Master's thesis

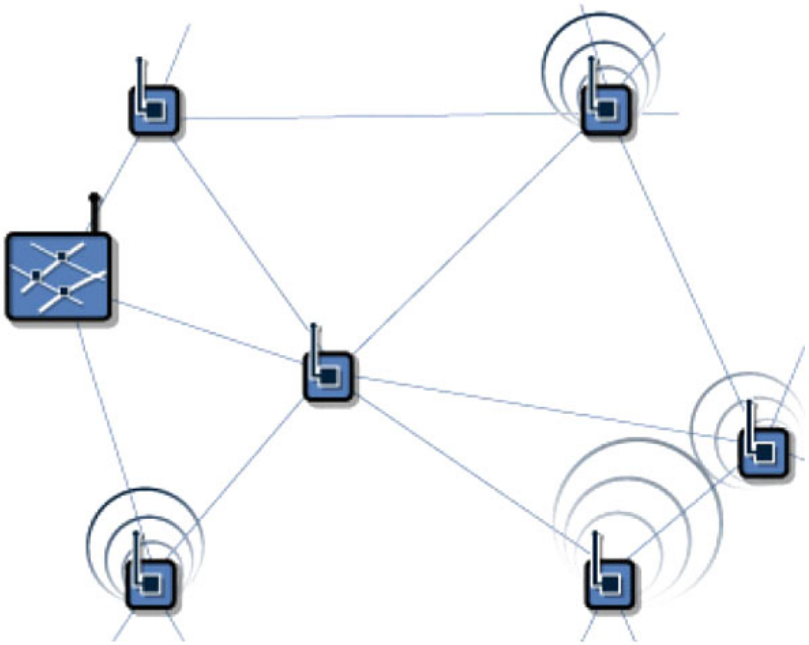
Nils Petter Eftedal

Evaluation of SmartMesh-XR

Wireless Multihop Network

Trondheim June 2006

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics



NTNU
Innovation and Creativity

Nils Petter Eftedal

Evaluation of SmartMesh-XR

Wireless Multihop Network

Master's thesis

Trondheim, June 2006

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

Academic supervisor: Tor Onshus

NTNU
Innovation and Creativity

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

MASTEROPPGAVE

Kandidatens navn: Nils Petter Eftedal

Fag: Teknisk Kybernetikk

Oppgavens tittel (norsk): Evaluering av SmartMesh-XR - trådløst multihopp nettverk

Oppgavens tittel (engelsk): Evaluation of SmartMesh-XR - wireless multihop network

Oppgavens tekst:

New technologies have made wireless communication more suitable for industrial environments. The new features comprise low power consumption, resulting in long-life battery powered nodes, and alternative routing abilities (mesh topology). Another property for the new standards is relaxed data rate requirements, usually around 250 kbps. Typical deployments are in networks with sensor nodes performing various measurements in industrial processes. Today, communication between such nodes and the processing central is based completely on wired solutions. Making this communication wireless will reduce installation and maintenance costs and provide more flexible networks.

A few networks of this kind are currently being considered as a basis for the new wireless HART standard, which will be presented shortly (sometime during the coming summer). One of the networks is called SmartMesh-XR, a wireless multihop product from Dust Networks. Interesting technology features like frequency hopping and mesh topology gives this network advantages over older technologies.

The main task of this project will be to do an evaluation of the SmartMesh Network. Different configurations will be tested, and performance metrics like latency times, reliability, and power consumption will be collected. The work will be carried out in cooperation with Statoil and ABB. Most of the work will be done using a Dust SmartMesh Evaluation Kit.

The key tasks for this project will be:

- A study of wireless sensor network theory.
- Get familiar with the SmartMesh product and key concepts.
- Installation of a test network at ABB, followed by performance tests.
- Installation of a network for collection of real "in process" data at Statoil. The results will be compared with a test performed with ZigBee.
- Based on the study of theory and the various tests, make an assessment on the use of SmartMesh-XR as a basis for wireless HART.

Oppgaven gitt: 9. januar 2006

Besvarelsen leveres: 6. juni 2006

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Veileder: Knut-Olav Fjell (Statoil) og Niels D. Aakvaag (ABB)

Trondheim, den 09.01.2006

Tor Onshus
Faglærer

Preface

This report documents the work of a master thesis at the Norwegian University of Science and Technology in Trondheim, department of engineering cybernetics. The project has been performed in co-operation with the Norwegian ABB Corporate Research Center in Oslo and the Statoil Research Center in Trondheim. The work has been carried out at both locations.

The thesis is a result of a continued co-operation from a previous project where ZigBee was tested for wireless transmission of HART. The conclusion back then was that ZigBee did not provide the necessary determinism required in large and dense networks. In the current thesis, a new wireless sensor network is evaluated, the SmartMesh-XR. The network is provided by Dust Network and comprises features like frequency hopping, support for mesh topology and timed- and synchronized communication with determinism for all network nodes. An assessment regarding wireless HART is a part of this project as well, although the main emphasize is on performance and properties in general.

There are several persons that have contributed to make this project both interesting and exciting. First I would like to thank PhD Niels D. Aakvaag at ABB Corporate Research Center who made this project possible in the first place, and for being the provider of the equipment used during the evaluation experiments. His guidance throughout the project has been priceless. I would also like to thank Knut-Olav Fjell, principal researcher at Statoil Research Center, who made all the administrative arrangements for the industrial experiment.

Special thanks are due to Kavindra Saxena, Director of Application Engineering at Dust Networks, who has provided detailed information throughout the experiments. Without the knowledge derived from our discussions, a lot of guessing would have been involved with the performance analysis and about the protocol in general. Finally, I would like to thank professor Tor Onshus, representing the department of engineering cybernetics, for constructive discussions and regular follow-ups.

Nils Petter Eftedal

Trondheim, June 2006.

Abstract

The IEEE 802.15.4 standard for LR-WPANs (Low-Rate Wireless Personal Area Networks) provides a low cost and low powered foundation which is utilized in a range of wireless sensor networks (WSNs) today. However, to obtain a fully deterministic network with guaranteed communication for all network nodes, the upper protocol layer (MAC) of the standard needs an upgrade. Improved versions of the MAC layer also involves features like frequency hopping to make more reliable and robust networks in the presence of interference and multipath fading. The increased reliability that is achieved through these upgrades makes the WSNs of today even more interesting for industrial applications than ever.

SmartMesh-XR is one of the new WSNs that has gained much attention due to properties like frequency hopping, mesh topology and timed and synchronized communication (TDMA), providing determinism and low power consumption. In co-operation with the Norwegian ABB Corporate Research Center and Statoil's Research Center at Rotvoll, a master's thesis has been formed that focus on the properties and performance of the SmartMesh network.

This report comprises both theoretical protocol analysis and physical performance tests. The physical experiments were carried out with a SmartMesh evaluation kit bought from Dust Networks - the provider and developer of the technology.

Two different test networks were installed in this project; one in the office environment of the ABB facility and one in the LAB environment at Statoil. The office experiments tested network performance in the presence of WiFi jamming and multipath fading due to human traffic, while the Statoil environment presents challenges like rotating equipment, vibrating engines and metal constructions etc. To make the industrial experiment at Statoil as realistic as possible, data was collected from real sensors in the environment. The external sensors required signal conditioning and could not be directly connected to the nodes. For this purpose there was made three interface cards - one for each external sensor. Further, it was stressed to install the network with nodes placed at identical locations as a previous experiment performed with ZigBee. The results are compared in this report.

Important performance metrics for all experiments include reliability, latency and path stability. These metrics have been collected in different topologies and with different configuration parameters, like frame lengths and report rates, to cover all aspects and scenarios where the network may be used. When the results from the two environments are compared, the path stability graphs show that re-transmissions occur more rapidly in the office facility than in the industrial site at Statoil. This indicates that RF-interference and multipath fading is even more challenging than RF-barriers like metal constructions and vibrating engines etc. The fact that the path stability decrease about 5% at daytime, when more people are in the hallways and the WiFi traffic is likely to increase, further backs up this theory.

In addition to the physical experiments, it is the aim of this report to provide the reader with a theoretical background to better understand the trade-offs involved with WSN protocols. This includes network topologies, RF Challenges and medium access approaches in addition to general requirements. There are also chapters containing detailed analysis of both the SmartMesh protocol and two of its competitors; Wavenis and SensiNet.

Based on the experiments and the protocol analysis, an assessment is made regarding the use of SmartMesh-XR as a wireless extension of HART. It has been shown that the SmartMesh network is a good alternative for wireless HART as long as its applications can tolerate additional latency and reduced report rates - both in the order of seconds.

Conclusions

SmartMesh-XR is one of the new protocols that improve the IEEE 802.15.4 standard with regard to increased reliability. Both the experiments and the protocol analysis of this project have shown that the SmartMesh Network is highly reliable, in the order of a few lost packets per 100000, as long as network configuration is carried out properly. This includes a frame length that complies with network size and topology, and the actual number of reporting nodes in the network. A suitable report rate must be chosen based on the frame length and available bandwidth. Bandwidth is here referred to as the number of assigned links for a specific mote.

To maintain full reliability in the presence of node failures or blocked signals it is critical that the network is connected in a full mesh topology. During the experiments it was shown that to achieve such topology, it is necessary that the network is installed at a time where the worst signal conditions are expected. The measured received signal strength between two motes sometimes varied with more than -20dBm, and even though a mote had multiple parents at the time of installation, this was not necessarily the case a few hours later.

The latency and power consumption of the network nodes are strongly connected and a matter of configuration. They are both affected by frame length and the network topology, but while latency is increased by a reduced frame, power consumption is increased. To handle this trade-off, the results of this report show that it might be necessary to provide certain motes with wired power. That is, if low latency is a requirement and assuming that the network is formed in a topology with multiple hops. A multihop network results in increased latency for each necessary hop from a node to the Manager and increased power consumption for nodes that have to forward data from other motes in addition to its own data.

Both the measured latency and power consumption of SmartMesh-XR were slightly higher than expected. To add some numbers to this statement; a connected 6 mote multihop/linear network had an average latency above 2.5 seconds for the leaf mote and an average power consumption about 1 mA for the mote closest to the Manager. The frame length during this experiment were set to 1750 ms resulting in a minimum (reliable) report interval of about 5 seconds. If the frame length is halved, so will latency and minimum report rates, but the power consumption will be close to the double. Hence, wired power may be required.

When the results from the industrial experiment of this project are compared to the identical experiment with ZigBee, some of the main differences between TDMA- and CSMA approaches are revealed. The latency of SmartMesh-XR is ten times higher than that of ZigBee, but in turn it achieves a much higher throughput assuming that no lost packets are allowed. In addition, ZigBee was configured without beacons and thus burn a lot more power than the TDMA based communication of SmartMesh-XR. Further, the differences between

SmartMesh-XR and ZigBee are likely to increase in dense networks with a large number of nodes. While ZigBee has to struggle with the hidden- and exposed terminal problems, all communication in SmartMesh-XR is scheduled prior to communication

Discussions with Dust Networks have revealed that they are planning to halve their current slot size and enable the multiple frame feature of their protocol for the end user. The former will reduce latency with about 50% while the frame feature will make it possible to run multiple frames simultaneously, allowing certain tasks more rapid reporting and less latency.

Based on the protocol analysis and the experiments of this project it has been discussed whether or not SmartMesh-XR is a good foundation for wireless HART. Even though its competitors have not been physically tested, the protocol analysis gives no reason to believe that any of these will perform better than SmartMesh-XR. On the contrary, if the new version of the SmartMesh protocol contains the planned upgrades, it might have an edge on its competitors. The final conclusion is therefore that if the limitations for latency and report rates can be accepted, SmartMesh-XR is a good and reliable alternative for wireless HART.

Contents

List of Figures

List of Tables

Definitions

Link	Links are the assigned time slots and frequencies at which two motes transmit information to each other. Manager assigns links as soon as motes establish a path. [15]
Manager	The line powered network node (gateway) that controls and monitors network performance. Manager coordinates routing, aggregates mote packets, collects network statistics, and uses an Ethernet port to publish data in XML format to a wired network. [15]
Mesh network	A network in which the routing of messages is performed as a decentralized, cooperative process involving many peer devices routing on each others' behalf. [31]
Mote/node	Ultra low-power wireless transceivers that digitize data from attached sensors and use an onboard radio to send the packets to neighbouring motes. These motes pass the packet on to other motes - and, in a series of "hops" deliver the data to SmartMesh Manager. [15]
Multihop network	A network, in particular a wireless network, in which there is no guarantee that the transmitter and the receiver of a given message are connected or linked to each other. This implies that intermediate devices must be used as routers. [31]
Path	A path forms between two motes when the motes have discovered that they are within radio communication range of each other. [15]
Route	A route is a set of paths between a mote and Manager. [15]
x1, x2, x3...	Used for important configuration parameters in this report, like frame lengths and report intervals. When a report interval is defined, it refers to the frame length multiplied with an integer value (e.g. frame length x 2 - x2). The frame length is calculated by the number of motes in the network multiplied with an integer value (e.g. number of motes x 3 - x3).

Acronyms and Abbreviations

ACK	Acknowledge (packet)
API	Application Programming Interface
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CRC	Cyclic Redundancy Check
CTS	Clear To Send
DLL	Data Link Layer
DSSS	Direct Sequence Spread Spectrum
FEC	Forward Error Correction
FFD	Full Functional Device
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In First Out
GTS	Guaranteed Time Slot
HART	Highway Addressable Remote Transducer Protocol
HCF	HART Communication Foundation
IEEE	Institute of Electrical and Electronics Engineers
IZT	Industrial ZigBee Test
LLC	Logical Link Control
LQI	Link Quality Indication
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Medium Access Control
NACK	Negative Acknowledge (packet)
NWK	Network Layer
PAN	Personal Area Network
PHY	Physical Layer
POS	Personal Operating Space
QoS	Quality of Service
RF	Radio Frequency
RPC	Remote Procedure Call
RSSI	Received Signal Strength Indicator
RTS	Request To Send
TDMA	Time Division Multiple Access
SLA	Service Level Agreement
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Background

Wireless sensor networks consist of a large number of sensor nodes that may be randomly placed without the need of structure and wires of any kind. Sensor nodes are small devices with processing capabilities that may be attached to sensors in the environment, or contain sensors of their own. These devices then communicate this data, typically to a centralized unit or to each other along a route to this unit. The advantages of such networks are increased mobility, reduced installation and maintenance costs, and increased range of use.

Wireless communication still includes the same issues and challenges it did several years ago, but with new technology and smart solutions the risk factor is now at an acceptable level for many applications. Wireless sensor networking is an area that has gained much attention due to these new advances. There is no doubt that the cost savings and opportunities presented by such networks could greatly benefit industrial automation and monitoring, either as an extension of an existing protocol, or in the form of a completely new standard.

The HART protocol is one of the leading communication technologies within the area of smart instrumentation today. In order to keep up with an increasing number of member companies, the HART Communication Foundation (HCF) has decided that an upgrade of the protocol is required. The new version has a planned release date this summer, and will comprise a standard for wireless HART as well as an upgrade of the wired protocol. Three different sensor network solutions are currently considered for the wireless part of the protocol; the SensiNet protocol from Sensicast, Wavenis from Coronis and the SmartMesh protocol from Dust Networks.

In a previous project performed in co-operation with ABB and Statoil, ZigBee was tested for wireless transmission of HART. The conclusion was then that the ZigBee protocol, with guaranteed communication for only seven nodes, did not fulfil the requirements for a standard like wireless HART. With the current project, this co-operation is extended, but instead of ZigBee it is now the SmartMesh protocol that is evaluated.

1.2 Purpose

Wireless sensor networks are a hot topic of research for both the ABB Corporate Research Center AS and Statoil's Research Center at Rotvoll. When it comes to HART, the upgrade of the protocol is of particular interest for ABB, which is a provider of HART devices and equipment, and a member of both HCF and the wireless HART group. Thus, the report will look into the capabilities of the SmartMesh network as a instrumentation tool in general, and not exclusively focus on wireless HART.

The experiments in this project comprise both an office facility and an industrial environment where the nodes are placed in a range of locations, like within explosion safe cells and on rotating devices. In addition to the physical experiments there will also be a theoretical approach to the evaluation. The main emphasis is on the SmartMesh protocol, but all the other protocols considered for wireless HART are examined as well. Hopefully, the combined result of theory and experiments will reveal the limitations and use of the new technology.

1.3 Report Structure

The report starts out with a chapter providing the necessary theory to understand the features and properties of the different protocols. This knowledge will come in handy while reading the following two chapters, comprising detailed information about both the SmartMesh Network and its competitors. Next, in chapter 5, the background and objectives of the experiments are discussed followed by a chapter describing the experimental installation and implementation. Chapter 7 explains the most important performance metrics emphasized in the experiments, which will be frequently used in the following chapters. These chapters describe the configurations and results of the performed experiments. Finally, a discussion is given about the various results and the presented theory.

Chapter 2

Wireless Sensor Network Theory

This chapter introduces requirements and issues of Wireless Sensor Networks (WSNs). Important trade-offs are discussed, emphasizing those concepts that are of particular interest for the present project. In addition to usage and design considerations of WSNs, there will also be sections covering RF challenges and solutions. Since power consumption is one of the major issues of WSNs, one section is devoted to the topic of energy scavenging.

2.1 Introduction

Over the last few years, the development of networks comprising low-cost, low power, multifunctional sensors has received increasingly attention. Smart sensors have existed for many years, but never before with the ability to communicate with each other the way they can today. Advances in wireless communications and electronics have made it possible to make sensors smaller in size than ever, and provide features that ensure reliable transfers over radio frequency channels. A smart sensor (node) has the ability to sense and process data, until finally transmitting this data to its destination or a different node on the route to the destination. The basic features of sensor networks are [1]:

- Self-organizing capabilities
- Short-range broadcast communication and multihop routing
- Frequently changing topology due to fading and node failures
- Limitations in energy, transmit power, and computing power

These characteristics, particularly the last three, make WSNs different from other wireless ad hoc or mesh networks. The use and properties of such networks will be discussed in the following sections.

2.2 Usage of Wireless Sensor Networks

WSNs may be used in a wide variety of applications. Today, they are mostly used in monitoring tasks, providing flexible and low cost data collection without the need of wires and expensive network maintenance. The range of use is, however, beyond the scope of monitoring

even now. With today's technology it is possible to meet the requirements of less demanding control and actuation tasks. Still, it is understandable that industry awaits a fully tested and established product. Downtime and deviations from the optimum set-point may lead to unwanted situations at high cost. Hopefully, this evaluation report will make it clear what to expect from the current technology and reveal its limitations and downsides.

In their book about protocols and architectures for WSNs [2], Karl and Willig list some of the most relevant application types for a WSN. In addition to pure monitoring or periodic measurements which are most commonly used, we find applications like event detection and tracking. Event detection is often used in combination with periodic measurements to report specified events, like high temperatures, critical failures etc. Tracking applications involve mobile nodes that report data every time they detect a neighbouring node. The data may involve position and time, and may potentially be reported to the sink (gateway) with estimates about speed and direction. To make these estimates, sensor nodes typically have to cooperate before updates can be reported. Such application may greatly benefit wildlife researchers and be helpful in both military and security scenarios. As a final application type, Karl and Willig [2] mention function approximation and edge detection. A good example is found in the scenario of finding the isothermal points in a forest fire application to detect the border of the actual fire.

2.3 WSN Requirements

In contrast to wireless business networks, like WiFi (IEEE 802.11), WSNs do not have their main focus on high throughput and low latency. Instead, they often emphasize low power consumption, low-cost nodes and support for various network topologies. Since WSNs have to prove that they can compete with wired sensor networks, reliability is of the essence. The current generation of WSNs provides features like frequency hopping, guaranteed timeslots and mesh topology. To cover large areas, data and control packets are usually forwarded in a multi-hop fashion, sent from mote to mote until reaching the sink node. The requirements mentioned in this section involve hard trade-offs that have to be taken into consideration. A discussion of these and other important requirements are carried out in the following sections. Each section represents important issues the reader should bear in mind throughout the rest of the report.

2.3.1 Low Power Consumption

The primary strength of WSNs is their independence of wiring costs and constraints. As the experiments in this report will show, rotating wheels and moving parts no longer has to be deployed with slip rings, exposed to wear and tear and signal noise. A simple installation of a battery-powered node is sufficient. Another good example to where wireless nodes come in handy, is at remote locations that are inconvenient to reach. However, if the batteries must be changed often, like every week or mount, not only will the initial cost savings be lost, but it will also make the network inadequate for many remote sensing applications. No monitoring or control can be performed while batteries are change, and as mentioned earlier, such downtime may lead to loss of both time and money. For instance, a complete system may need to shut down, if batteries must be changed on a mote attached to a rotating wheel. Therefore, long battery life is essential and preferable in the order of years

Adaptive Transmission Power and Adaptive Bit Rates

There is no doubt that low power consumption is a must if no wired power supply is provided, making this a research topic of great interest. Lately, much attention has been given to adaptive transmission power and adaptive bit rates. The former has been discussed by Banerjee and Misra in their article [3]. The basic idea is to consider the impact of receiver noise on packet errors, and adjust the transmission power to minimize the total energy spent in reliably forwarding a single bit. There is more than one way to deal with this, but further discussion on this topic is beyond the scope of this project. Adaptive transmission rate methods utilize the fact that lower bit rates results in lower power consumption. For this reason, the bit rate of most WSNs is in the order of kbps, and not Mbps as for many field buses and business networks.

