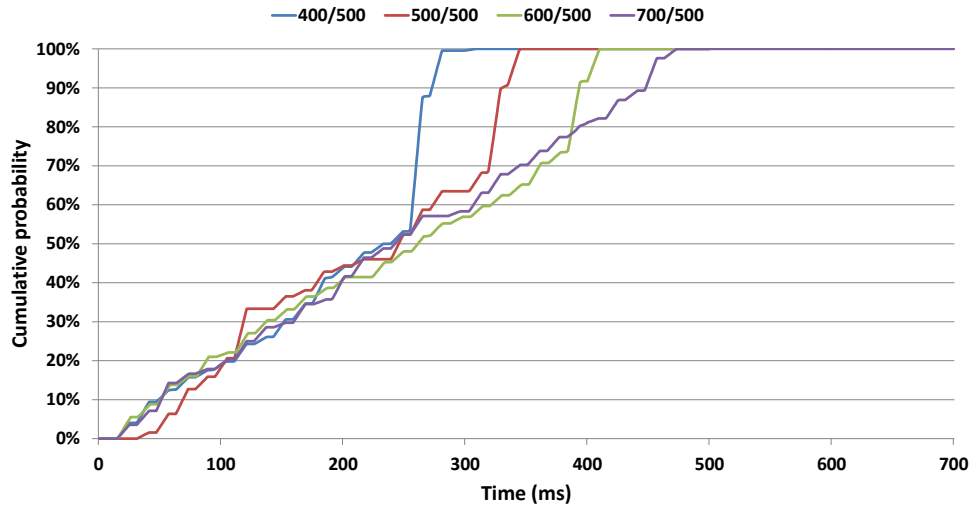


Figure 6.5: Solving hard failures duration when changing HBs. The times (ms) in the legend denote t_A/t_B as introduced in Chapter 5.

It ranges from 400/500 to 700/500, where the first number represents the amount of ms between HBs from the controller. The second number represents the timings for the actuator HBs. The data show some parts with a strong increase of solved hard failures. It can be concluded from the graph that theory matches experiment data better, when using less frequent controller HBs.

Chapter 7

Conclusions and future work

7.1 Conclusions

The Priva company wants to make the step from wired sensor and actuator networks towards wireless sensor and actuator networks (WSANs), while keeping reliability. This thesis used the following main research question to guide in this step: "How to make actuator applications wireless while maintaining reliability?". Soft and hard failures are classified to analyze reliability and robustness. A solution has been developed to fit the WSAN robustness requirements set by Priva. The following sub research questions are used to answer the main research question:

1. What are the general WSAN requirements?

The requirements of WSANs are mainly based on reliability and robustness. Failures in WSANs are critical for themselves and their surroundings. WSANs should deal with failures occurring from randomly dropping packets. Other WSAN constraints, which are introduced in Chapter 2 are on energy consumption (duty cycle between 1:140 and 1:85,000), network size (≤ 100 nodes), performance ($\leq 360,000$ messages per hour), memory (32 KB for code), multicast (required), network topologies (tree-like network), bandwidth (58.4 kbps for total network), and security (encryption and authentication).

2. What are the wireless link failure properties?

To identify the impact of failures, two types were classified in Chapter 3, which are soft failures and hard failures. Hard failures are classified as out-of-sync situations between the controller and the actuator, while soft failures are general failures to send setpoints (SPs).

The ratio between hard and soft failures was modelled by equations calculating the probability of each failure type, related to the packet arrival probability and the hopcount between controller and actuator. Experiments showed a

ratio between hard and soft failures ranging from 1:5 to 1:25 for single hops, depending on the link quality.

Interference was monitored to be able to indicate the causes of the failures. Experiments explained in Chapter 4 showed most failures occurring during office hours. Sun radiation, 2.4 GHz interference, and temperature influences were shown through experiment data. Most soft failures seem to be taking less than 500 ms in poor link conditions, showing that in practice most soft failures can be ignored.

3. How to make WSANs robust?

Based on failure analysis a robustness solution was proposed in Chapter 5 solving hard failures, and making the system able to cope with soft failures. The solution uses heartbeats (HBs) containing the SP and state, so the receiving side can change its SP or state to match the sending side whenever it did not. Further, the HBs provide a constant link quality status by monitoring failed HBs.

We analytically modelled the impact of HB frequencies on the time HBs take to solve hard failures. The model shows that the most frequent HB determines the behavior significantly more than the less frequent HB.

The solution was implemented, and experiments from Chapter 6 show the state ratios and the state transitions as a result of the failures. Experiment data also shows the way in which the state machine behaves and the time it takes to solve hard failures. The times gathered from experiments are similar to the ones calculated by the analytical model.