Duty cycling

The best way to ensure low power consumption is to put nodes into sleep mode when they are not communicating. More power is saved, the longer period a node spends in sleep mode. To achieve such duty cycling, some kind of synchronization is needed between the nodes. Synchronization adds complexity and overhead to the network, but as will be shown in section 2.7, it can also be used to avoid collisions and thereby increase reliability. In many networks utilizing synchronization, the solution has been either to let parent nodes periodically update its children with the correct time, or by letting children nodes periodically ask their parent(s) for time updates. Examples of both methods are given in the following chapters, where different protocols are analyzed.

In his book [4], Callaway describes a common way to calculate the average power consumption of a single node in a WSN. This formula is illustrated in expression 2.1.

$$P = \alpha \cdot P0 + (1 - \alpha) \cdot Ps \tag{2.1}$$

In the formula, P is used in terms of average power consumption, while Ps and P0 indicate power consumption in sleep mode and with active transceiver respectively. The small alpha, α, represents the duty-cycle, which is the fraction of time that the transceiver is active. Since P0 is much greater than Ps, it is obvious that a low duty-cycle (α) will reduce the average power consumption considerably. Both the SmartMesh protocol and its competitors support this scheme independent of topology.

Alternative Power Sources

Over the years, much attention has been given to alternative power sources. Even though this has always been a topic of research, the development of WSNs has given it a focus and goal that makes the demand for ubiquitous energy sources even greater than ever. In the near future we are likely to see sensor nodes powered by magnetic fields, vibrations or perhaps the well-known solar energy. With an infinite amount of energy, duty-cycling may be reduced or perhaps eliminated altogether. This will make WSNs appropriate for a whole new range of applications. More details about power scavenging and alternative power sources are given in section 2.5.

2.3.2 High Reliability

One of the most important requirements of a WSN is its ability to maintain a reliable network while keeping performance on an acceptable level. Wireless communication has a lot of issues which are not present in traditional wired networks. The RF environment is constantly changing, and nodes may be exposed to interference from other wireless sources, like office WLANs and cordless phones. Features to handle RF challenges are discussed in section 2.6, while other reliability mechanisms are covered in the present section.

Quality of Service

Quality of service (QoS) refers to the capability of a network to deliver data reliable and timely. Depending on the applications they are running, an important issue in many WSNs is that of QoS guarantees. In a RF environment it might take a long time for a packet to reach its destination, and the exact latency (delay) is hard to predict. This is because a packet may get held up in long queues, may need to be retransmitted, or takes a less direct route to avoid congestion. On the other hand, the packet may follow a fast direct route and thus get really low latency. In many WSN solutions it is possible to set guarantees or thresholds for latency, reliability and other performance metrics. Such agreement is usually termed as an SLA (Service Level Agreement). The idea is to take action if a threshold is breached and by this maintain the performance/reliability of the system.

Fault Tolerance

There is always a chance that nodes may run out of energy, get damaged, or that communication between two nodes gets permanently interrupted/blocked. Thus, it is important that the WSN as a whole is able to tolerate such faults. To tolerate node failure, redundant deployment is necessary, using more nodes than would be strictly necessary if all nodes functioned correctly. The importance of this point will become clear as we move on to the experiment chapter of this report.

Security

Reliability involves more than just the reception of a message, it is perhaps even more important that the received message has not been tampered with in any way and is from the sender it purports to be. In many applications, like alarm or control systems, security is critical. Networks with such applications should enable intrusion detection and tolerance as well as robust operation in the presence of failure, because in most cases, the sensor nodes are not protected against physical mishandling or attacks. Eavesdropping, jamming, and listen-and-retransmit attacks can hamper or prevent the operation. For this reason, access control, message integrity, and confidentiality must be guaranteed.

2.3.3 Scalability and Flexibility

Since a WSN might include a large number of nodes, scalability is a critical factor that guarantees that the network performance does not change significantly as the network size (or node density) increases. Another important factor is that of flexibility, in the presence of environmental changes the network should adapt to the new environment to maintain the best performance possible.

2.3.4 Support Different Network Topologies

For many applications it may be sufficient with a conventional star network, employing a single master and one or more slave devices. The drawback with such approach is of course that of limited area coverage. Network range can be extended to a certain point by turning up the transmit power, but this will greatly increase the power consumption of the nodes. The best solution to the range-problem is to employ a network that supports multi-hop routing, with topologies like mesh or cluster types. Such topologies increase the complexity of the network, involving routing algorithm and tables, additional memory requirements, network maintenance, etc. The added complexity must be supported without excessive cost or power consumption, introducing new trade-offs and possible implementations.

2.3.5 Relaxed Latency and Throughput Requirements

The trade-off between power consumption and the data rates was explained in section 2.3.1. To maintain long battery-life and thus long-lasting nodes, WSNs have relaxed throughput requirements compared to other wireless standards like Bluetooth or WiFi. The typical data throughput for many applications may be as low as 1 bit/s or even lower [4]. It should be mentioned that the actual raw data transmitted on the channel, may be sent at a much higher rate than the throughput indicates. The IEEE 802.15.4 standard [10] offers a data rate of 250 kb/s in the 2.4 GHz band including both payload and overhead.

The throughput limitations and the liberal QoS requirements, make WSNs unsuitable for transmission of real-time video and audio. As a direct consequence it also reduces the latency requirements of the network, but this will vary for different protocols and depend on configuration and topology. In many applications, a latency of seconds or minutes is quite acceptable.

2.4 Network Topologies

Different topologies strongly affect the properties and performance of a WSN. Hence, it is important to install a network with a few trade-offs in mind. The last two topologies presented in this section, star mesh and full mesh, is the ones most appropriate to form a reliable widespread network. During the writing of this report it is debated which design is currently the best. Some companies, like Sensicast, claim that the best way to connect a network is by letting wire powered nodes form the backbone of the network, while battery-powered devices may connect these nodes in the form of leaf nodes. The idea is to increase the range of the backbone nodes by increasing the transmit power, and by doing so reduce the number of routers in the network. In contrast, the full mesh topology allows all nodes to be equally designed with battery-power[1] and routing abilities. The details will be more thoroughly explained in the following sections.

2.4.1 Star

The star topology provides a fast and reliable way to collect data. All nodes in the network are placed only on hop away from the gateway (sink node). This reduces latency and makes the

[1] When wired power is available, this is preferable over batteries.

network more reliable in the sense that a faulty or blocked node will not affect the performance of the other network nodes. The fact that all nodes have only one path to the gateway makes them more vulnerable to environmental changes, no alternative routing is possible. Another drawback is that of area coverage. Because the transmit power of the nodes is limited by government regulations and battery life concerns, network range can only be extended to a certain extent. The range of the gateway node restricts the coverage area of the entire network.

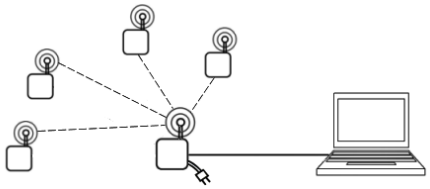


Figure 2.1: Star topology

2.4.2 Tree

When a network is connected in a tree topology it may cover a large area. The major disadvantage with the topology is that a faulty or blocked node will result in the loss of all nodes connected to that specific node. In the worst case scenario, a node connecting a big sub-network is lost, resulting in the loss of a major part of the network. It should also be noted that in a multihop topology such as this, latency increases with the number of hops.

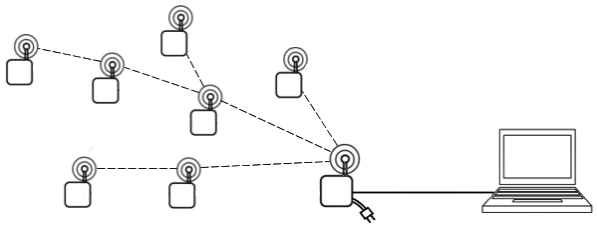


Figure 2.2: Tree topology

2.4.3 Star Mesh

The star-mesh topology comprises two different component types, Full Functional Devices (FFD) and Reduced Functional Devices (RFD), the same components that are described in the IEEE 802.15.4 standard. In the scenario with the current topology, the FFDs are typically powered by wire and have higher transmit power than the RFDs. The idea is to let FFDs form the backbone of the network, enabling coverage over a large area with as few devices as possible. Only FFDs implements routing abilities. The reasoning behind this is to allow the battery-powered leaf-nodes to save power, while wire powered FFDs handles the more complex tasks like routing, etc.

Calling this topology star mesh may be to underestimate its capabilities, because if the FFDs are installed at good thought-through locations, the leaf nodes may indeed connect more than one FFD achieving a full mesh topology. However, this may involve additional devices resulting in a more expensive network. Sensicast is one of the providers of such network structure, more information about their SensiNet protocol is presented in section 4.2.

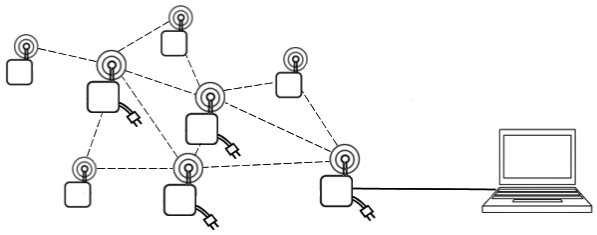


Figure 2.3: Star-Mesh Topology

2.4.4 Full Mesh

A full mesh topology is the most reliable topology there is. To achieve such topology, all nodes need at lest two parents (upstream nodes or nodes closer to the gateway). When the network is connected this way, data can always be routed on alternative routes if a node is blocked, failed or exposed to congestion (full queue). The cost of this extra reliability is increased complexity resulting in increased power consumption. Different trade-offs are exploited in the process to keep long-lasting battery-powered nodes. This may include power saving schemes like reduced data rates, low throughput, reduced transmit power, low duty cycle, etc. Various protocols handle this in different ways; chapter 3 will look into the implementation of Dust Networks.

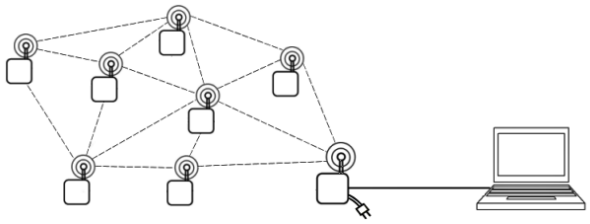


Figure 2.4: Full Mesh topology

2.5 Energy Scavenging

In the dictionary [5], energy scavenging (sometimes termed power harvesting) is defined as the process of acquiring energy from the surrounding environment and converting it into usable electrical energy. This concept is of special interest for WSNs, where it may be used to ensure truly long-lasting nodes. Karl and Willig [2] summarize several approaches that may be used to extend battery-life (by battery charging) or eliminate batteries altogether;

The first approach mentioned is photovoltaics, describing both the well-known technology of solar cells and the transformation from solar cells to usable electricity. The success of this approach depends on whether nodes are used outdoors or indoors and the light intensity at the time. According to Karl and Willig, the resulting power is about 10 μW/cm^2 indoors and 15 mW/cm^2 outdoors. Single cells achieve a fairly stable output voltage of 0.6 V. Since solar cells strongly depend on location and light intensity they are most suitable for recharging of secondary batteries.

Utilization of temperature gradients is another interesting approach that may be exploited. Differences in temperature can be directly converted to electrical energy and even temperature changes as small as 5 Kelvin can produce considerable power. In one example it is claimed that a seebeck effect-based thermoelectric generator may achieve 80 μW/cm^2 at about 1 V from a 5 Kelvin temperature difference.

Other approaches make use of pressure variations and the flow of air/liquid. The former is exploited by the use of piezoelectric generators, which is in fact already in use. As an example, such generator has been placed in the heel of a shoe, to generate power as a human walk about. This device can produce, on average, 330 μW/cm^2. The flow of air/liquid includes miniaturization of the technology used in wind mills or turbines. This approach has so far not produced any notable results[2].

The use of magnetic fields as a power source was left out in the summary of Karl and Willig. This approach came to mind while reading the evaluation report [6] of a former master student, Erik Undheim. He predicted that it is just a matter of time until we have sensor nodes powered by sources like magnetic fields. Today, we are even closer to this goal. The technology exists and may be used, albeit it has been hard to find industrial examples where such solutions currently exist. Other alternative power sources, like vibration powered generators, have also emerged and are already in use. The design and performance of such generator has been presented by the Electronic Systems Design Group [7].

Clearly, it would be to great benefit for WSNs if these ubiquitous energy sources could be used to eliminate batteries altogether. As it stands today, they are considered too expensive to economically benefit a complete network. According to an article in the EE Times [8], even in military systems where the economic factor is removed, it is still difficult to get electronic systems to run on ambient-energy sources exclusively. This statement was backed up by the founder of Dust Networks, Kris Pister, which was interviewed in the same article. He believes that with the technology of today we are still better off with batteries and power saving schemes like deep duty cycling.

2.6 RF Challenges and Solutions

As mentioned earlier, wireless communication has to deal with a lot of issues that aren't present in traditional wired networks. The RF environment is constantly changing, involving multipath fading and various forms of interference. An interesting article [9] published by IWWIA (International Workshop on Wireless and Industrial Automation) studied the effect of such phenomena; some of the results are presented in the following sections.

2.6.1 The RF Environment

Multipath Fading

One of the greatest challenges for wireless communication is that of multipath fading. Multipath fading is often experienced as ”mysterious” behaviour, like a stationary radio that seems to be working perfectly for a while and then stops working for a period of time. This can be termed as time variant multipath, a phenomenon that can make systems extremely difficult to install, as a link that seems to be good at one time can become unreliable seemingly at random. The multipath profile can change every few seconds in environments that are used by people, or every few hours with the movement of objects such as vehicles, equipment, doors, and chairs. In locations with minimal human activity it is possible find a different form of multipath, termed static multipath. Since even tiny changes in the environment may change the multipath profile, such phenomenon is rare. But when it does occur, it is due to destructive interference causing radio nulls (cancellation effect).

The graph in figure 2.5 illustrates channel fading in the 2.4 GHz band, measured in a typical industrial environment [9]. To avoid confusion; the dark points represent the actual measurement whereas the lighter points represent the fading as averaged over a 2MHz bandwidth. Note that the graph shows frequency dependent fading in the order of 15 dB. This alone indicates that link quality could be substantially improved by the use of a protocol utilizing multiple frequencies.

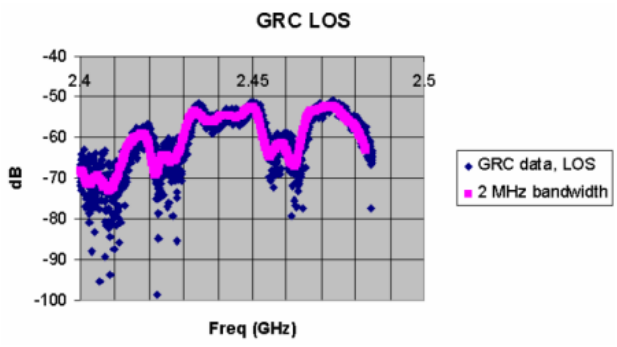


Figure 2.5: Channel fading; line of sight channel [9]

Interference

Interference is a well-known phenomenon, experienced by most people at some time. A good example is that of online gaming. Players that are using WiFi to connect to the internet may experience reduced performance if they have a cordless phone (2.4 GHz) in their house. In the worst case scenario, a WSN will be jammed by both of these sources. Such interference is termed time variant interference, also including interference from Bluetooth and other wireless devices, usually bursty by nature. These devices are not limited by requirements in the same way as WSN nodes, and they usually have a much larger transmit power than in most WSNs. The ratio of sizes is illustrated in figure 2.6, where a typical WiFi interferer is compared to the 16 channels of the 2.4GHz band as utilized by an IEEE 802.15.4 radio. To add numbers

to the comparison, the blue peaks represent an output power of 1 mW (0 dBm), which is typical for wireless sensor nodes. The red peak represents a typical WiFi device, usually with an output power of about 100 mW (20 dBm).

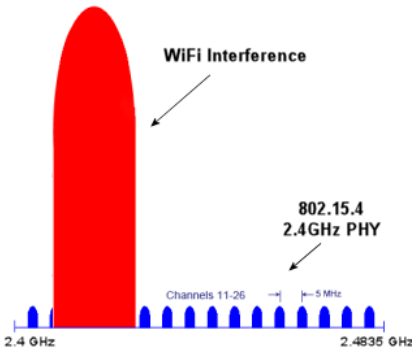


Figure 2.6: WiFi interference [12]

Examination of figure 2.6 shows that a WiFi interferer may take out several channels in a WSN. Hence, a protocol utilizing static channels is not a good choice in an environment where it coexists with other wireless devices. Configuring the network with the "wrong" channel will result in poor performance, especially when the WiFi traffic is high. A solution to the problem is to do environmental analysis prior to configuration, but there is no guarantee that the environment will stay this way forever. Again, a protocol utilizing multiple channels seems to be the best way to improve the overall link quality.

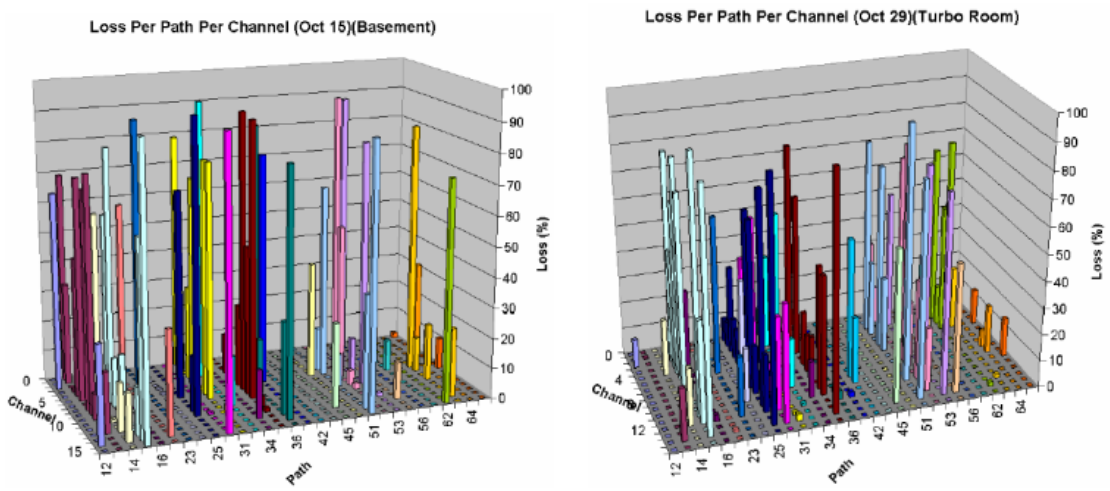
In addition to time variant interference we have interference in the environment, which may completely block one or all channels for a period of time. This is termed static interference, involving microwave ovens, RFID interrogators etc.

Link Performance Example

For further illustration of multipath and interference, an example from a real industrial experiment [9] is provided. The experiment involved six devices equipped with IEEE 802.15.4 radios and single board computers with local data storage. These devices were placed at locations in industrial facilities where wireless sensors for equipment monitoring would typically be placed. Communication was performed with one device transmitting while the remainder were listening and recording. Transmission was then cycled to a different device where the same procedure was repeated. To avoid packet collisions, the devices were synchronized in time. For more details about the experiment, see [9].

The graphs in figure 2.7 plot the packet loss rate versus path and channel in two different environments. Paths between all devices are included, with numbers indicating the sending and receiving device. This included both devices within line of sight as well as devices with non line of sight. The graph in figure 2.7(a) was retrieved in a machine room, an isolated area with little to no motion (static multipath). Examination of the graph shows that none of the channels allowed reliable communication over all paths for all devices throughout the entire test period (4 hours). In the graph in figure 2.7(b), retrieved in an industrial gas compression

facility, only channel 15 was clear for all paths. These results indicate that much can be gained from a multi-channel or frequency agile approach.



(a) Machine room

(b) Compressor house

Figure 2.7: Packet loss versus path and channel [9]

2.6.2 Basic Solutions for Challenging Environments

There are several strategies that can help to improve network performance in a harsh industrial environment. To get the most reliable network possible, combinations of these techniques should be considered. Four basic approaches will now be presented.

Spatial Diversity

Spatial Diversity involves multiple paths from every node in the network. This way a node is able to route data on alternative paths if one path is blocked, either due to a failed node, interference or multipath. As described in section 2.4.4, a mesh topology will increase both reliability and power consumption. That is way different protocols have different approaches to this strategy. Recall the star-mesh topology in section 2.4.3.

Frequency Diversity

Frequency hopping provides strong immunity to both multipath and interference. The standard implementation is to let communication follow some predefined pattern, transmitting on a different frequency if the last transmission failed. This requires network synchronization, but in most cases this is needed anyway, to provide reliable collision free communication (section 2.7). In addition to increased reliability, frequency hopping also provides increased bandwidth and throughput capabilities due to 16 available channels. Nearby nodes no longer have to wait in turn, but may now transfer data simultaneously. A criterion is of course that they are correctly scheduled, a task that is usually performed by some centralized controller

(gateway). Dense networks have much to gain by such approach. In combination with Direct Sequence Spread Spectrum (DSSS) or decoding and interleaving schemes, this makes the network robust to most challenging environments.

Temporal Diversity

Retransmissions are a fundamental technique utilized by most protocols, both wired and wireless. In wireless communication the chances for a failed transmission, whether it is due to collisions, interference or multipath, is much greater than in any wired network. In the case of interference or multipath it is necessary to combine this approach with that of spatial and/or frequency diversity. In fact, temporal diversity alone is seldom the key to a completely reliable WSN. However, it is necessary in all approaches to maintain full reliability.

Increased Transmit Power

Higher transmit power improves the link margin in general and increases the radio range of the sensor nodes. This will in turn reduce the number of routers needed in the network. However, it also increases the power consumption of the nodes and may in some cases force central nodes to be powered by wire. It should also be mentioned that even though increased transmit power can make the network more robust, it can't give any performance guarantees. Recall the example of online gaming, where the performance of WiFi [2] was strongly affected by a wireless phone. These trade-offs must be considered when deciding on what kind of protocol that is required, and whether or not battery-powered nodes are essential for the current network.

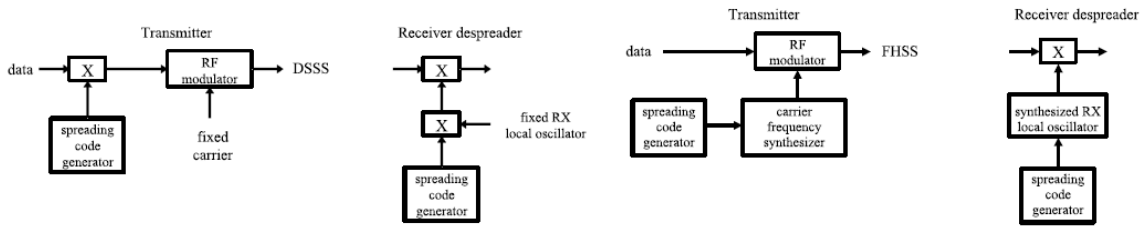
2.6.3 DSSS versus FHSS

Today, when a protocol is utilizing Frequency Hopping Spread Spectrum (FHSS), it usually doesn't involve pure FHSS in its original definition, but a combination of Direct Sequence Spread Spectrum (DSSS) and frequency hopping. Sensicast has named this technique for Distributed Frequency Spread Spectrum (DFSS), a name that will be used in the remaining part of this section. Throughout the rest of the report, FHSS is used synonymous with DFSS if nothing else is specified.

Original frequency hopping (FHSS) is actually just a collection of conventional narrowband signals. The principle is simple; a spreading signal is used to change the frequency on the carrier provided by a carrier generator. This hopping carrier is directly modulated by the data that is to be sent. In contrast, DSSS combines the information signal with a spreading signal having much wider bandwidth. This wide modulation is then applied to a fixed frequency carrier signal for transmission (RF modulation). The principles of DSSS and FHSS are shown in figure 2.8. For more details about the technologies, see [11].

In the presence of narrowband interference, the performance of DSSS and FHSS techniques differ significantly under certain operating conditions [11]. Without going into the details, the spreading and despreading process of DSSS makes it immune to most moderate level interferers. The spreading process can be thought of as a way to "break" the data into little pieces, while the despreading process, using the same spreading code, "knows" where these

[2] WiFi devices have a typical output power of about 100 mW or 20 dBm.



(a) DSSS (b) FHSS

Figure 2.8: Principles of DSSS and FHSS [11]

pieces are and collects them back together. In this reassembly process, any other signals will not match and is therefore broken up into pieces of its own. In contrast, FHSS may become completely blocked on a specific channel if it is exposed to a narrowband interferer. However, for a large out-of-band interferer, the opposite is true. The DSSS process is sensitive to such interferers while the FHSS system is not. Hence, a combination of the two, like in DFSS, may be the best way to sidestep most interferers.

2.7 Medium Access Approaches

Medium Access Control (MAC) protocols coordinate the access of a number of nodes to a shared communication medium. This should be done in a way such that certain application-dependent performance requirements are satisfied, including delay, throughput, fairness, and low power consumption. The last criterion is of special interest for WSNs, where battery-powered nodes are essential in many applications. There are hard trade-offs involved in the choice of the most suitable MAC protocol and many approaches currently exist. This section[3] briefly discusses some of the fundamental techniques and strive to give an overview of the challenges they present.

2.7.1 Protocol Background and Basics

The OSI (Open System Interconnect) reference model [5] divides the functions of a communication protocol into a series of layers (seven). Each layer has the property that it only uses the functions of the layer below, and only exports functionality to the layer above. The idea is to make development of protocols more structured, easier to maintain and allow simultaneous work on the different parts of a protocol. Most WSN protocols usually implement the three lowest layers of the OSI model and add an application layer on top of that. The three lowest layers consist of a physical layer (PHY), a data link layer (DLL) and a network layer (NWK), presented in a bottom-to-top order.

To give a brief description of the three; the PHY layer comprises the interface to the transmission media and is mostly concerned with modulation and demodulation of data. The DLL layer places data into frames and establish communication. This includes encoding and decoding of data into bits. Within this layer we also find the MAC sublayer, which is the topic of this section. Since it is the first protocol layer above PHY, it is heavily influenced

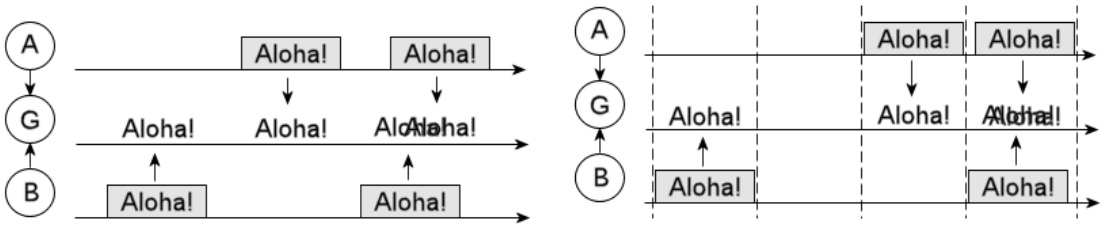
[3]All images in this section, except figure 2.12, are modified from a web seminar[12] hosted by WINA.

by its properties. The second part of the DLL consists of a sublayer, Logical Link Control (LLC), with responsibilities like error and link control. That leaves only the NWK layer, which provides switching, routing and path creation for node-to-node transmissions.

2.7.2 ALOHA

The ALOHA system [4, 5] is generally described as the first wireless computer communication system employing random access. It was originally designed to allow people in different locations to access a main computer system, and unlike existing solutions at the time, it was going to do so using a shared medium for transmission (packet radio). This presented issues that are still a hot topic of research even today, over 30 years after the invention of ALOHA.

The pure ALOHA network uses a star topology consisting of a central node and a number of remote devices. Two physical channels are used for communication, one for inbound transmissions and one for outgoing transmissions. The remote devices can at any time, and in a completely asynchronous manner, access the central node on the inbound channel. When collisions occur, data has to be retransmitted after a random back-off time. Assuming that the offered traffic follows a Poisson distribution, the maximum throughput of ALOHA is about 18.4 % (theoretically). Any attempt to exceed this limit would only increase collisions and the overall data throughput would actually decrease, a phenomenon known as congestion collapse. The operation of a pure ALOHA network is illustrated in figure 2.9(a).



(a) Pure ALOHA

(b) Slotted ALOHA

Figure 2.9: Operation of pure and slotted ALOHA

When the first version of ALOHA was released, it immediately sparked interest in others who saw the opportunities a shared transmission medium could bring. From this day on, researchers all over the world have been working on protocols that improve the performance and reliability of such systems. Many techniques have been suggested to reduce the number of collisions in ALOHA. One solution is to use a different frequency for every node, a system known as frequency multiplexing. However, as the number of nodes increases, so will the frequencies. The result is an expensive system with poor flexibility and poor scalability.

Another solution is to have ”time slots” into which nodes are allowed to transmit data, known as slotted ALOHA. The nodes were, however, not assigned specific time slots like in TDMA, which will be discussed in section 2.7.4. As a result, collisions can still occur if two or more nodes try to send their data in the same time slot. However, the nodes are now restricted to start a transmission at the beginning of a time slot, and thus collisions are reduced. Assuming Poisson distribution, this technique increases the maximum throughput to 36.8 % (theoretically). The operation of slotted ALOHA is shown in figure 2.9(b).

Slotted ALOHA also introduced the need for network synchronization. In this particular protocol, the star topology made this an easy task. A centralized clock sent out small clock tick packets to the remote devices, which immediately upon reception was allowed to send their data in return. This approach was soon followed by more advanced techniques, which will be presented shortly.

2.7.3 CSMA

The study and research to improve the ALOHA system resulted in a range of new protocols and a constantly increasing body of theory. The Carrier Sense Multiple Access (CSMA) protocols represent a large family of such protocols. All of which share a common access method that require the nodes to check for channel occupancy before transmission.

In pure (non-persistent) CSMA, a node first checks the channel to see if it is idle, and if it is, transmits its data. In the case of the opposite, when the channel is busy, the node backs off for a random period until it again tries to transmit its message. The same procedure is then repeated. The principle is depicted in figure 2.10.

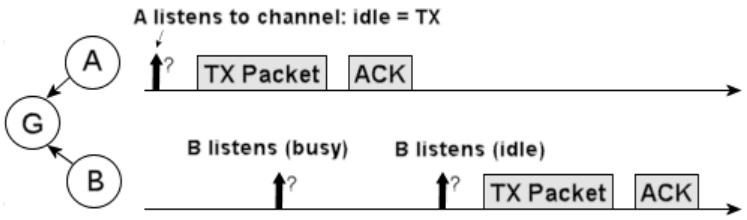


Figure 2.10: Operation of CSMA

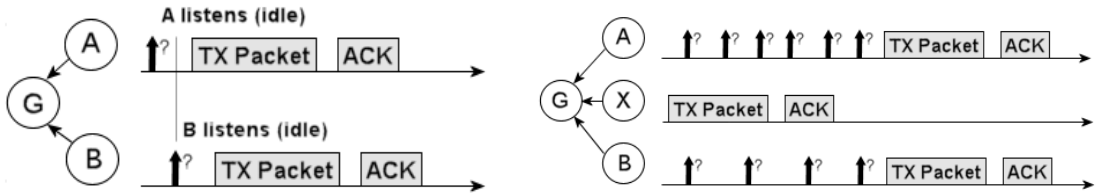
Challenges and CSMA Improvements

The random back-off period of CSMA introduces some important issues. When non-persistent CSMA is used, a node will keep on waiting throughout the entire back-off period, even if the channel becomes unoccupied during that time. However, if nodes could detect such idle time and transmit immediately (persistent), a much more efficient network would be achieved. This of course requires some kind of mechanism to avoid collisions, which may occur if more than one node is waiting to transmit. To deal with this scenario, p-persistent CSMA was proposed. In this particular approach, nodes transmit upon idle channel detection only with a probability, p. With a probability (1-p), they delay their transmission. The optimum value of p is a function of delayed traffic rate[4].

Another issue of CSMA presents itself when two nodes both have data to send, and at approximately the same time. The transmitter of a node cannot change directly from receive to transmit, and as a consequence there is a gap in time where other nodes may listen and find the channel idle. The problem is illustrated in figure 2.11(a), where node B listens and finds the channel idle while node A is switching from receive to transmit mode. In a pure CSMA network, such collisions are not discovered until the end of the transmission, resulting in a lot of wasted bandwidth. Improved versions of CSMA, like CSMA/CD, include collision

detection (CD). With such protocol the nodes are able to detect when a collision occurs and immediately stop transmitting. The standard procedure with a random back-off period is then initiated.

There also exist solutions that tries to avoid the problem altogether. One protocol of this kind is CSMA/CA, which provides collision avoidance (CA). Before a transmission takes place, a node has to inform all other nodes that it intends to transmit. When the other nodes have been notified, the information is transmitted. Many variations of collision avoidance exist, and solutions involving request to send (RTS) and clear to send (CTS) have been implemented in many forms. However, there is still a possibility that collisions may occur, and if they do, CSMA/CA does not include collision detection. The result is again wasted bandwidth.



(a) Switching problem

(b) Synchronized listen problem

Figure 2.11: Issues in CSMA

The scenario depicted in figure 2.11(b) once again refers to that of back-off periods. In the picture, both node A and node B periodically listens to the channel to check if it is idle, while node X is transmitting. The problem arises if the two nodes (A and B) are allowed to synchronize their listening, making collision unavoidable when node X has finished its transfer. This once again shows how important a good back-off scheme is to the overall performance.

The Hidden and Exposed Terminal Problems

Two other well-known problems that CSMA suffers from are ”hidden terminal” and ”exposed terminal”. The hidden terminal or ”hidden node” problem is easily explained by a quick look at the nodes in figure 2.12. The circles indicate node range. Thus, node A and C are in range of B, but not in range of each other. As a result, if either node A or C is transmitting to node B, the node that is not transmitting may sense the channel as idle and start a transmission that interferes with the ongoing transmission to B. This indicates that in the presence of hidden terminal, CSMA is actually reduced to that of pure ALOHA.

The other problem, exposed terminal, occurs during the same range conditions. In a scenario where node B is transmitting to node A, node C may very well transmit a message to node D, without interfering the ongoing transmission between A and B. The problem is that node C will sense the channel as busy, and instead of transmitting its message to node D, it will back-off for a random period and wait for an idle channel.

Both hidden and exposed terminal results in reduced bandwidth or channel capacity. The first solution proposed was the use of ”busy tones” [4]. This involve the use of two different channels, where one channel is used by a receiving mote to indicate that it is receiving, while the other channel is used for receiving the actual message. Nodes that intend to transmit first sense the

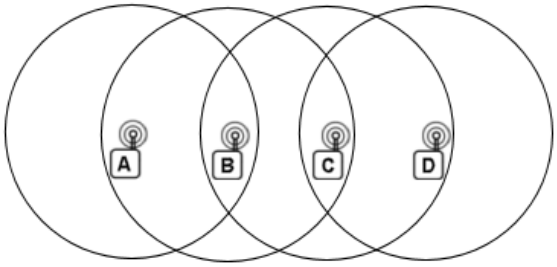


Figure 2.12: The hidden and exposed terminal problems

second channel to check for busy tones, and if none are detected, it starts transmitting. The solution, however, comes with the price of added node cost and increased power consumption, and hence it is not the most suitable approach in a WSN. Another method, which has already been mentioned, is to use a CSMA protocol with collision avoidance. Such protocols reduce, but do not eliminate, the possibility of packet collisions.

Synchronization and Slots in CSMA

In the presence of unlimited power (wired), nodes implemented with CSMA may have their transmitters in receive mode all the time, only switching to transmit mode when they have something to send. However, this is rare for most WSNs, usually comprising battery-powered nodes with low power consumption as the number one requirement. Instead, nodes have to implement duty cycling, which in turn require some form of synchronization. The IEEE 802.15.4 standard [10] solves this with beacons in combination with CSMA/CA. It also includes slotted communication with guaranteed time slots (GTS), although limited in number. The details of this protocol are not discussed in this report; instead, a real network experiments with a TDMA based protocol (SmartMesh-XR) will be performed, and results are compared to that of a previous experiment with a network based on 802.15.4 (ZigBee).

2.7.4 TDMA

Time Division Multiple Access (TDMA) protocols divide time into time slots. Each slot defines a period of time where two specific nodes are allowed to communicate. Broadcasting of messages from one mote to its neighbours is also possible, but only one node may transmit per time slot. In all other time slots where the nodes are not assigned communication, they may turn off their transmitter and enter sleep mode to conserve power. A number of time slots, typically set by the user, are defined as a frame and is repeated as time moves on. The length or number of slots in this frame sets the network ”pace”, and may be adjusted to the needs of the specific network. A long frame usually involves reduced power consumption, but also longer delays and slower reporting. These trade-offs will be more thoroughly discussed in chapter 3, also providing details about a TDMA implementation - SmartMesh-XR.

A simple example of TDMA is shown in figure 2.13. At first glance it may look identical to that of slotted ALOHA, but recall that ALOHA did not assign unique timeslots for certain nodes. In contrast to ALOHA, that makes TDMA immune to collisions. Problems like hidden terminal and exposed terminal, mentioned in the CSMA section, is not an issue. With

the exception of new nodes joining, all communication is determined and scheduled during network formation, and not during normal operation. However, to obtain a both flexible and scalable network, it is important to provide "open listen" slots to sense and connect new nodes as fast as possible. The number of such slots is determined by the network requirements and structure. In a network exclusively powered by wire, low power consumption is not a criterion, and all slots in the frame can be assigned for communication, improving both performance and formation/joining times.

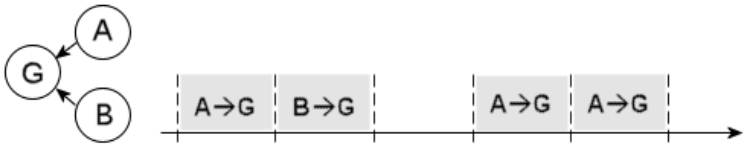


Figure 2.13: Operation of TDMA

Challenges in TDMA

Some of the challenges that TDMA faces have already been mentioned. The first challenge that comes to mind is perhaps the need for a global time reference. To maintain a common global time for all network nodes, time synchronization must be included. In an ad-hoc, multi-hop network this may be a nontrivial task. Periodic time updating by parent nodes and beacons are possible solutions to the problem. To prevent unnecessary traffic, time updates typically piggyback on data packets. Many smart solutions currently exist.

Other challenges are that of cell scheduling and dynamic bandwidth allocation. To ensure that data is delivered as fast and with as little delay as possible, it is important that links (assigned communication) are placed in the correct order in the frame. In a large network it is also important to remember that nodes close to the gateway must forward data from outer nodes, and will therefore require more bandwidth/links than nodes several hops away. In the occasion of a joining node, bandwidth must be dynamically allocated to the inner nodes to handle the extra traffic. This often involves reallocation of links in order to maintain a fast and reliable network.

Together, scheduling and bandwidth allocation form a considerable amount of work. Network management tasks like these are often handled by a centralized controller or gateway, which in turn needs a powerful processing unit (CPU) to handle the demanding work. Such device may be expensive, and downscaled versions should exist so that the cost won't exceed its utility value, for example in small networks.

Multiple Channels and Network Structures

Study and analysis of various environments, like the results discussed in section 2.6, have made it clear that both in industrial and harsh environments in general, link quality can be substantially improved by frequency diversity. In addition, it also makes it possible for a network to utilize several channels simultaneously, and by this achieve a huge increase in available bandwidth. The SmartMesh Network, evaluated in this report, is a good example on

a TDMA based protocol that utilizes this extra bandwidth. Instead of using just one channel, all 16 channels of the 2.4GHz band is used simultaneously.

The communication matrix in figure 2.14 illustrates how communication may be scheduled in a TDMA network utilizing multiple channels. In contrast to a one channel network, where it is sufficient to schedule one node to transmit while all other nodes may be set to listen, it is now important to schedule both transmit and receive. All nodes must in addition to a time slot, also be assigned to a specific channel. Apart from this, TDMA still functions as described in the previous sections. The increased bandwidth may now be used to increase network performance, but it is important to be aware that more assigned slots leads to increased power consumption. Because of this trade-off some protocols/networks choose to implement a backbone structure with nodes powered by wire. It is even possible to combine various medium access approaches, for example, by using TDMA in a wired backbone network and CSMA/CA for the leaf nodes. Real network examples are presented in the following chapters.

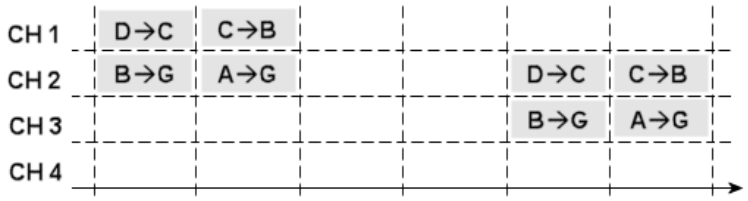


Figure 2.14: Communication matrix

Chapter 3

The SmartMesh Network

This chapter contains fundamental information about SmartMesh[1] technology and will give an overview of the SmartMesh Network. The basics are then followed by a section concerning the more advanced theory of network operation. To sum it up, the final goal of this chapter is to provide information to fully understand the properties and behaviour of the SmartMesh Network.

3.1 Introduction and Overview

This section briefly explains the most important features of the SmartMesh protocol and provides a background for the more advanced details in the following sections. Readers who are already familiar with Dust Networks and the SmartMesh technology may move on to section 3.2.

3.1.1 Dust Networks and their SmartMesh technology

Dust Networks [13] is a privately held venture-funded company that specializes in wireless sensor networking systems for industrial and commercial markets. The company was founded in 2002 by a team including Kris Pister, a professor at the University of California, Berkeley, and the originator of the Smart Dust concept. California is also the state of which the company is currently located, more specific in the city of Hayward. Their latest achievement is SmartMesh-XR, a product which has received much attention because of the many convincing properties it comprises.

SmartMesh Networks are ultra low-power wireless mesh networks with properties that make them highly reliable and easy to install. These characteristics make SmartMesh Networks suitable for a wide range of monitoring applications, such as building automation, industrial monitoring, and remote site security. A discussion of the technology that gives SmartMesh-XR its unique properties is carried out in the following section.

3.1.2 SmartMesh Components and Network Structure

In a SmartMesh Network two types of devices can be found; at least one SmartMesh Manager (Manager) and up to 250 motes. The motes are ultra low-power wireless transceivers with

[1] Dust Networks and SmartMesh are trademarks registered by Dust Networks.

connections for analog, digital, and serial sensors and actuators. When a mote is integrated with one or more sensors it should be called a SmartMesh device, although this term is often skipped. More information about SmartMesh devices is provided in section 3.3.1.

In the case where the mote is embedded in a SmartMesh device it digitizes data from the integrated sensors and sends the packets to neighbouring motes. The neighbours then pass it along to other motes, and in a series of "hops" the data is eventually delivered to the Manager. The Manager is the line powered network node that controls and monitors network performance. It is responsible for network synchronization, routing and collection of mote data. In addition it collects network statistics and streams data to client applications over a wired IP network. Client data is sent in Extensible Markup Language (XML) format.