7.2 Future work

Many elements influence the experiments and solution benefits, where one of the most influencing is hopcount. Hopcount was taken into account when creating the analytical model, but not when doing the experiments. Some extra experiments concerning hopcount and the effect of the end-to-end acknowledgments in practice would complete the 802.15.4 analysis.

The solution uses HBs that occur periodically at fixed intervals. These intervals are expected to change the system properties and depend on the environment. A future solution could monitor the deployment environment to find the best intervals.

Finally, it could be possible to experiment with a variable HB interval, adapting to its environment on the fly. This goes further than the usual single quality scan, and would obviate expensive manual reconfiguration.

Bibliography

- [1] Co-existence of IEEE 802.15.4 at 2.4 GHz. Application Note, JN-AN-1079, Feb 2008.
- [2] M.K. Aguilera, W. Chen, and S. Toueg. Using the heartbeat failure detector for quiescent reliable communication and consensus in partitionable networks. *Theor. Comput. Sci.*, pages 3–30, Jun 1999.
- [3] R. Alena, R. Gilstrap, J. Baldwin, T. Stone, and P. Wilson. Fault tolerance in zigbee wireless sensor networks. In *Proc. AERO '11*, pages 1–15, 2011.
- [4] G. Anastasi, M. Conti, M. Di Francesco, and V. Neri. Reliability and energy efficiency in multi-hop ieee 802.15.4/zigbee wireless sensor networks. In *Proc. IEEE ISCC '10*, pages 336–341. IEEE Computer Society, 2010.
- [5] H. Benítez-Pérez and F. García-Nocetti. *Reconfigurable Distributed Control*. Springer, 2005.
- [6] M. Buettner, G.V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proc. SenSys '06*, pages 307–320. ACM, 2006.
- [7] A. Dunkels, F. Österlind, and Z. He. An adaptive communication architecture for wireless sensor networks. In *Proc. SenSys '07*, pages 335–349, Nov 2007.
- [8] A. Dunkels, T. Voigt, and J. Alonso. Making TCP/IP Viable for Wireless Sensor Networks. In *Proc. EWSN '04, work-in-progress session*, Jan 2004.
- [9] S. Duquennoy, N. Wirström, N. Tsiftes, and A. Dunkels. Leveraging IP for Sensor Network Deployment. In *Proc. of the workshop on IP+SN '11*, 2011.
- [10] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels. Making sensor networks ipv6 ready. In *Proc. SenSys '08*, pages 421–422. ACM, 2008.
- [11] V. Garg. *Wireless Communications & Networking*. Morgan Kaufmann Publishers Inc., 1st edition, 2007.
- [12] D. Gislason. *Zigbee Wireless Networking*. Newnes, Aug 2008.
- [13] R. Jurdak, P. Baldi, and C. Videira Lopes. Adaptive low power listening for wireless sensor networks. *IEEE Transactions on Mobile Computing*, pages 988–1004, Aug 2007.
- [14] P. Kinney, P. Jamieson, J. Gutierrez, and M. Naeve. IEEE 802.15 WPAN™(TG4). <http://www.ieee802.org/15/pub/TG4.html>, Mar 2012.
- [15] R. Koo and S. Toueg. Effects of message loss on the termination of distributed protocols. *Inf. Process. Lett.*, pages 181–188, Apr 1988.
- [16] J. Kooker. Bluetooth, ZigBee, and Wibree: A Comparison of WPAN Technologies. Nov 2008.
- [17] T. Lennvall, S. Svensson, and F. Hekland. A Comparison of WirelessHART and ZigBee for Industrial Applications. In *WFCS '08 IEEE*, pages 85–88, May 2008.

-
- [18] M. McInnis and T. Harrington. IEEE 802.15 WPAN™(TG4f). <http://www.ieee802.org/15/pub/TG4f.html>, May 2012.
 - [19] G. Mulligan. The 6lowpan architecture. In *Proc. EmNets '07*, pages 78–82. ACM, 2007.
 - [20] A. Nayak and I. Stojmenovic. *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*. Wiley, 2010.
 - [21] Y. Sankarasubramaniam, Ö.B.Akan, and I.F.Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proc. MobiHoc '03*, pages 177–188. ACM, 2003.
 - [22] Z. Shelby and C. Bormann. *6LoWPAN: The wireless embedded internet*. Wiley, Jan 2010.
 - [23] Y. Tian, K. Xu, and N. Ansari. TCP in Wireless Environments: Problems and Solutions. In *Communications Magazine, IEEE*, pages S27–S32, Mar 2005.
 - [24] C. Wan, A.T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *Proc. WSNA '02*, pages 1–11, 2002.
 - [25] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen. TCP Performance Issues over Wireless Links. *Communications Magazine, IEEE*, pages 52–58, Apr 2001.
 - [26] M.Z. Zamalloa and B. Krishnamacharis. Analyzing the transitional region in low power wireless links. In *Proc. IEEE SECON '04*, pages 517–526, 2004.