Clients are also allowed to configure, monitor and manage the network through the Manager Application Programming Interface (API). This is done by Remote Procedure Call (RPC) requests from the client applications[2], which in turn receive responses and other data from the Manager via XML-RPC. The topic is covered more extensively in section 3.3.3.

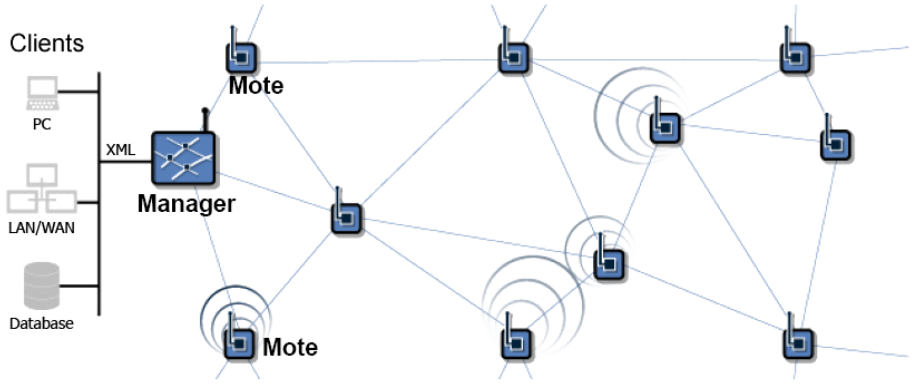


Figure 3.1: The SmartMesh Network [15]

3.1.3 Network Topology

For the network to form in a reliable mesh topology, all motes need at least two parents. The placement of the motes is therefore crucial, more information about the subject can be found in section 3.3.2. If these steps are followed, each mote is set up with more than one possible path. In cases of mote failure or blocked signals from a mote, the children of that mote will simply use other available paths. This is often referred to as resilient or self-healing routing. Further, to make installation and maintainance an easy task, the SmartMesh protocol implements self-organizing algorithms (ad-hoc networking).

3.1.4 Frequency Hopping

In addition to the reliable topology, Dust has also integrated frequency hopping to sidestep radio frequency (RF) interference. Motes can communicate on different frequency channels

[2]Dust Network provides a client application for this purpose, named SmartMesh Console.

within the 902 - 928 MHz range or 2.4 - 2.483 GHz range, depending on the chosen network devices. Each time data is sent on a specific path it results in a change of frequency. More specific, a frequency change occurs at the end of a time slot occupied by the specific path. This is called a link, a concept which will be more extensively explained throughout the present report. The frequency changes ensures that each mote has an alternative path if one path is blocked due to RF interference. It also leads to ease of installation, due to the fact that no frequency selection needs to be done.

3.1.5 Time-Synchronized Communication Protocol

Determinism for all motes in the SmartMesh Network is achieved through Time Division Multiple Access (TDMA) technology (discussed in section 2.7.4). In short, this is a way to divide network communication into different communication time slots. Together these slots form a (time) frame, which is repeated throughout the network lifetime. This makes communication timed and synchronized for the different motes along a route. As a result, mote radios only need to be powered on for the time it takes to transmit a packet to a neighbour or listen for a packet from a neighbour. This in combination with low data rates keeps the power consumption low and makes it possible for long-life battery powered motes. Depending on the network configuration, such as duty cycle and the amount of network traffic, motes can operate for years on standard AA batteries.

3.2 The Network Protocol

While the previous section introduced the key concepts of the SmartMesh network, this section will look into the details of the protocol. This section can be used as a reference throughout the report.

3.2.1 Physical Foundation

All SmartMesh motes in the 2.4GHz ISM band are equipped with radio transceivers of the type, CC2420. This is an IEEE 802.15.4 compliant radio produced by Chipcon AS [3], a former Norwegian company specialized in short-range RF communication. The same transceiver has also been used on Chipcon's ZigBee evaluation kit, CC2420DB. Later in this report the two technologies will be compared.

The IEEE 802.15.4 standard [10] defines the Physical layer (PHY) and Medium Access Control sublayer (MAC) for Low-Rate Wireless Personal Area Networks (LR-WPAN). It was developed to make a common foundation for battery powered wireless applications with relaxed throughput requirements and a limited Personal Operating Space (POS) of 10 meters.

The modulation technique of the 802.15.4 radios is based on Direct Sequence Spread Spectrum (DSSS), but Dust has implemented algorithms in the above layer to achieve Frequency Hopping Spread Spectrum (FHSS) [4]. Dust also provides their technology for the 915MHz ISM band, but since the evaluation kit used in this project makes use of the 2.4GHz, the main focus will be on this frequency band.

[3] Chipcon AS [21] was acquired by Texas Instruments [20] on January 24, 2006.
[4] Recall the discussion, DSSS versus FHSS, carried out in section 2.6.3

As the above section indicates, Dust only uses the PHY layer of the 802.15.4 standard. In the MAC layer they share some basic ideas like synchronized communication with frames and timeslots, but Dust has taken it a step further and introduced a contention-free implementation with guaranteed timeslots for all nodes.

Frequency band	2450 MHZ	915 MHz	868 MHz
Data rate	250 kb/s	40 kb/s	20 kb/s
Number of channels	16	10	1
Channel separation	5 MHz	2 MHz	N/A
Lisence free	Global	USA	Europe
Modulation	Q-QPSK	BPSK	

Table 3.1: Properties of the different frequency bands [10, 22]

3.2.2 Medium Access Control and Network Communication

TDMA was briefly explained in section 3.1.5. In addition to the mesh topology and frequency hopping feature, this is the backbone of the SmartMesh protocol. The implemented version of this technology will now be examined in detail.

Implemented TDMA Technology

A path may form between two motes when the motes have discovered that they are within radio communication range of each other. The final decision in such scenario is up to the Manager, which always tries to maintain the best possible routes and paths in the network. When a path is created it is assigned to a timeslot in the frame, the path is now termed as a link. The frames are repeated continuously, and each time that link is reached communication can happen between the two motes. All timeslots have the same specific length, and within this time one transmission may be performed. FHSS technology makes it possible to assign up to 16 links per timeslot, all using different frequencies in the 2.4GHz ISM band. To sidestep RF interference these frequencies are changed each time a link is used.

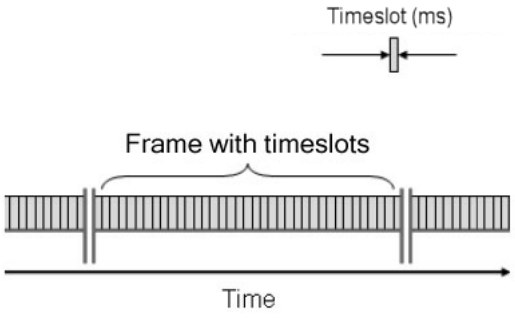


Figure 3.2: Frames and timeslots

The principle with frames and timeslots is illustrated in figure 3.2. It is important to point out

that the motes only send/receive data when their assigned timeslot comes up; else they sleep and conserve their power. The power consumption may be even further reduced if the network is handling tasks with low report rates and relaxed latency requirements. The frame length may then be extended with more timeslots, giving the motes more time between each wake-up. However, configurable settings like frame length and report intervals should be handled with care. A long frame length does not only lower power consumption, it also increases the minimum report intervals and can lead to unacceptable latency for some applications in the network.

Routing and Path Creation

The Manager is responsible for the creation of a path and its assignment to a time slot. This must be done in such way that best possible routes are achieved. Every time a mote reaches its assigned timeslot it will check its queue to see if it has data to send. If data is available it will be sent instantly. A packet is not removed from the queue until an ACK is received.

Path choices are made based on hop depth as long as the Received Signal Strength Indication (RSSI) is above a certain threshold. This threshold is currently set to -87 dBm, but can be changed by adjusting a register value. Hence, during the neighbour discovery process all neighbours that make the -87 dBm cut get prioritized based on their hop depth. The algorithm was chosen because RSSI values are constantly changing and are thus not trustworthy as a metric for best path.

Time Slot Assignment

Assignment of time slots is done during network formation. This slot assignment may, however, be adjusted if new motes join the network. In addition to the obvious upstream links there are also links for downstream communication, listen links for joining and a link for neighbour discovery. The downstream link is used to listen for possible information (broadcasts) from the Manager, all motes need at least one such link per frame. Every mote also has one listen slot per frame, to listen for potential children trying to join. The last link type is neighbour discovery. These links are found in the first slot in the frame, slot zero, and is used by all motes to find out who could possible talk to whom. Every time the slot is reached a new mote is selected to transmit, and all other motes are listening. The Manager then uses this information to maintain a table which can be used in cases of mote failures and the like.

For an integrator to make correct configuration choices it is important to fully understand how the Manager assign links. Upstream links are assigned to a mote based on its number of children. It is important that a mote is given enough bandwidth to forward the packets it receives from other motes. Frequency hopping makes it possible with 16 links per slot, but the Manager can still receive data from only one mote at a time. The absolute minimum frame length is thereby limited to the number of motes (N) in the network, plus additional four slots for broadcasting, neighbour discovery, open listen, and one slot for advertising. The advertising slot is necessary for new motes to join the network when the Manager is the only participant. More information is provided in the next section, ”Joining of a New Mote”.

The minimum frame length (N+4) is, however, only applicable in a pure star topology. If motes are connected in multi-hop or meshed together with multiple paths, the network will need more slots to carry data. In the case with a linear/multi-hop network, like the one

depicted in figure 3.3, the mote closest to the Manager is the one in need of most slots. As figure 3.3 illustrates, this will leave that particular network with a minimum frame length of nine slots. The ninth slot must be added to cover neighbour discovery. These links are not shown in the picture because they all share the same time slot.

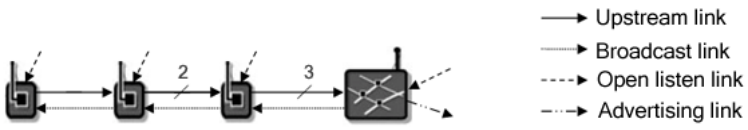


Figure 3.3: Network with a minimum frame length of nine slots

As stated earlier, the absolute minimum frame length (N+4) is too short for most networks. For a network to form in a reliable mesh topology it is necessary for each mote to have at least two parents. This results in more upstream links for the connected parent motes, and sometimes the need for an extended frame length to cover the extra links. A simple example of such network can be viewed in figure 3.4. An additional mote has been added to the network in figure 3.3, to illustrate how the minimum frame length is extended with several slots. Also notice that the broadcast links are located in the same time slot. Because motes can have more than one parent, some motes are likely to receive the broadcasts more than once. As with the previous figure, one extra slot is needed for neighbour discovery, not illustrated in the picture.

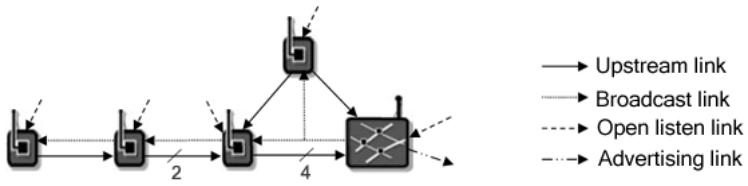


Figure 3.4: Network with a minimum frame length of eleven slots

Joining of a New Mote

When a mote wants to join a network, it keeps its radio in receive mode listening for packets with advertising trailers. The advertising trailer is piggybacked on regular or keep-alive packets and gives no extra overhead in the network. It contains all necessary information for a new mote to synchronize to the network, including frame size, parent open listen slot (and frequency channel), and parent downstream broadcast slot (and channel). Information about current-time and network ID is found in the MAC header. Joining motes must have an identical network ID as the network it tries to join; more than one network may coexist when SmartMesh technology is used.

During the discovery period a joining mote spends a certain max time (settable in registers) listening on the different channels. When max time is reached it moves on to the next channel and so on. Upon receiving an advertising packet, the mote synchronizes its listen times for

each time slot (making it consume a lot less power since the radio is not on all the time). The mote stays in this synchronized listen state for a maximum duration (also settable in a register), gathering the channel and offset values of all its neighbour's broadcast links and open listen links. When the state timer expires, the mote moves into the join state.

In the join state the motes creates local receive links corresponding to every neighbour's downstream broadcast link. This is done in order to listen for a potential "activate" command. The mote then randomly selects a neighbour and sends a join request to this neighbour's open listen link. Join requests are forwarded to the Manager, which has a much better view of the entire network. The Manager decides the two most suitable parents for the mote and sends it an activation packet trough one of its neighbours. At this point, the activating neighbour does not need to be aware of the joining mote. For example the Manager could communicate to "mote 35", to broadcast an activation packet to "mote 47". Then, if the activation is received successfully by "mote 47", the parent will be aware of the child.

This is what happens from the Manager point of view: An activate child packet is sent to the coming parent of the joining mote. The Manager then awaits an ACK from the parent, indicating that the parent received the command. This ACK should be followed by an activation confirmation from the child. The next step is then to send add link commands to the child, parents and other upstream ancestors as needed. ACKs must be received for all added links. A sensor configuration packet may then be sent to the child. As with all other commands, an ACK should be received shortly.

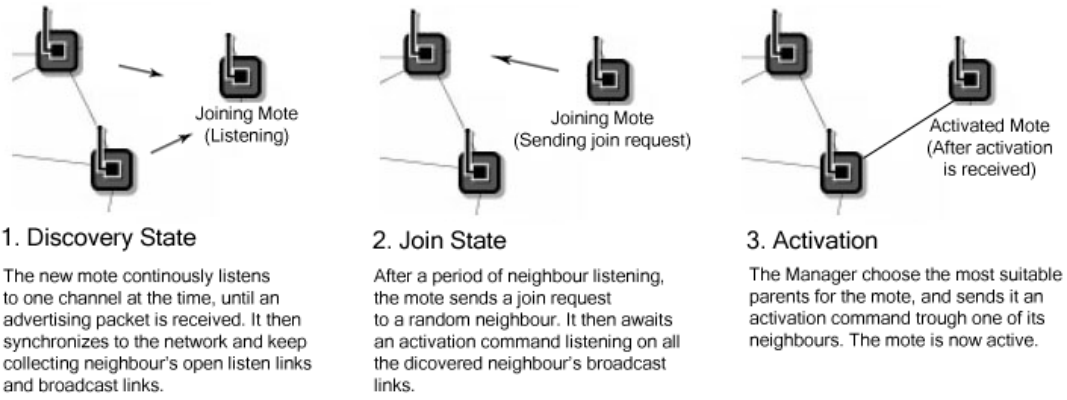


Figure 3.5: Summary of the joining process

Multiple Frame Feature

The SmartMesh protocol is capable of running multiple frames simultaneously. In the current version of the software this feature is not fully turned on for the end user, but is restricted to a fast frame (31 slots) that is initiated if the main frame exceeds 50 slots. The fast frame, frame 1, is used to enable fast joining and hence faster formation of the network. The main frame, frame 0, is still the basic frame that handles all necessary network and user tasks. When the two frames are running simultaneously, frame 0 has priority over frame 1. This means that if two different motes are scheduled to talk to the same receiver in the same slot,

but within two different frames. The receiver will choose to listen to the frame 0 transmitter, regardless of whether the transmitter sends anything or not. Hence, if the frame 1 transmitter sent something at that time, then that packet will get lost (not get an ACK).

Frame 1 is not always on, but is turned on automatically in the presence of a qualifying event. Such event could be a new mote joining a network, an existing mote resetting and trying to rejoin the network, etc. Once active, the frame remains on for 1 hour. As a result, the mote power consumption is slightly higher at start-up and during the listed events.

In a future release of the software it is planned to enable multiple frames for users, making them able to configure and customize frames for specific needs. The idea is to run sub-networks (portions of a network) on multiple frames to achieve specific functionalities. For example; in a 50 mote network running frame 0 at 200 slots, one could have 2 motes like a light switch and a light running on a very fast 2 slot frame, or a HVAC and a Temp sensor running on a 30 slot frame, etc. According to Dust, the infrastructure for such scenarios is already in place, and they are currently figuring out ways for a user friendly interface.

Transmissions Details

As previously mentioned only one transmission may occur in a time slot per mote. When a packet is sent, the sending mote switch to receive mode and wait for ACK. If ACK is not received when the slot ends, the mote will retransmit the packet in the next available slot. This also happens in cases where a negative acknowledge (NACK) is received. NACKs are generated when a receiving mote gets packets that it cannot process because its queue is full due to congestion, or if the packet is corrupted as indicated by bad CRC, etc. In this case it sends back a NACK, which are also counted as failed transmissions.

A mote is currently programmed to retry a certain number of times or for a specific period of time (for both parents), whichever has the highest value. These numbers are programmed in mote registers and can potentially be changed. The mechanism, however, will essentially remain the same. A counter is used for every path to keep track of the retransmissions (or the time), and are reset every time it gets an ACK. Once the counter expires the mote sends a ”path alarm” to the Manager via its second parent. The Manager acts upon this by finding another suitable parent for it. If both parents are down the mote resets itself and starts a join sequence just like a new mote would when it joins the network.

The largest possible packet size is 128 bytes with 80 bytes of user payload, hence the overhead is 48 bytes. Maximum user payload of 80 bytes can only be achieved by using the mote serial interface, for example if a smart sensor where connected to the mote. Sample generation from the analog inputs is limited to 18 bytes.

Manager Tasks Concerning Reliability

In the current version of the SmartMesh software the Manager only deals with reliability issues when a path has been struggling for some time. As explained in the ”transmission details”, ”path alarms” are only sent when the specified retries/timeout threshold is breached. It is of course possible to deal with this at application level, setting alarms to go off if the reliability or path stability goes down below a desired level. In the SmartMesh console this must now be acted upon manually, but it should be an easy task to make software to automatically reset

motes with issues like these. Bear in mind, though, that health reports are only received once every 15 minutes.

Discussions with Dust Networks have revealed that a new version of the software will contain a new feature called dynamic optimization. The Manager will then be able to dynamically allocate bandwidth - add bandwidth/links where needed and take away where it is not. Thus, motes with higher report rates will get more bandwidth and many issues associated with congestion will be alleviated.

Queue Management

The current implemented software contains a simple FIFO queue in each mote. To maintain full reliability, a mote will not ”drop” packets unless it resets, but if the queue is full (holding 8 packets) it will stop generating samples. This results in NACKs and longer periods between samples in a congested network. As described in the ”transmission details” a mote will not reset until all paths are down, which will happen when all paths (both parents) have reached the register-programmed time without receiving ACK. The topic of queue management is currently under research, and queue-FIFO management features are planned in a future release of the software.

Security

In addition to the obvious increased security obtained by FHSS, the SmartMesh protocol also include 128 bit encryption. This combination should make it hard for people with bad intensions to read or write any information to the network. To send data a hacker must know the hopping sequence, time between transmissions and the current frequency. And even if this were known, the encryption will make it close to impossible. Jamming of the network traffic will also be a challenge, due to the FHSS and the mesh topology.

3.3 The SmartMesh Network from Factory to Installation

Before a fully configured and functional SmartMesh network is achieved, the motes are involved in several primary processes. This section will look briefly into the major parts and give an overview of the different steps on the road to completion.

3.3.1 Sensor Original Equipment Manufacturers

SmartMesh motes can be integrated with a wide variety of sensors and actuators to create SmartMesh devices. This process is carried out by Original Equipment Manufacturers (OEMs), which integrate motes in their sensor designs to make wireless devices suitable for specific tasks. A device can be defined as a self-contained product with its own power supply and sensor(s). By this definition the motes provided in the Smart Dust evaluation kit are indeed SmartMesh devices. Not only do they contain their own temperature sensor and power supply, but are also extended to contain inputs for external sensors. Dust Networks’ main target is, however, to provide motes for other larger sensor OEMs. Each such mote contains a unique MAC address, SmartMesh software and a default network ID (and password).

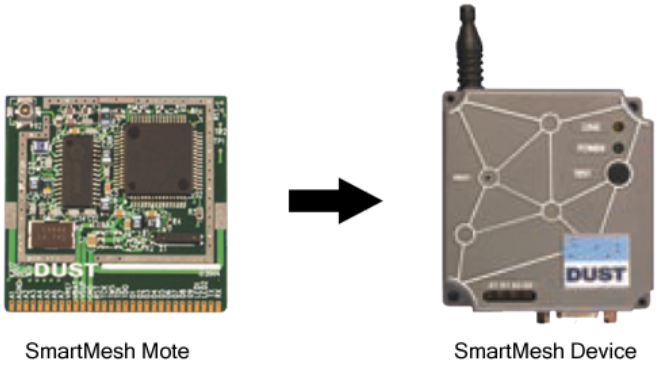


Figure 3.6: Mote integrated into a SmartMesh Device

According to the sensor OEM guide [14] the motes currently provided (M2020 and M1020) have seven analog inputs, eight digital channels and a serial interface. It is obvious that not all of these will be used in all device designs, and thus it is necessary for the sensor OEMs to provide configuration files for their devices. These configuration files are called datasheets, and define the mote channels used on the SmartMesh device. For each SmartMesh device design, it must be created a separate datasheet with a unique datasheet identifier (ID). The datasheet ID is an eight-digit identifier that allows the Manager to identify the configuration of a SmartMesh device when that device joins a network. For this to work, the datasheet ID must be programmed onto the mote in each device, and the datasheet must be downloaded to the manager. The former is done by the sensor OEM, while configuration of the Manager is performed by the system integrator. Before SmartMesh devices can be marketed to system integrators, the SmartMesh device design must meet the regulatory requirements for the region in which it is to be sold and receive appropriate certification. Sensor OEMs must also provide calibration coefficients like slope and intercept values that define the calibration of the device's analog sensors. For more information about mote integration and specifications, see the Sensor OEM Guide [14].

3.3.2 System Integration

System Integration involves several tasks, which in the end results in a fully functional and verified network solutions for the customer. The first step in this process is to gather specific information about the site to estimate the network size, determine the type of sensors needed, and scope the cost of the network. Then, if the customer accepts the bid, it is time to purchase network components and software, and pre-configure these for installation. Recall from the previous section that SmartMesh Managers are provided by Dust Networks, while SmartMesh devices and control software are provided by sensor OEMs and independent software vendors.

In all the mentioned steps there are important rules that should be followed. This section only emphasizes those of particular interest for this report, like the most important configuration parameters and installation considerations. For more details than presented here, see the System Integrator Guide [16].

Estimating and Planning a Network

The network size varies with the required number of sensors, the location of these sensors and the challenges presented by the environment [5]. Sensor locations are important, both because external sensors at close locations may be wired to the same physical mote, and also because long distances results in the need for more router motes. More routers may be required if the network is located in a harsh environment or is exposed to strong interfering signals. A thorough analysis of the environment to locate RF barriers and strong sources of interference is therefore crucial. The former comprise heavy machinery, concrete walls, large appliances and metal structures etc., while a typical RF interferer could be Bluetooth, WiFi, a cordless phone or other equipment utilizing the 2.4GHz band. It is also important to locate LAN and AC power access points that are near the measurement areas. The Manager(s) must be placed at such locations, preferably close to the highest concentration of motes or near multiple clusters of motes. This placement reduces the number of motes needed as signal repeaters and minimizes network latency [6].

With these considerations in mind and some knowledge about mote range in the presence of various challenges, it should be possible to find the most suitable locations to install the motes. To achieve a reliable mesh network it is recommended that all motes have at least three neighbours. Details about mote range and planning strategies are found in [16].

Pre-configuring a Network

The pre-configuration of a network is performed with a software application like the SmartMesh Console, which is able to program and communicate with the Manager. Before the network may be employed there are three configuration tasks that need to be completed, all of them involving the SmartMesh Manager. The tasks include configuration of network settings, importing of the appropriate data sheets and creating of profiles.

Network settings comprise many important configuration parameters. To allow any communication at all, it is important that both motes and Manager(s) are programmed with identical IDs and passwords. As long as these parameters are unique, it makes it possible for multiple networks to coexist in the same area. Other important settings like frame length and maximum number of expected motes decides the maximum network size and affect properties like latency, power consumption and available bandwidth.

The appropriate data sheets must be imported to the Manager for it to identify the configuration of a mote when that specific type of mote joins the network. Profiles are used to customize the motes, for example to turn off unused channels, decide report rates and report types etc. The number of channels and features to customize depends on the mote as described in the data sheet.

Mote Installation

When installing motes, there are a few guidelines that should be followed [16]. The importance of these simple rules will be illustrated by the experiments of this project, where it is shown that even centimetres can make a huge difference in signal strength. To get the best

[5] RF Challenges were discussed in section 2.4.

[6] Recall the discussion about network topologies, section 2.6.1 - more hops results in higher delays.

possible result, motes should be placed at least one to two feet above the ground with its antenna pointing upwards. Alternatively, the mote may be placed on its side to allow it to transmit both below and above its current position. The mote RF signal is transmitted in an arc, with maximum signal strength occurring in the area 45° above and below the tip of the mote antenna. This is illustrated in figure 3.7. Since the environment and the location of the motes may require that they are positioned at different heights, it is important to make sure that they are within this 45° range.

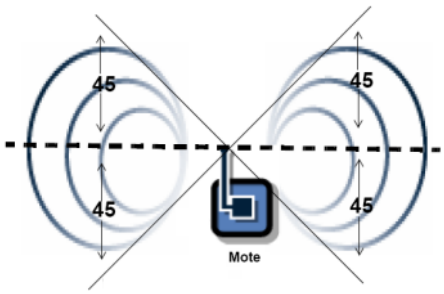


Figure 3.7: Mote Antenna Range [16]

The remaining rules are all concerned with the environment. Make sure to avoid mote positioning on metal objects as long as it is possible, and if there is no way to avoid such installation, try to position the mote with its antenna as clear of the metal as possible. Another location that should be avoided is next to equipment utilizing the same frequency band as the motes, like a WiFi access point or a cordless phone. This may degrade performance and in the worst case completely block the mote for a period of time. As a final guideline, try to avoid vibrating surfaces and locations where the temperature is outside the rated range of the motes (0°C to 60°C). This might damage the mote and degrade performance.

3.3.3 Client Applications

Client applications allow users to interact and exchange data with the network through the XML-RPC interface provided by the Smart Mesh Manager. The Manager uses two channels for communication with a client; a two-way control channel that permits requests and responses between the Manager and a client, and a one way (asynchronous) notification channel that streams data from the Manager to a client. This data typically consists of events like alarms, joining motes, disconnected motes and so on, and depending on the network type, the client may be programmed to act upon such events and take action through the control channel. Dust networks provides a XML-RPC API [17] with the necessary commands to perform most network tasks.

XML

The Extensible Markup Language (XML) is best described as a cross-platform, software and hardware independent tool for transmitting information. In short, it provides a text-based, self-describing and human-readable means to apply a tree-based structure to information.

XML is a markup language much like HTML, but while XML is about describing information,

HTML is more focused on the displaying part. Moreover, the XML language do not utilize predefined tags like HTML, but let the programmer define his/her own tags that makes it easy to both handle and recognize data. For this purpose, XML uses a Document Type Definition (DTD) or a XML Schema. A simple XML example is illustrated below.

```
<imaginaryMethodCall>
  <imaginaryMethodName> sendData <imaginaryMethodName>
  <parameters>
    <parameter><int> 123 </int></parameter>
    <parameter><int> 456 </int></parameter>
  </parameters>
</imaginaryMethodCall>
```

XML-RPC

As described in [17], XML-RPC is a simple framework that enables a computer to execute remote procedures on another computer using HTTP and XML. In a typical XML-RPC exchange a client computer uses HTTP to send an XML document containing a method name and arguments to a server. The server invokes the method with the arguments, and then wraps up the return value of the method in another XML document, which it sends back to the client.

More information about XML and XML-RPC can be found in [23, 24]

Chapter 4

SmartMesh-XR Competitors

Dust Network is not the only company providing reliable mesh networks. Among the networks currently considered as a basis for wireless HART we find products from both Sensicast and Coronis. Until recently, before this new technology emerged, ZigBee was the number one choice for the standard. But because of its shortcomings of deterministic transfers it was found inappropriate for industrial applications with hard real time demands. This chapter will focus on the (new) SmartMesh competitors and how they compare to the SmartMesh protocol.

4.1 Wavenis from Coronis

Coronis Systems provides wireless solutions for a wide variety of applications. According to their webpage[25] they are one of the worlds leading providers of wireless sensor network (WSN) products and services with over 500 000 deployed products. They also claim to have installed a Wavenis network with over 50 000 nodes.

In common for both Dust and Coronis is the desire to make a reliable WSN with lowest possible power consumption. They also share some basic ideas and use the same radio transceiver in the 2,4GHz ISM band, CC2420. A short description of the Wavenis protocol will now follow. For more information contact Coronis for their technology description document [26].

4.1.1 Wavenis Components and Network Structure

In contrast to the SmartMesh Network, Wavenis only feature one type of device. In a network with many such devices, one is set up as a root device (gateway). This device is assigned ”Level 0” in the network. The other levels in the network are determined by the number of hops away from this device. If a device is directly connected to the gateway it is a ”Level 1” device, and a node connected to a ”Level 1” device is a ”Level 2” device and so on. Up to four levels (hops) are supported, letting devices function as repeaters to extend network radio range.

Thanks to a highly optimized radio link budget[1], nodes achieve long-range capability. This is done to compensate for bad propagation conditions and signal attenuation indoors, and

[1] A link budget is the accounting of all of the gains and losses from the transmitter (through the medium) to the receiver.

to compensate for poor antenna gain due to tiny footprint design. The high link budget is possible through high receiver sensitivity and moderate output power. In comparison to the SmartMesh motes with -96 dBm receiver sensitivity, Wavenis devices achieve -110dBm with the lowest possible data rate (4.8 kbps) and -113 dBm when using 19.2 kbps rate [26]. No numbers are given for the maximum programmable data rate at 78.6 kbps.

In "SmartMesh Components and Network Structure" it was mentioned that the SmartMesh Manager could handle up to 250 motes. The Wavenis network has no such limitations. However, time critical applications by their very nature limit network size (due to TDMA management when feedback is required after a broadcast command).

Wavenis implements two-way communication to manage acknowledgement and spontaneous alerts when necessary, as well as services to operate three communication modes. These are point-to-point, point-to-multipoint (broadcast, polling) and repeater or multihop mode where data is routed over multiple hops.

The supported network topologies include; star topology for simple networks, tree topology for networks with range extension needs, and mesh topology for reliable and flexible networks. The mesh topology of Wavenis has the same properties as the topology implemented in SmartMesh with features like self-organizing and self-healing algorithms.

In addition to fixed network monitoring, Wavenis also hold the possibility for walk-by monitoring. This way data from hard-to-reach sensors can be retrieved by the use of a handheld terminal.

4.1.2 Network Communication

The Wavenis protocol includes time-synchronized communication, and comprises both TDMA and CSMA-CA in their implementation. Unlike the SmartMesh technology, which uses TDMA for all communication, Wavenis only uses TDMA if feedback is requested after a broadcast message. When a child receives a message of this type it calculates its time slot using a pseudo-random sequence that depends on its PHY address. In case of conflicts, retransmission is handled according to a pseudo-random sequence [26].

Polling in a star network

This implementation of TDMA makes it possible to calculate maximum network size, depending on application requirements. The Wavenis technology overview gives an example of a network installed in star topology with a time-critical application requiring polling of all nodes once every second. It assumes that the requested data is HART command 9 with a resulting transmission time of 50 ms (indicating a payload size between 27 and 47 bytes) if the typical data rate is used (19.2 kbps). This would result in a maximum of 20 devices in the network. With maximum data rate (76.8 kbps) the number of devices can be increased to 80. These numbers are calculated assuming 100% path stability and no retransmissions, which is highly unlikely in a real industrial environment. To be on the safe side, at least 50% path stability should be expected. Hence, the above device numbers must be reduced to make the calculations apply to a real world network.

CSMA-CA

Wavenis implements CSMA-CA[2] as an optional mechanism to be used in applications with high risk of simultaneous transmissions. This indicates that CSMA-CA should be activated in most applications. The principle can be summarized as follows. A child senses the RF channel before transmission. If the channel is free, the child sends an RTS (Request To Send) to the parent. The parent manages potential conflicts and sends back a CTS (Clear To Send). When CTS is received by the child it immediately sends data. It is obvious that a trade-off exists here, between the increased network traffic and resistance to hidden terminals[3]. In a network with rapid reporting much can be gained, not only in performance, but also in terms of reduced current consumption due to avoidance of hidden node transmissions.

Normal Operation

When a synchronized fixed network is up and running, all nodes are in receive/standby mode for a programmable period ranging from 12.8ms to 12.8s (with 1.28s as its typical value). Reception time is only 500us if there is no energy on the channel. This time is extended to 1.6ms if there is energy but no coherent signal. In the case where a useful message is received reception time will be set to normal operation. Both parents and children may at all time initiate communication, the access time is then defined by the predefined period. To minimize the over-hearing phenomenon and to make the network more robust against interference, the receiving frequency channel is changed each period. The channel is changed following a pseudo-random sequence that depends on the PHY address. In addition, for a given parent, all children implement a delay that depends on the PHY address and clock of the parent. This is done to even further reduce over-hearing. Because parent and children "know" each other, the time shift between their 1.28s clocks is accounted for during initiation of radio-communication. To simplify the explanation, each parent-children group can be thought of as a cluster with its own dedicated delay and its own hop table. When communication is to be initiated, whether it is between two children, parent to child or the other way around, the transmitter sends a wake up frame followed by the data packet. Synchronized clocks and known hop tables and delays make the needed wake-up sequence short in duration (50ms in average). It is designed to cover the receive time slot of the target device(s). The concept of receive/standby mode is illustrated in figure 4.1.

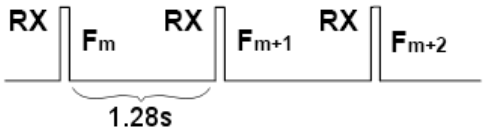


Figure 4.1: Default operation for the Wavenis devices [26]

[2]See section 2.7.3 to refresh the details of CSMA.

[3]The problem can occur when children nodes, out of radio range, are communicating with the same parent. When the motes cannot sense each other they may transmit at the same time, resulting in collisions (also explained in section 2.7.3).

4.1.3 Synchronization

All parent nodes in the network are responsible for synchronizing their children periodically. This is done by letting the parents send a short time synchronization beacon once every 20 minutes. When children receive the synchronization beacon, they align their own clocks to the parent clock. According to the technical overview, no ACK is required. Instead, if a child misses a synch beacon, it will enlarge its receiving time window to try to catch the next one. This process will continue every 20 minutes, until a successful re-synch is accomplished or the widest time window is reach. If the latter is the case, the child is self-declared "out of the network" while the application layer determines whether or not a new route needs to be found (self-healing) or if the device has to remain in standby mode.

The 20 minute timer will only be initiated when a device becomes "parent" for the first time. From then on the parent transmits synchronisation beacons to its children as described above. As a result, all 20 minute parent timers are fully asynchronous because device connection to a parent solely depends on the time of installation.

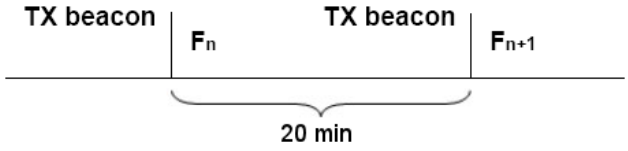


Figure 4.2: Network synchronization [26]

4.1.4 Joining of a Wavenis Device

For fast assignment of a new mote in a fixed synchronized network, one possibility is to implement a dedicated connection mode in the application layer. According to Coronis, many of their customers and partners now support this. At the right time, for a predefined duration, all devices will switch from 1.28s with a pseudo-random hop table to a 400ms period on a dedicated connection channel only. This will allow new devices to enter the network fast and easy through the self-organizing algorithm. The mode may of course also be used for network maintenance upon request of the network administrator.

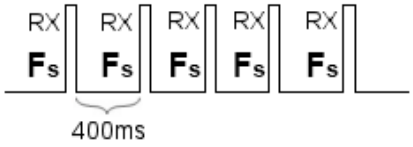


Figure 4.3: Dedicated connection mode [26]

In a network without a dedicated connection mode, as discussed above, the access time will be a little longer. A channel dedicated for connection can be sensed every 5 periods. This channel is programmable and is predefined by the application. When a new device (handheld devices, sensor devices, etc.) is trying to connect the network, it will first sense the expected

channel according to the pseudo-random sequence of hops. Then it will wait for the dedicated connection channel. As a result, to get deterministic access time, a preamble of typically 6.4s is transmitted before data transmission.

4.1.5 Transmission details

Wavenis combines FHSS with Forward Error Correction (FEC) and data interleaving mechanisms. The used FEC method, BCH (31, 21), is categorized as block coding with 1/3 data redundancy. Adding data interleaving, a way to arrange data in a non-contiguous way, gives data a good chance to be decoded correctly even with several bits in error. An illustration of the concept is depicted in figure 4.4.

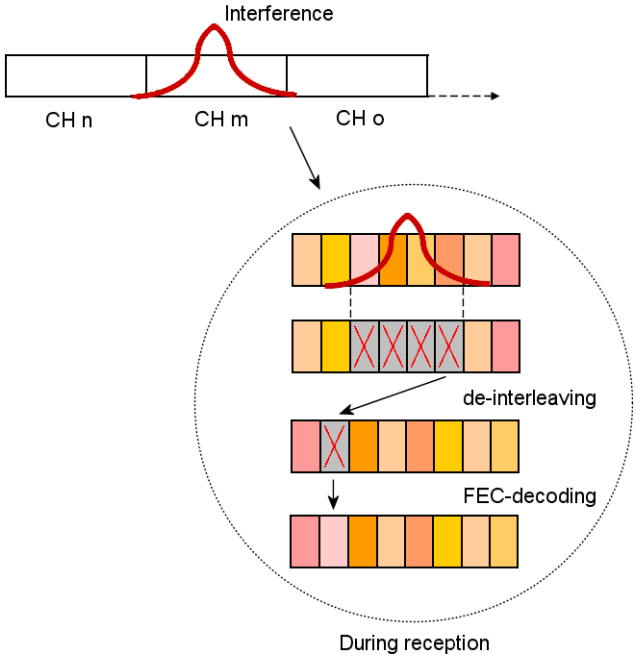


Figure 4.4: FHSS, FEC and data interleaving (modified from [26])

Wavenis implements fast FHSS with hops every 20 bits, following a pseudo-random hopping sequence. Because applications require different packet sizes, Wavenis defines data blocks of 32bytes. A data packet may consist of one or more such blocks. The first Wavenis block contains 5 bytes of payload, while the subsequent blocks may hold 21 bytes of payload (with the FEC accounted for). Before RF transmission, each byte is converted to 10 bits. Since the fast FHSS makes hops every 20 bits, this results in hops every 2 bytes. Hence, if you want to send 1 byte of payload, 32 bytes will be sent resulting in sixteen FHSS hops. With the typical data rate (19.2 kbps), this will lead to 16ms long transmission duration. If the payload is extended to 6 bytes, two blocks must be sent and the transmission duration will double.

In case of unsuccessful attempts (where no ACK/feedback are received when requested) Wavenis implements automatic retransmissions. After three such failed transmissions, an ”error” flag is delivered to the application layer.

4.1.6 Wavenis Protocol Specific Features by Layer

The Wavenis protocol implements several of the seven layers of the OSI-model, depending on the type of Wavenis device. The Host Controller Interface (HCI) is defined to be the layers above the network layer. In single CPU devices like; Wavecard, Waveflow, Wavesense, etc., these upper layers consist only of the application layer. Of these devices, Wavecards (in addition to Waveports) can also be used to communicate with a host in the form of a PC. The layers implemented in the host will then comprise all layers above the network layer. More information about the different Wavenis devices can be found on the Coronis website, as referred to earlier. The layer specific features of the Wavenis protocol are shown in figure 4.5.

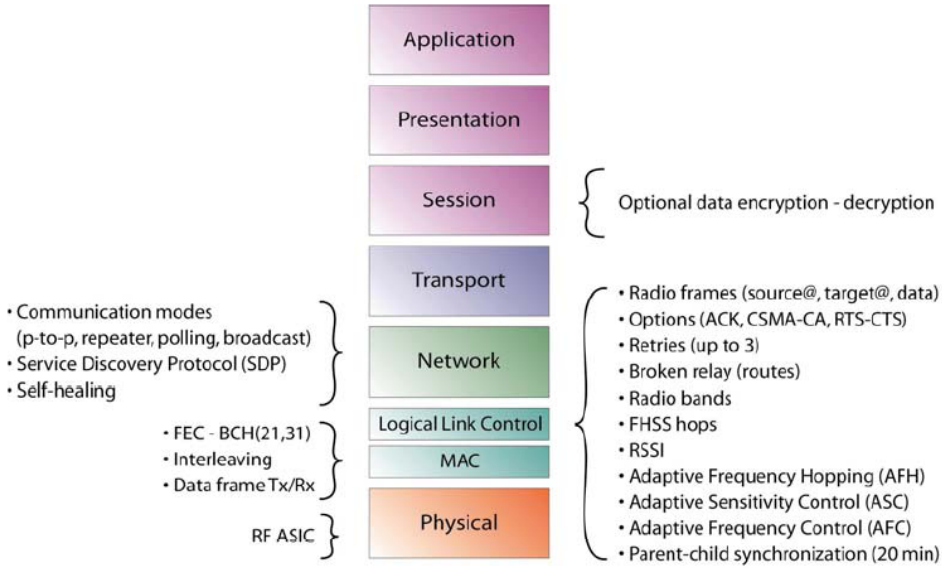


Figure 4.5: Wavenis protocol specific features by layer [26])

4.1.7 Discussion

Wavenis implements many good features for reliable communication. Their implementation of TDMA on the other hand, does not have finite timeslots and is only used when feedback is requested after broadcast messages. An implemented (and optional) version of CSMA-CA is used in all other communication scenarios. This means that polling with broadcast messages must be used to get a fully deterministic network. Broadcasting will lead to increased network traffic resulting in increased average power consumption. The minimum packet size will even further increase consumption if small packets are sent, due to the extra time transmitters must be turned on. However, this should be considered with pure TDMA in mind. In such implementation power will be consumed for each link in the frame, independent of data being transmitted or not. Referring to the consumption of motes in the specific time slots they are present (and in receive mode). Note that Wavenis is limited to a maximum of four hops, which may reduce the network range compared to SmartMesh-XR.

4.2 SensiNet from Sensicast

Sensicast Systems Inc.[27] was formed in September 2002, the same founding year as Dust Networks. The two companies share more than founding dates. For instance they both work with the development of intelligent wireless sensor network systems. And like all providers of these products they strive to develop the most reliable, low power and low cost systems possible. To achieve these goals, Sensicast uses some of the same technology previously discussed for both SmartMesh and Wavenis. The mesh topology and features such as frequency hopping are needed to obtain a reliable and robust network. Sensicast includes these features in its SensiNet protocol, which will now be presented.

4.2.1 SensiNet Components and Network Structure

The low level SensiNet platform is built on the base of the IEEE 802.15.4 standard [10]. In common with the SmartMesh protocol, SensiNet rebuilds the MAC layer to handle frequency hopping and implements features for extended determinism (like more guaranteed time slots (GTS)). The SensiNet implementation is, however, more conservative than in the SmartMesh protocol. Sensicast has kept the original 802.15.4 components and uses the beacon enabled strategy along with CSMA/CA in their design. Their main goal is more in the area of a standard extension, than the making of a completely new protocol. On top of the MAC layer, SensiNet implements a robust network layer to handle the mesh topology and other network features.

As illustrated in figure 4.6, SensiNet includes three different devices in their network. In the original definition of IEEE 802.15.4, the star nodes would correspond to end devices while the mesh nodes would be referred to as coordinators. The bridge/gateway node is similar to the PAN coordinator. Communication between the mesh nodes and their star nodes is as per the 802.15.4 standard. This means that the star nodes are reduced functional devices (RFD), unable to forward or receive data from other star nodes. Recall that in the SmartMesh network, all motes can send and receive from all network motes within range. Communication between mesh nodes are handled by SensiNet. Also through SensiNet, star nodes with radio connectivity to multiple mesh nodes will set up redundant links to secondary mesh nodes.

Sensicast provide both complete solutions as well as modules that sensor OEMs may implement directly into their designs. Like the SmartMesh motes a generic star node has a sensor interface port and digital and analog inputs and outputs. Examples of such nodes are Sensicast OAS (Object Alarm System) and Sensicast EMS (Environmental Management System).

The mesh nodes are Full Functional Devices (FFD) that can be used as coordinators to repeat or route data between the star nodes and a Host. They transmit messages from star nodes to other mesh nodes or to bridge nodes. Mesh nodes incorporate features such as memory for message buffers and routing tables, support for DC power, a serial interface port and digital and analog inputs and outputs. To extend radio range and penetrate noisy RF spectrums, the mesh nodes are implemented with increased transmission power. This results in stronger signals and the need for fewer overall devices, but it also leads to increased power consumption. Hence, it may be necessary with stationary power in a crowded network.

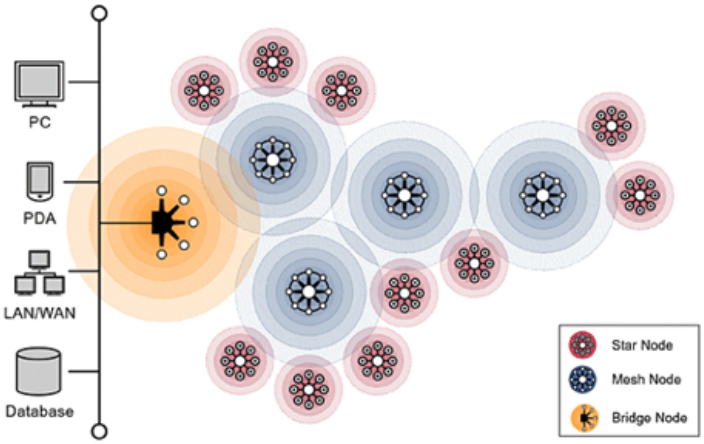


Figure 4.6: The SensiNet Network [28]

4.2.2 Network Communication

Network formation

SensiNet comprises self-formation, self configuration and self-healing. These features are covered by all SmartMesh competitors in addition to the SmartMesh protocol itself. In order for nodes to join the network, they must first discover the frequency-hopping sequence of the new network. The bridge node and any mesh nodes that have already joined the network transmit a beacon that broadcasts this sequence. Nodes that have not yet joined a network can listen until they receive this beacon. Once a node acquires the frequency hopping sequence, it is synchronized with the network and can listen to all neighbouring transmissions. The node can then send out a request to join the network. Each node then selects two parent mesh nodes, a primary and a secondary, if available. This process reminds a lot of the join sequence of the SmartMesh Network. Switch beacons with ADV packets and the process is much like the one implemented in the SmartMesh protocol. However, to increase the join time of new motes, SensiNet makes use of CSMA/CA for this process.

Path assignment

When a node establishes communication with the network, each packet is delivered with an assessment of received signal strength. Additional connectivity assessments are periodically made comparing the number of received packets to the number of expected packets. Nodes are assigned to the their primary and secondary mesh nodes based on these assessments of connectivity, and thanks to the periodic reassessments data is always sent via the highest-quality communication link.

Normal operation

During normal operation, nodes periodically send ”heartbeats” to their primary and secondary mesh nodes. Heartbeats serve several purposes and are an important feature for star motes to conserve battery power. Star motes are normally in sleep mode and only wake up when they

are to send heartbeats or data. While motes are sleeping their parent mesh nodes may have received data from the host, such as reconfiguration commands or notification that a firmware upgrade is available for download. Heartbeats provide a way to poll for these pending-store-and-forward messages. In addition they are used to indicate whether or not motes are working properly. Heartbeats are sent at fixed intervals, and if a parent mesh node fails to hear from an expected child, either mesh or star node, it will report the issue to the host. The heartbeats also generate packets that can be used for connectivity assessments. This relieves the nodes from additional transmissions and resulting power overhead. Heartbeat intervals and other node settings are user configurable, and is handled remotely through the Host.

Like all protocols stressing high reliability, SensiNet implements ACKs and re-transmissions for all network communication.

4.2.3 Reliability Issues

SensiNet enables automatic self-reconfiguration of the network in response to changing conditions. As the network is formed, routing tables are maintained by the host, allowing packets to be rerouted if necessary and the network to be reconfigured or rebuilt in the case of significant network changes. Such changes include changed mote connectivity, network topology changes (like the loss of a mesh node) or the loss of all network connectivity. To handle movement of motes and changes in connectivity conditions, routing tables may be updated dynamically by the host.

Connectivity Problems

Nodes can automatically search for new parents. This happens if connectivity problems prevent consistent communication between a node and its primary and secondary mesh nodes. The host updates the routing tables on request from the node. If reasonable connectivity is maintained, a star or mesh node will only switch to new parent nodes if requested to do so by the host. This is similar to the behaviour of the SmartMesh network where the Manager only intervenes and creates new paths if the retries/timeout threshold is breached.

Loss of Connectivity

In the event that a node loses its connectivity to the network entirely, the node can autonomously rejoin the network. The joining process is then identical to that described in section 4.2.2.

Rebuilding of the Network

When host detects numerous communication failures, it can autonomously decide to rebuild the network. The Host discards existing routing tables, allows network communication to time out, and reforms the network as if it were a new network. Mesh nodes and star nodes will detect the timeout and start a search for new network to join.

4.2.4 Discussion

The SensiNet protocol has good solutions for maintaining a reliable network. In contrast to the SmartMesh protocol, SensiNet has kept most of the original 802.15.4 features, and

implements both CSMA/CA and a beacon enabled strategy. The former is kept to speed up the joining process. Recall that the SmartMesh protocol also adds features for faster join times. With a fast (short) and simultaneous running frame (frame 1), join times may be considerably decreased.

SensiNet has also kept the different components of 802.15.4. As a result, their star nodes can only be used as end devices without routing capabilities. If routing capabilities is wanted for all motes, there is still a possibility to install a network consisting solely of mesh nodes. But this would increase the overall power consumption in the network. In contrast, all smart mesh motes may be used as routers.

Chapter 5

Experimental Background and Objectives

This chapter contains background information and discusses the main objectives of the project experiments. Planned configurations, environmental challenges and available equipment are some of the topics covered. All experiments are performed based on these sections.

5.1 Equipment and Software

For this project there was bought a SmartMesh Evaluation Kit from Dust Networks, decribed in appendix A.1. The kit contains twelve motes, three sensors and one Manager. For communication with the Manager the kit also comes with a client application, the SmartMesh Console 1.5-148. This application can be used for network configuration as well as for data collection and control. It also gathers detailed network statistics every 15 minutes, containing latency times, link stability and network reliability. All paths and connections between motes and Manager can be viewed trough its Graphical User Interface (GUI). Pictures will be found in the following chapters.

5.2 Office Experiments at ABB

The experiments performed at ABB Corporate Research Center include tests on all possible parameter settings and analysis of the resulting statistics. There will also be experiments regarding power consumption and different network topologies. However, the main emphasis is on the reliable mesh topology. All twelve motes are used in these experiments to get the widest network possible. The motes will be placed at random, letting the network form without interference. More information about the installation can be found in the following chapter, while the actual experimental configurations and procedures are described in chapter 8. A possible mote scenario is depicted in figure 5.1. The figure shows a reliable mesh network with parents marked with the letter, P, and children marked with a C. Arrows indicate paths between the motes.

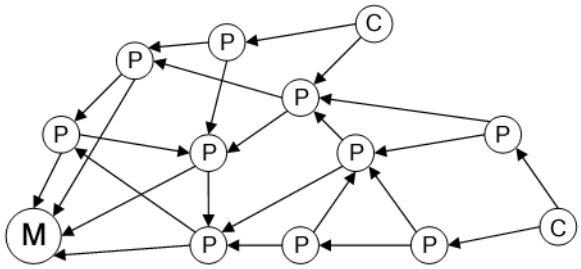


Figure 5.1: Possible mesh configuration

5.2.1 Tests on Power Consumption

To get reliable power consumption measurements, the network must be installed in a configuration where all behavior can be accounted for. This is ensured by installing the motes in a linear topology, leaving only one possible route from any mote to the Manager. Depending on a mote's position in the topology, it is now possible to measure how the power consumption increases when a mote routes date from other motes in addition to its own data. The topology may also be used to examine performance metrics like latency per hop.

A way to achieve the above topology is to place motes fare apart, making it impossible to form in a mesh topology. In the ABB facility this limited the network to four motes. Further extensions would only make the motes mesh together.

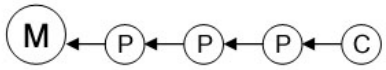


Figure 5.2: Linear topology

Later, it turned out that there was an alternative to this method, which made it possible to include more motes in the experiment and form the network in a range of topologies. The solution was to send special formation commands to the Manager by logging on to its command line interface (CLI), but this required a superuser password and of course some knowledge about the CLI commands. Fortunately, both the password and a CLI Commands Guide [18] were provided by Dust Networks in time to complete the experiments.

5.2.2 Environmental Challenges

The Norwegian ABB Corporate Research Center represents a typical office environment with 802.11b/g wireless LAN and office walls with windows as the greatest challenge for the SmartMesh network. Compared to a real industrial environment involving vibrating engines and reflecting metals, this may seem as a kind environment to test performance. However, curious people touching and moving motes, in addition to highly crowded hallways may indeed help to level these differences. The experiments in this project actually showed that an office environment may present a greater challenge than that of an industrial environment.

5.3 Industrial Experiment at Statoil

Statoil Research Center at Rotvoll in Trondheim makes it possible to test SmartMesh-XR in a real industrial environment. As an extra bonus, Statoil has a history when it comes to testing WSNs. Not long ago, an experiment was carried out regarding the performance of ZigBee. Installing SmartMesh-XR in an identical configuration with the same tasks and objectives provides a unique opportunity to compare the different technologies. This section will give an overview of both the previous experiment and the intended configuration of this project. The details concerning the actual installation and implementation of SmartMesh-XR will follow in the next chapter while the experimental procedures are explained in chapter 8. Results and comparison graphs are presented in chapter 9 and chapter 10 respectively.

5.3.1 Industrial ZigBee Test (IZT)

A previous wireless experiment performed at Statoil involved ZigBee as a tool to collect data from a real industrial process [33, 34]. The objective of this experiment was to compare the properties of ZigBee with traditional wired instrumentation and to see how well ZigBee would perform in an industrial environment. Statoil have many special constructions for testing purposes in their labs. Access to these constructions is restricted and limited, but the previous project team eventually managed to get clearance to modify a special ”wheel” for data collection. The ”wheel” is used in research on hydratization in pipes transporting oil. It is placed in a safety cell consisting of three explosion proof concrete walls and one glass wall pointing towards free air/ocean. Data collected from the wheel was retrieved from a torque sensor with an analog output signal ranging from 0-30 mA.

In addition to the ”wheel” a special tank construction was made to function as a combined demonstration- and training installation. The construction was named Ziggy to indicate the purpose to which it was built. Ziggy was originally designed by Gisle H. Bedin and was built by the apprentices Linda Wrangen and Eivind Utheim. It consists of a flow loop with two tanks, accumulator and heat element. For instrumentation purposes it is equipped with sensors for pressure, level, flow and temperature. For further illustrations, see the attached pictures in appendix A.3. From the available sensors, pressure and level was chosen for data collection. A single ZigBee node was placed for each sensor.

5.3.2 Industrial SmartMesh-XR Experiment

The final IZT installation consisted of seven nodes. As described in the previous section, three nodes were connected to sensors, while the remaining nodes functioned as routers to collect data from the ”wheel”. In the present project it will be stressed to make the SmartMesh experiment identical to this configuration. Figure 5.3 gives an overview of the planned configuration. Motes are given the same names as in the IZT. All motes marked with, R, are routers while, E, indicates ”end motes” with sensors attached. The Manager/gateway is marked in bold with the letter M.

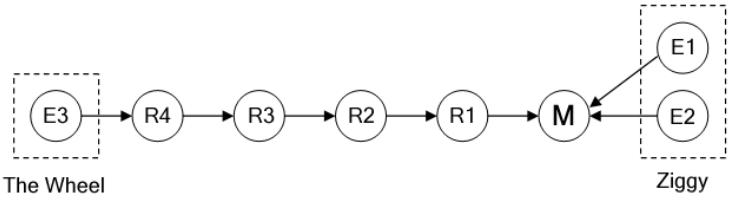


Figure 5.3: Planned SmartMesh configuration

5.3.3 Environmental Challenges

The greatest challenge for the industrial experiment is the safety cell. Within the thick concrete walls we find vibrating engines and metal equipment of various types. In addition, the mote placed in this room will be in constant motion, due to the placement on the rotating wheel. Some of the router motes are also placed great distances apart with several doors between them. See appendix A.3 for more information.

Chapter 6

Experimental Installation and Implementation

This chapter covers the physical installation of the network and the interfaces between motes and sensors. The measurement procedure regarding power consumption will be briefly explained and illustrating pictures will be given.

6.1 Office Experiments at ABB

6.1.1 Network Planning and Installation of Motes

Mote placement and installation were carried out as described in section 3.3.2. The network was installed to cover the largest area possible with the limited number of motes. To see what the network could really handle, mote placement was done on the edge to bad signals, but always with more than one possible neighbor. This must be done to get the most out of the mesh topology.

The final mote positions are shown in figure 6.1. Before these positions were found many different angles and positions were tried out. The SmartMesh Console gives a "helping hand" in this process by displaying the Received Signal Strength Indication (RSSI) value for each path. This gives an indication on where router-motes should be added, or which motes should be moved. It is also possible to see if a mote has less than two parents, if this is a fact, the same rule as above applies.

During installation motes were found to be very sensitive to nearby objects and the angle of the antenna. Mote placement in window frames can be used to illustrate the level of sensitivity. When motes were placed within the frame they were sometimes a few centimetres out of sight from nearby motes. In this situation, weak paths were often chosen to motes farther away. It should be mentioned that the Manager tries to come up with the shallowest network configuration possible, in terms of hop depth from a mote to the Manager. So, the reason for these observations might as well be that fewer hops were chosen over good RSSI values. In any case the instructions provided by Dust about antenna sight and placement should be taken seriously for a good result.

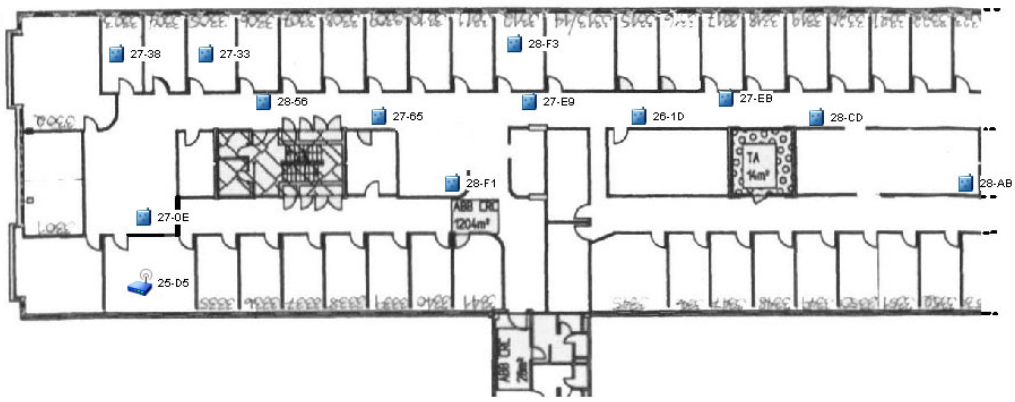


Figure 6.1: Overview of the mote placement at ABB

6.1.2 Procedure for Power Consumption Measurements

Power consumption was measured with a Tektronix oscilloscope featuring mean value calculation. A resistor with the value of 10 ohm were connected in series with the mote batteries, and the oscilloscope was used to measure the voltage over the resistor. Using ohm's law, I = U/R, it is now a simple task to find the current consumption of the mote. All measurements were performed over a minimum period of 6 minutes.

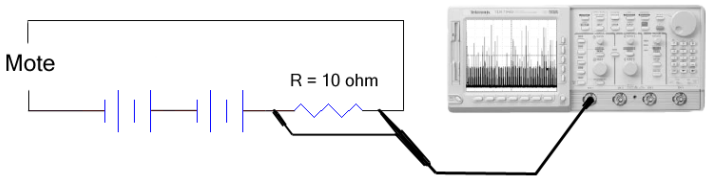


Figure 6.2: Setup for power consumption measurements

6.2 Industrial Experiment at Statoil

6.2.1 Sensor Interfaces

It has been mentioned that the Dust motes have seven external analog inputs. Five of these inputs are located on a HD-15 female connector. They are all unfiltered and assume signals ranging from 0 - 1.5V. The remaining inputs are found on a 4-position terminal block connector that accepts 12-22 AWG wire. It assumes 0-5 V inputs, and contains internal circuitry to filter 60 Hz noise. This property in combination with screw-connections, made these inputs preferable over the HD-15 connector.

There are three different external sensors involved in the experiment, all with different output values. To connect these sensors to the 0-5V inputs on the motes, interface cards had to be made to transform their output signals to suitable values. The different sensors and their respective output range are summarized in table 6.1.

Sensor location	Output	Sensor type
The wheel	0-30 mA	Torque Sensor
Ziggy (LT-101)	0-10 V	Level Transmitter
Ziggy (PT-312)	4-20 mA	Pressure Transmitter

Table 6.1: Overview of the sensors used in the experiment

Each interface card was designed for direct connection to a sensor. Thus, the combined resistance of the interface card functions as a loop resistor when connected to the sensor. The minimum value for a loop resistor is 250 ohm and it is therefore important to choose circuit resistors wisely. Resistors in the voltage divider are connected in parallel to keep the power low and to maintain the use of ordinary $\frac{1}{4}$ W resistors. All interface cards also include zener diodes connected as shunt-regulators to protect the motes from high voltage peaks. Even though the motes have their own input filters, a capacitor is connected in parallel to provide filtering of low frequency noise (-3db point just below 2 Hz). Simulation and analysis of all interfaces was done prior to installation. The design made for the pressure transmitter is depicted in figure 6.3. The rest of the interfaces can be viewed in appendix A.2.

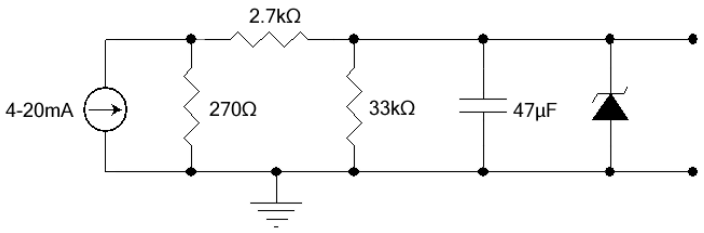


Figure 6.3: Interface design for the pressure transmitter

6.2.2 Mote placement and Installation

Mote placement

The IZT project group only documented approximate node locations without detailed information about node placement. Previous tests at ABB showed that even centimeters may have significant impact on the RSSI values. To make the experiment as accurate as possible, an assumption was made that the IZT nodes were installed with best possible Link Quality Indication (LQI). Because the RF environment is continuously changing, this task is harder than it seems. But with the use of common sense and the Dust instructions in mind, good RSSI values were achieved. All motes were installed at least one meter above the ground and it was stressed to obtain clear sight between motes. Installation close to- or on metal or other reflecting objects was avoided if possible.

The join process

Motes cycle into power-saving mode if they are out-of-range of neighboring motes and receive no transmissions during a ten minute period. In this mode, the mote conserves battery life

by powering on for only 100 ms each minute until it finds a neighboring mote. The joining process is slower in a linear topology than in a mesh network with few hops, and as a result many motes enter power-saving mode before they are joined in the network. To speed up this process, motes that had not yet joined the network were physically reset every ten minutes. In addition, all motes were given reporting profiles to increase the number of sent ADV packet.

Start-up procedure

As expected, the greatest installation challenge was to join the mote inside the safety cell. During the first twenty minutes after network start-up, the mote on the wheel was the only mote remaining unknown. One hour later the mote was still not found and it was decided to install an extra router mote inside the safety cell. This setup was left running during the night to give the motes plenty of time to join the network. The previous experiments at ABB showed that RF-conditions tend to improve at night-time. Possible reasons are less human- and WLAN traffic. If this is the case for the Statoil facility as well, the chances for a successful join will increase during this time. The wheel was left in a locked position (not moving) to further increase the chance of a successful join.

At dawn, the motes had still not joined the network. This indicated that the safety cell had too thick walls for the mote signals to penetrate. A thick explosion safe metal door is located on one of the concrete walls. The door, which had been shut during the night, was now opened, and the mote were physically reset. The extra router mote was removed. Fifteen minutes later, the mote had still not joined the network. A reset of the entire network was now performed.

Twenty minutes after the network reset, all motes were found, including the mote on the wheel. However, the connection to the wheel-mote turned out to be very unstable. After ten minutes the mote was once again unknown, only to get reported live again just a few minutes later. In fact, this behavior seemed to be the only stable aspect with the connection, repeating itself as time moved on.

To deal with the unstable behavior described above, extra router motes were placed between the wheel-mote and the closest router. Two hours were spent trying to find suitable locations for the new routers, but all new positions came up with the same result. They all joined the network, but no connections were made with the wheel-mote. It was also observed that motes connected the router on the other side of the concrete wall, instead of making paths to the router outside the open door. These observations indicated a faulty wheel-mote and a poorly placed router mote. To be on the safe side, both motes were replaced by new motes and their positions were slightly changed. The new wheel-mote was moved a few centimeters away from the metal on which it was placed and the router mote was placed 80 cm to the right of its previous location. Pictures illustrating the installation details can be found in appendix A.3 and on the attached CD.

After the final changes explained above, all motes were joined in the network. Stable connections were achieved and it was even possible to close the explosion safe door. When the wheel was set in motion (1 m/s), reliable connections were still maintained. This illustrates how sensitive the motes are to correct placement - even centimeters can make a huge difference. The installation process is further slowed down by the amount of time it takes to actually see if a change has made a difference. When network resets are required, the waiting time is

extended even further.

Figure 6.4: Overview of mote posititions for the industrial experiment

Chapter 7

Performance Metrics, Network Alarms and Configuration Parameters

This chapter presents performance metrics and definitions that are used for studying the performance of Dust Networks. In addition to performance metrics, some of the most frequent alarms and configuration parameters are summarized, and may be used as a reference throughout the report.

7.1 Performance Metrics

Data reliability, path stability and data latency metrics are defined as documented in [15]. Network statistics are collected from each mote every 15 minutes, and used by the Manager to calculate these metrics. A more thorough explanation of the metrics will now follow;

Data Reliability - Measures the percentage of expected data packets that were actually received by Manager. For example, if the Manager expects to receive 200 packets and only receives 180, data reliability is 90%. Throughout the rest of the report, data reliability is referred to simply as reliability.

Data Latency - The average time (in milliseconds) required for a data packet to travel from the originating mote to Manager. As with data reliability, data latency is often used without the word, data, in front.

Path Stability - Measures mote-to-mote transmissions. It is the percentage of data packets that have successfully reached their destination. If a transmitting mote does not receive an acknowledgement, it resends on an alternate path. Due to the unique benefits of mesh routing, data reliability can be 100% even with very low path stability.

Failed packets - Denoted either as *Pk Fail A/B* or *Pk Fail B/A* in the SmartMesh Console. It measures the number of packets transmitted by mote A which were not received by mote B (or transmitted by B and not received by A). This data is used in the Manager's calculations of path stability. Packets may fail for a number of reasons, such as:

- The sending mote did not receive an acknowledgement from the receiving mote that the

packet was received.

- The packet was damaged (NACK in return).
- The mote's buffer was too full to receive the packet (NACK in return).

Lost Packets - Measures the number of data packets expected that were not received. The Manager uses the report interval in the mote profile to determine when packets are expected from the mote. Packets are considered lost if they are not received within the report interval. This is used by the Manager to calculate data reliability.

The remaining definitions are not defined in the Console Reference, but are important metrics used throughout this report.

Throughput - Represents the amount of data delivered per time unit. This is usually measured in bits per second (bps), but may also take the forms of bytes per second (Bps) or packets per second (pps) depending on the current context.

Starvation - Occur if some motes get priority over other motes, leaving no time for the low priority motes to send their data. In general, starvation means that a task/mote is infinitely delayed, but in the context of this report it is defined as a delay to the extent that data is no longer usable. This is not a factor in a TDMA based network as long as it is configured correctly. In contrast, a CSMA based network may experience starvation if traffic is high and some tasks have more frequent reporting than others.

Bandwidth - When talking about the SmartMesh network, bandwidth is synonymous to available links in the frame. Assigning additional links to a mote enables that mote to make more transmissions per frame and thus higher bandwidth.

7.2 Network Alarms

As mentioned in the previous section, one of the Manager's tasks involves calculation of performance metrics based on the statistics it gathers from each mote. When calculated, it compares these metrics with the thresholds that are set for the network in the service level agreement (SLA) and triggers alarms when thresholds are not met. The SLA defines the quality of service that is expected from a network and comprises reliability, path stability, latency and battery lifetime. The respective alarms are denoted; *slaReliability*, *slaStability*, *slaLatency* and *batteryLow*.

In addition to the SLA alarms, which are settable in the SmartMesh Console, there are two remaining alarms that should be mentioned. The alarm, *moteDown*, refers to the scenario where the Manager is no longer able to communicate with a mote, or that the Manager is unable to locate the mote. The last alarm, *moteBandwidth*, indicates that more links are needed than the Manager can provide, given the network frame length.

7.3 Configuration Parameters

The configuration parameters have already been discussed, the information given in this section is only a summary of the basics and some abbreviations for the most frequently used parameters. These are used throughout the rest of this report.

Frame length - Sets the pace of the network. A short frame makes faster reporting possible and reduces network latency, but it also leads to higher power consumption. These tradeoffs must be thoroughly discussed upon network integration. Dust Networks recommend a frame length three times the network motes. Throughout the report, this is referred to as an *x3* frame length. Further, an *x4* is a frame length is four times the number of motes and so on.

Report interval - Sets the interval at which a mote is to send data to the Manager. The report interval must never be lower than the frame length, if it is, we risk buffer overrun resulting in lost samples. It is possible to avoid this by adding more links for a mote per frame, but this will make the network integration a lot more complex. Dust Network recommends a report interval three times the frame length. This is referred to as an *x3* report interval throughout the report.

Sample interval - The interval at which data is to be sampled. The maximum sample interval is found by dividing the report interval on the number of samples per report.

Chapter 8

Experimental Setup

This chapter will explain the configuration of the various experiments and their purpose. The results from each experiment can be viewed in the next chapter.

8.1 Office Experiments at ABB

The experiments performed at ABB comprise seven experiments, each including tests of their own. These tests involve changes of different parameter settings. The various settings are always documented in the test procedure sections and sometimes illustrated in the same table as the initial profile settings. Note that all changed settings are indicated with arrows in the performance graphs (presented in chapter 9) and thus it is only necessary to look back in this chapter for additional details.

8.1.1 Experimental Set One - Recommended Configuration

In the first test configuration the recommended guidelines from Dust will be followed. There will be made two new profiles for this purpose. One to handle ordinary motes, temp profile, and one to take care of the motes with external sensors attached, sensor profile. The name temp profile is used on ordinary motes to indicate that they are sending data samples from their built-in temperature sensor. Following Dust guidelines, the frame length for the system is set to be three times the number of motes, which gives us 36 slots (1125 ms). Also notice the report intervals, which are set to approximately three times the frame size. This is recommended to ensure a reliable network (there will be tests on this subject later). The number of samples per report is set to its maximum.

Profile Calculations

The maximum number of samples a mote may generate per packet/report is 144 bits. Knowing that a sample from the analog channel consists of 12 bits and that 1 bit is required for a digital sample, it is an easy task to calculate the maximum number of samples per packet. In the case of the sensor profile below, 5 analog channels and 2 digital channels are in use, resulting in a maximum of 2 samples per report. Next, it is time to decide what report interval that is appropriate for the profile. Following Dust guidelines the minimum interval is about 3 times the frame length, resulting in reporting every 3375 ms in this particular case. Dividing the report interval on the maximum number of samples per report provides you with the maximum

sample interval. Sampling slower than this will make some samples miss the transmission. To avoid this problem, the SmartMesh Console is programmed in such way that the report interval is set automatically based on the max sample interval and max samples per report. In the current experimental set, sample intervals were set to 1650 ms and 280 ms for the sensor profile and temp profile respectively. This resulted in a 3300 ms report interval for the sensor profile and 3360 ms for the temp profile. The calculated profiles below are only guidelines.

`System Settings`				
Frame length	:	3 x (number of motes) = 3 x 12	=	36 slots
Frame length in ms	:	slots x 31.25 ms = 36 x 31.25 ms	=	1125 ms
`Sensor Profile`				
Max samples per report	:	144 / (5 analog ch. x 12 + 2 digital ch.)	=	2 samples
Max sample interval	:	3 x 1125 ms / 2 samples	=	1687 ms
Report interval	:	3 x (frame length in ms) = 3 x 1125 ms	=	3375 ms
`Temp Profile`				
Max samples per report	:	144 / (1 analog channel x 12)	=	12 samples
Max sample interval	:	3 x 1125 ms / 12 samples	=	281 ms
Report interval	:	3 x (frame length in ms) = 3 x 1125 ms	=	3375 ms

Purpose

The purpose with the first test is to get an overview of the network statistics when recommended parameter settings are being used. The most interesting statistics will be latency, reliability and the link stability.

Test Procedure

At first the ordinary network behaviour will be logged. When enough results are collected it is time for more advanced testing. A mote will be removed from the network to see how the statistics are affected. At this point it will be interesting to see if the system will stay reliable or not, and if unreliability is the case, check how much time will pass before the network is once again reliable.

Expected Results

When running with the recommended configuration settings it is expected that the network will be close to 100% reliable. Every mote is installed to have at least two parents, only a mote with direct connection to the Manager may be allowed only one path. This should ensure maintained reliability when a mote is disconnected.

8.1.2 Experimental Set Two - Rapid data collection

In the second test the focus will be on rapid reporting with limited amount of data. The frame length will still be set at 1125 ms, whilst the other configuration parameters can be read in table 8.1. Setting the number of samples per report to one will result in a sample interval identical to the report interval. As explained in chapter 7, the report interval cannot be longer than the frame length. This would have led to backlog in the motes, and eventually failure when a mote is out of memory. Therefore the sample and report intervals are set to 1200 ms.

Profiles	**Sensor profile**	**Temp profile**
Enabled analog channels	3, 4, 5, 6, t	t
Enabled digital channels	1, 5	none
Samples per report	1 sample	1 sample
Sample interval	1200 ms	1200 ms
Report interval	1200 ms	1200 ms

Table 8.1: Initial profile settings used for the second experimental set

Purpose

The purpose with this test is to see how fast it is possible to get data from the motes, without making the system unreliable. To minimize the risk of data loss only a minimum of data is sent. An important part of this test is also to find the reset-time of the system. In industry, time is money, and in some cases down-time of a system could also lead to dangerous situations. In the end of the test it will also be looked at how the network handles mote-failures with the new settings.

Test Procedure

The new profiles will be applied to the motes while the network is running. A reset of the system will then indicate the beginning of this experiment. Start-up time will be noted before the network is left alone. There will be no interference while its behaviour is studied over time. When enough statistics are collected, it will be tested how the network reacts when a mote is disconnected. The mote will then be joined and the join-time will be observed. In the end, the sample time will be adjusted to see if we can achieve full reliability if this is not already the case. The first adjustment will be to increase the sample/report interval by two. Notice that this is still below the recommended threshold (three times frame length).

Expected Results

As long as the link stability in the network is good and the packet sizes are as small as possible there is a good chance that the network might be reliable even at this pace. The restart-time is expected to be approximately the same as with other settings since the frame length is left unchanged. The disconnection of motes might lead to more trouble in this configuration because of more traffic in the network. Increased latency is also expected.

8.1.3 Experimental Set Three - Shorter frame length

The previous tests were performed with a recommended frame length. This test will check the network behaviour when a frame length below the recommended threshold (number of nodes x 3) is configured for the system. More specific, a frame length with a value two times the number of nodes (750ms). A shorter frame length should lead to less latency, increased power consumption and make it possible to use lower report intervals. In this test all nodes will be sending maximum payload in each packet.

`System Settings`				
Frame length	:	2 x (number of motes) = 2 x 12	=	24 slots
Frame length in ms	:	slots x 31.25 ms = 24 x 31.25 ms	=	750 ms
`Max Profile`				
Report interval	:	3 x (frame length in ms) = 3 x 750 ms	=	2250 ms
Max samples per report	:	144 / (5 analog ch. x 12 + 8 digital ch.)	=	2 samples
Max sample interval	:	Report interval / 2 samples	=	1125 ms

In the above calculations we found the recommended settings for the new frame length. Notice that the frame now only contains 24 slots, which in turn means fewer links for the network motes. Instead of using the new calculated report interval the last tested interval from the previous experimental set was used - sample interval: 1200 ms and report interval: 2400 ms. This makes it easier to compare the results from the two experimental sets.

Purpose

Check the network behaviour with a frame length below the recommendations.

Test Procedure

The network will be restarted with the new configuration and set to run over a weekend. Results will then be analysed and new settings like reduced report interval will be tested.

Expected results

A shorter frame length should in theory result in less latency.

8.1.4 Experimental Set Four - Longer Frame Length

In the previous test configuration we looked at a short frame length. This test will focus on the opposite. A frame length four times the number of motes will be used.

`System Settings`				
Frame length	:	4 x (number of motes) = 4 x 12	=	48 slots
Frame length in ms	:	slots x 31.25 ms = 48 x 31.25 ms	=	1500 ms
`Max Profile 2`				
Report interval	:	3 x (frame length in ms) = 3 x 1500 ms	=	4500 ms
Max samples per report	:	144 / (5 analog ch. x 12 + 8 digital ch.)	=	2 samples
Max sample interval	:	Report interval / 2 samples	=	2250 ms

Purpose

Check the network behaviour with a long frame length.

Test Procedure

The network will be restarted with the new configuration and results will be observed.

Expected Results

In theory, latency should increase with the use of a longer frame length.

8.1.5 Experimental Set Five - Linear/Multi-Hop Topology

The linear or multihop topology guarantees that data follows a certain route and is thus great for tests regarding hop dependent latency. To make sure that the network is connected in this topology, the Manager was configured through its Command Level Interface (CLI) prior to formation. The CLI contains a method, ppath, to establish or delete persistent paths between two motes. With parameters consisting of MAC addresses and the optional parameter, *only*, this and only this path will be made between the two specified motes. Adding two persistent paths for all motes, one path to a parent and one to a child, makes it is possible to obtain a linear topology with several motes.

In the integrator guide [16] provided by Dust, it is recommended to install multiple networks if a mote is more than ten hops away. Hence, this test will be limited to a network consisting of 10 motes. Since all motes are sending data, the recommended report interval (motes x 3) will be used during all tests.

Note that this test is performed under optimal conditions with all motes located just a few meters away from the Manager. This will result in excellent RSSI values and should result in 100% path stability. A best case scenario that is hard to achieve in a real network.

Profiles	**Test 1**	**Test 2**	**Test 3**
Enabled analog channels	1, t	1, t	1, t
Enabled digital channels	none	none	none
Samples per report	6 samples	6 samples	6 samples
Sample interval	218 ms	437 ms	874 ms
Report interval	1311 ms	2622 ms	5244 ms
Frame length	437 ms	874 ms	1750 ms

Table 8.2: Test Profile for the fifth experimental set

Purpose

Latency varies depending on frame length, RF environment and number of hops from a mote to the manager. The purpose of this experiment is to find out exactly what latency to expect depending on route and frame length.

Test Procedure

The network will be started with the minimum frame length. When enough data is collected a new frame length will be initiated, twice as long as the original frame. The final frame length in the experiment will be four times the minimum frame length.

Expected Results

Latency will increase with extended frame lengths and more hops from a mote to the manager. The question is not if, but how much.

8.1.6 Experimental Set Six - Star topology

The fact that all motes are just one hop away in a star network makes this the fastest and less latency demanding topology. Minimum frame length will be used during all tests to achieve the most rapid reporting possible. Similar to experimental set five, this experiment is performed with the ppath command and optimal mote locations - no more than a meter away from the Manager. This is again a best case scenario that would be hard to obtain in a real world network. Bear this in mind when the results are examined.

Profiles	**Test 1**	**Test 2**	**Test 3**
Enabled analog channels	1, t	1, t	1, t
Enabled digital channels	none	none	none
Samples per report	6 samples	6 samples	6 samples
Sample interval	250 ms	166 ms	83 ms
Report interval	1500 ms	1000 ms	500 ms
Frame length	500 ms (16 slots)		

Table 8.3: Test Profile for the sixth experimental set

Purpose

The star topology makes it possible to find the lowest latency for a twelve mote network. The main goal of this experiment is to find the lowest achievable average latency for such network and compare the results with other topologies.

Test Procedure

To find the minimum report interval, the report interval will be decreased until a reliable network is no longer maintainable.

Expected Results

The final result should indicate the lowest average latency possible to obtain in a network at this size.

8.1.7 Experimental Set Seven - Power Consumption

This experiment was performed along with experimental set five. Power measurements were performed every time a new node was connected to the network, averaging the power consumption for a minimum period of 5 minutes. Motes close to the Manager are expected to burn more power than motes farther out, and an extended frame length are expected to result in lower values for average power consumption. See table 8.3 in section 8.1.5 for details about the various tests.

8.2 Industrial Experiment at Statoil

The planned configuration and actual installation of the SmartMesh network were discussed in chapter 5 and 6 respectively. To summarize; it was managed to install the SmartMesh Network with the same number of motes and identical mote placement as in the previous experiment (IZT). This involved three nodes installed to collect data from sensors, while the remaining nodes were set up to route data from these nodes to a gateway. Two different profiles were made to handle the different tasks - one for data reporting and a ”no data” profile for the router motes. Like the IZT, this experiment comprises tests on latency, throughput and starvation. The results can be viewed in the following chapter while comparisons are carried out in chapter 10.

8.2.1 Main Experiment and Objectives

Maximum Throughput and Previous Experiments

Maximum throughput is found by adjusting the recommended settings downward, until a reliable network is no longer maintainable. Both frame size and report intervals must be reduced to find the highest report rate possible. In contrast to the ABB experiments, only three motes are used for data collection. As a result, the industrial experiment is most likely to handle more frequent reporting before congestion is encountered. Hence, the results from the ABB experiments may deviate from the results we achieve at Statoil.

Configuration Tradeoffs and Integrator Choices

Even though conditions and configuration is different in the present experiment, some knowledge from the ABB experiments may still be used. For example, when a choice has to be made regarding reduction in either frame length or report interval, it comes in handy to recall the tradeoffs between power consumption and reliability. The minimum frame length has the lowest latency and makes it possible to use a lower report interval, but it also has the highest power consumption. If a choice is made to lower the report interval, it is important to remember that a reduction below the recommended threshold may downgrade the reliability of the network. As an example, if an x1 report interval is used, no faulty transmissions can occur. If they do, the mote will eventually fill up its buffer, and stop generating samples until available buffer space again is available. This is what happens when there isn't enough bandwidth/links in the network. However, it is possible to use client commands and manually add extra links for such motes. The choice is up to the integrator.

Test Profile Calculations

When this experiment was performed, it was assumed that N+4 was the minimum frame length independent of network topology. During link analysis, however, it was found that this rule only applies to simple star networks. The theory was then tested as described in experimental set six where it turned out that even in a star network, a few slots must be added to the N+4 frame to connect all motes. There is of course an exception to this as well. In networks with many router motes there are assigned more links than necessary because the Manager assigns the same number of links regardless of mote profiles. This results in more available links for the motes that are actually sending data. Hence, in some cases N+4 may be sufficient for other topologies as well. For example, the network installed at Statoil. The configuration for the different experimental tests are summarized in table 8.4.

Profiles	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**
Enabled analog channels	1, t	1, t	1, t	1, t	1, t
Enabled digital channels	none	none	none	none	none
Samples per report	6	6	6	6	6
Sample interval	328 ms	172 ms	172 ms	114 ms	57 ms
Report interval	1971 ms	1314 ms	1032 ms	688 ms	344 ms
Frame length	657 ms (N x 3)		344 ms (N + 4 = 11 slots)		

Table 8.4: Industrial profile configuration

Purpose

Check the performance of SmartMesh-XR in an industrial environment. Find the maximum throughput and analyse other performance metrics at this point.

Test Procedure

The network will be restarted with recommended setting and adjusted downwards as the configuration table describes.

Expected Results

With only three reporting motes in the network the chances for congestion is slim, even at high report rates. The greatest challenge occurs when an x1 report interval is used. Motes with limited bandwidth/links, like single parent motes without children, must then rely on all their transmissions to be successful. This is very unlikely in the harsh environment where the wheel-mote is placed. If the bandwidth is not increased, this might affect the overall reliability. Thus, the lowest frame length in combination with the x2 report interval is expected to give the highest reliable report rate.

8.2.2 Starvation

One of the test procedures in the IZT tried to provoke starvation of a node. To achieve starvation some nodes were set to send data with increasing transmission rates, while one node was set to send data at a regular rate. As the transmission rates of the motes are increasing, it should be possible to notice stagnation in received packets from the node sending data at a regular rate.

Even though SmartMesh-XR is a fully deterministic network, it is possible to provoke similar behaviour with erroneous configuration settings. A congested network may very well lead to starvation of motes many hops away from the Manager. This will be illustrated with use of minimum frame length and an x1 report interval. If all motes are set to send data, network congestion is expected after a few retransmissions. Recall that the mote buffers only hold eight packets[1].

8.2.3 Mote Range

No specific tests are performed on mote range. However, network performance will indicate possible differences between the IZT nodes and the present motes. It is assumed that node installation in the IZT was done to get the network as shallow as possible, in term of hop depth. Identical mote placement will thus reveal differences in mote range.

[1] See the discussion chaper for more details.

Chapter 9

Experimental Results

This chapter presents the results given by the various experimental sets in the previous chapter. A detailed discussion of each result is carried out.

9.1 Results from the Office Experiments at ABB

The ABB experiments gave both expected and unexpected results. Correct installation and configuration were found to be critical for network performance. It should be mentioned that in a real industrial process, a few motes are most likely to be used as routers. This highly reduces the level of congestion in the network compared to the present results. The topics for each experiment are connectivity, reliability, latency and path stability.

9.1.1 Experimental Set One - Recommended Configuration

The first experimental set used a recommended report interval (frame x 3) and a recommended frame length (motes x 3). Recommended settings provide statistics that can be used as a basis for further comparison. The experiment also comprises network behaviour in the presence of mote failure. This was accomplished by disconnection of a parent mote.

Connectivity and Paths

Step 1: Behaviour over time
The first step in the test process was to let the network run without interference and study the natural behaviour of the system. When the Manager is powered up for the first time it starts a search for the motes/devices in the network. Paths between motes are chosen based on the RSSI value and the principle rule; always to make the network as shallow as possible. The word shallow is here used in terms of hop depth from a mote to the Manager. The final path choices of the Manager can be viewed in figure 9.1.

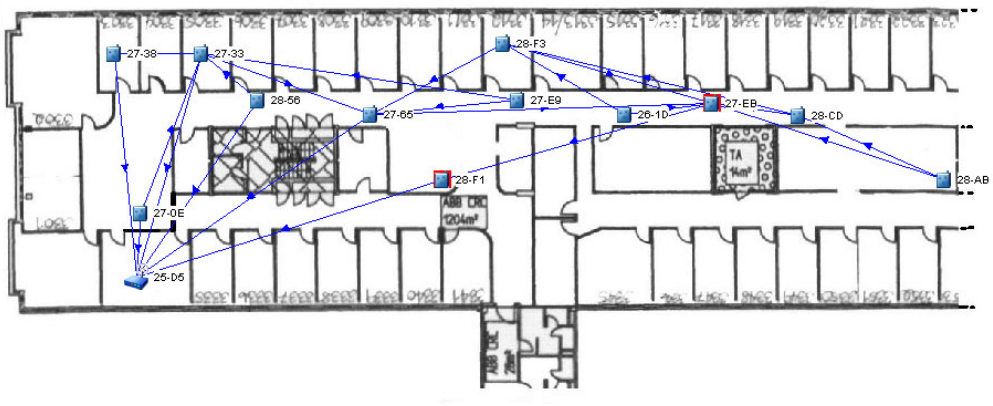


Figure 9.1: The connected network with paths indicated

The paths can of course vary over time. Many environmental factors are constantly changing the RSSI values of the different paths, and there is always a possibility that a mote might get damaged or somehow get blocked from the other motes. In cases like these the manager takes action and tries to find new paths and neighbours for the affected motes. The next step in the test process addresses this situation. A parent mote is here turned off to imitate a failed mote. This could indeed be the case in a real industrial environment, where there can arise situations much rougher than a coffee-trolley in the hallway.

Step 2: Disconnection of a mote
During the installation, all motes in the network had at least two parent motes. This was not the case when the second step was to be executed. In figure 9.1 it is clear that both mote 28-F1 and mote 27-EB only got one parent. The fact that these motes function as routers for other motes, makes the situation even worse if something should happen to one of them. To see what will happen in the worst case scenario, mote 28-F1 was removed from the system. The resulting new paths can be viewed in figure 9.2.

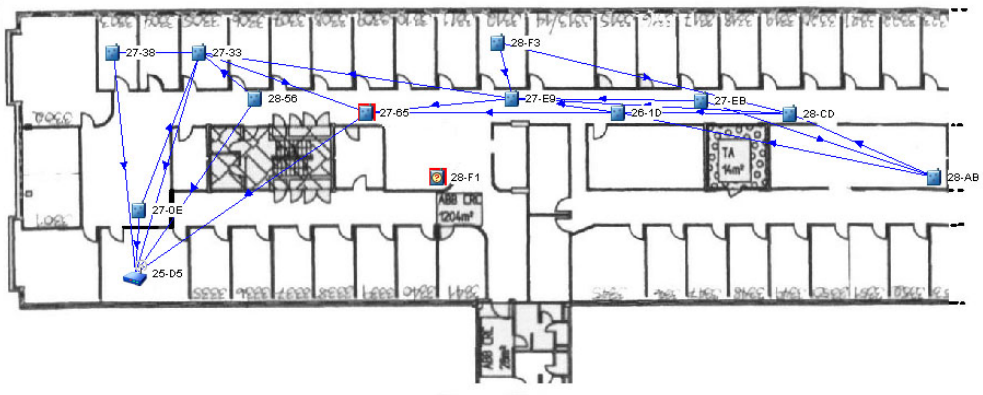


Figure 9.2: The network after the removal of mote 28-F1

When approaching an age[1] of 300 seconds an alarm was set telling that mote 28-F1 was down. This event also led to consequences for other motes in the network. Table 9.1 summarizes the alarms encountered after the disconnection of mote 28-F1, and up to the point where the network again reached 100% reliability[2].

Alarm Type	**Object Type**	**Impacted Mote(s)**	**Time**
Mote 28-F1 disconnected			14:26:00
Alarm opened	Mote down	28-F1	14:31:38
Alarm opened	Mote down	28-EB	14:31:38
Alarm closed	Mote down	27-EB	14:31:45
Alarm opened	Mote down	28-F3	14:33:08
Alarm closed	Mote down	28-F3	14:34:48
Alarm opened	Mote bandwidth	27-65	14:36:00
Alarm opened	Sla reliability	Network	14:45:01
The last two alarms are still not closed			15:54:22

Table 9.1: Summary of the alarms after the disconnection of mote 28-F1

The time for the disconnection of mote 28-F1 is only an approximate number compared to the time stamped alarms from the Manager. There might be deviation between the computer and the Manager clock. Because of this, the main attention should be put on the alarm times.

As a consequence of the disconnection of mote 28-F1, both mote 27-EB and mote 28-F3 are hidden from the Manager for a short period. When they are fully operational again an alarm about mote bandwidth present itself. This means that more links are needed than the Manager can provide, given the network frame length. The next alarm, sla reliability, indicates that the network reliability is below the minimum threshold set in the Service Level Agreement (SLA) settings. As will be explained in the next section this is actually not the case. More information about this will follow.

It should be mentioned that table 9.1 only shows the most important network alarms. In between the alarms there are events in the form of paths and links being created and deleted. There are also occurrences of changed path directions and joining motes. For a full description of network events, see the event log on the attached CD.

Reliability Statistics

Under normal conditions the reliability of the system was at 100%, as expected. The drop in reliability, observed around 2.30 pm, is due to the disconnection of mote 28-F1. It should be noted that the network is actually almost 100% reliable after just a few minutes. The reason for the low graph values is that the packet loss from the disconnected mote is still taken into account, even after the mote is found to be down. For more information about specific motes, see the attached CD.

[1] The number of seconds since the last packet was received.
[2] The network actually reached 100% before 3.54 pm. See the next section for more details.

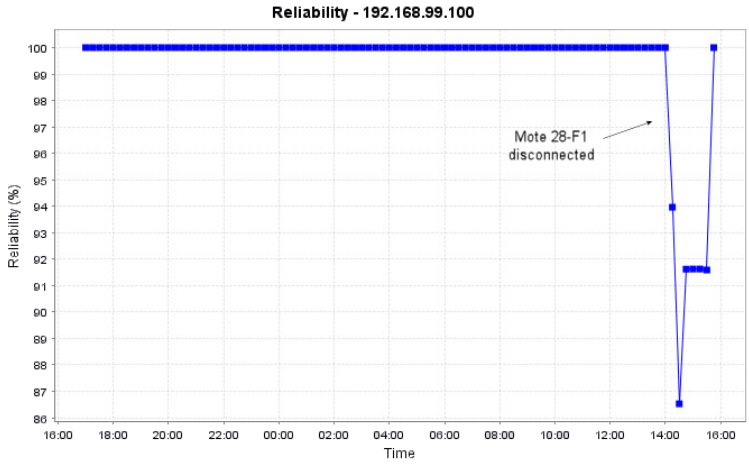


Figure 9.3: Average network reliability

The reliability statistics in figure 9.3 does not indicate which motes are suffering the most after the loss of mote 28-F1. An additional graph was made for this purpose, shown in figure 9.4. Analysis of the graph shows no unexpected behaviour. The children and the grandchildren of the disconnected mote are the only motes suffering packet loss. Mote 27-EB has the highest number of lost packets because it had the disconnected mote as its only parent. If the two parents rule had been followed there is a good chance that the network would have stayed reliable. The next experiment will look into this.

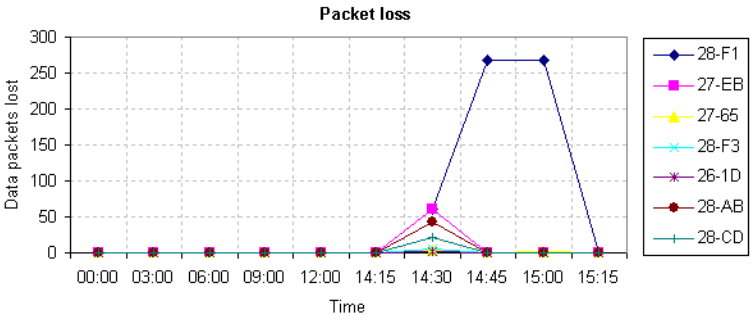


Figure 9.4: Packet loss for different motes

Latency Statistics

With recommended frame length and report intervals, average latency vary between 500 ms and 750 ms. If the latency statistics is compared with the statistics for path stability there should be noticed that there is a slight increase in latency when the path stability decreases. The reason for this increase in latency is that more retransmissions occur when links are turning bad or unstable. When mote 28-F1 is disconnected, a large peak is observed in the average latency. Data now has to travel a longer route, in terms of hops in the network.

Until new links and paths are assigned there will be huge delays compared to the original configuration.

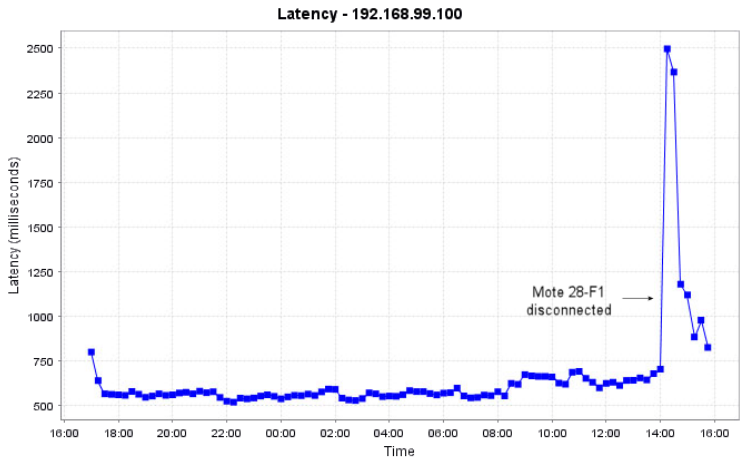
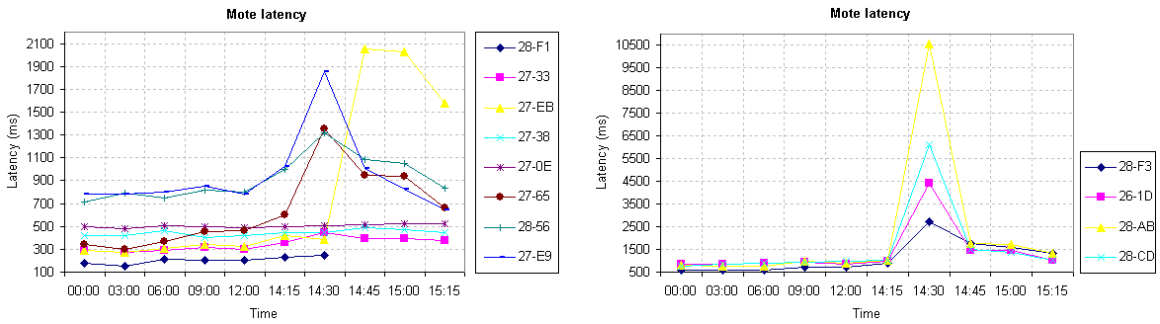


Figure 9.5: Average latency for the network

To see how latency is affected by hops and routes, two graphs (shown in figure 9.6) are provided to illustrate latency for different motes. They are divided into two graphs just to lower the scale for the motes with relatively low latency values. To make it easier to see the relationship between hops and latency, the graphs should be examined in combination with the connectivity figures in the first section. Note that the disconnection of mote 28-F1 resulted in a huge increase in latency even for motes close to the Manager.One reason for this is the increased traffic through the motes, since they now also have to route data from the motes previously sending data through the disconnected mote. The average network latency will close in on its original value when new links are assigned and new paths are made.



(a) Motes with low average latency

(b) Motes with high latency peaks

Figure 9.6: Mote latency

Path Stability Statistics

Signal strength is constantly changing for all motes. Proof of this can be found by looking at the maximum and minimum RSSI values for a mote. In some cases it is possible to encounter numbers deviating with more than -20dBm. An analysis of figure 9.7 shows relative stabile paths at night and a strong decreasing curve at dawn. The obvious explanation is that the motes are more exposed to multipath fading at daytime, in form of people walking by, doors being opened and so on. Increased traffic in the office WLAN may also be an influencing factor. Recall the discussion in section 2.6.1.

The removal of mote 28-F1 can bee viewed as an increase in average path stability. The reason for this behaviour might be new and better paths, and perhaps closure of paths with weak RSSI values.

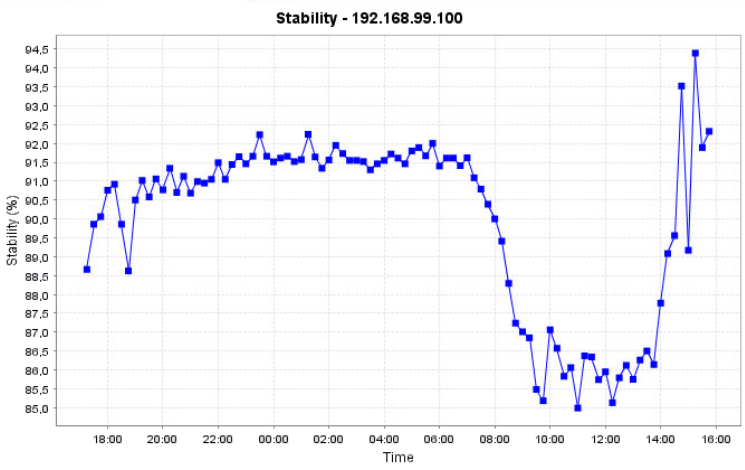


Figure 9.7: Average path stability in the network

9.1.2 Experimental Set Two - Rapid Data Collection

The main goal of this experimental was to find the fastest report rate possible while maintaining a reliable network. To obtain full reliability with an x1 report interval, it is necessary with full path stability. The first experimental set showed that even the best case path stability was below 100%. Hence, it is impossible to avoid network congestion without changes in network structure. The network was, however, left running to analyze network behaviour during congestion. Various tests were performed in this matter.

Connectivity and Paths

Step 1: Restart of the system
As described in the experimental setup, this experiment was initiated with a restart of the network. When all motes in the network were found they were connected as illustrated in figure 9.8. At this point the motes have still not formed a reliable mesh network, five motes are operating with only one parent. All of these are leaf motes or motes where its children

have redundant routes. If contact is lost with motes like these it will not affect other motes in the network. The motes will, however, get more paths as time moves on.

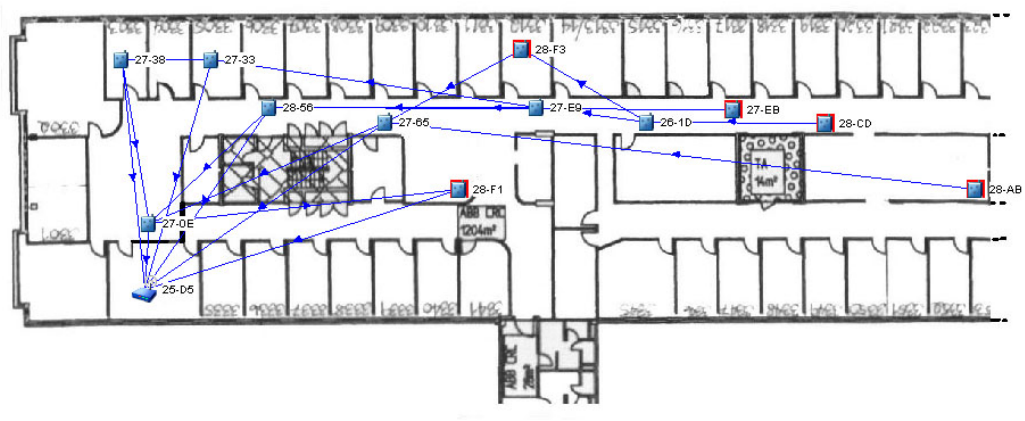


Figure 9.8: The network paths when all motes are located

In table 9.2 the mote discovery times can be viewed. Several motes joined and disconnected before they again tried to join. The table summarizes the final join and live times for the motes. For more details about the events during restart, see the event log on the attached CD.

Mote	Joined	Live
Manager	16:25	16:25
27-0E	16:29:21	16:29:22
27-33	16:29:23	16:29:24
27-38	16:29:24	16:29:26
28-56	16:29:28	16:29:28
27-E9	16:30:14	16:30:22
27-65	16:30:20	16:30:37
28-F3	16:30:31	16:30:37
26-1D	16:31:34	16:31:50
28-AB	16:34:22	16:34:54
28-F1	16:37:50	16:37:51
28-CD	16:41:48	16:41:49
27-EB	16:45:34	16:45:34

Table 9.2: Join and live times for motes in the restarted network

Step 2: Disconnection of a mote

After the reset of the system the network was left alone for seventeen hours. During this time new paths were assigned between the motes, as shown in figure 9.9. Only one mote is now operating with only one parent, making the network more resistant to mote failures.

A quick look at the average network statistics reveals that the network is already stressed

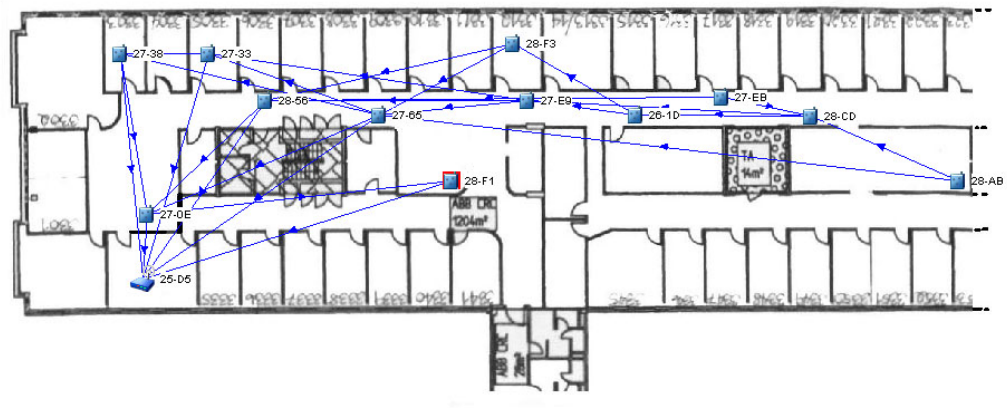


Figure 9.9: Network paths as displayed after 17 hours

beyond its capabilities. This results in a decrease in generated packets and higher latency. It is still of interest to see how the network will handle the loss of one of its most significant router motes. In contrast to the first experimental set, a mote will now be selected, which has only children with more than one parent. In theory this should not lead to decreased reliability for the other motes due to redundant routes. The mote named 27-65 was chosen for disconnection, the new paths are illustrated in figure 9.10.

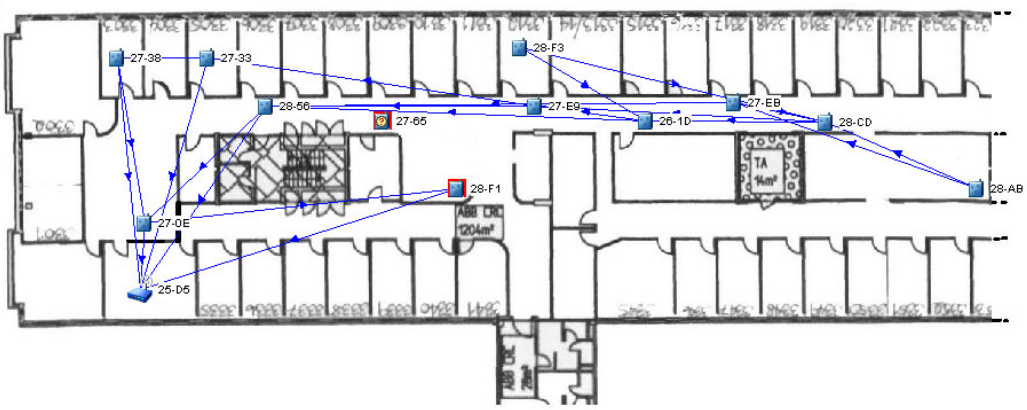


Figure 9.10: The network after the removal of mote 27-65

When 27-65 was disconnected it resulted in failure for other motes as well. The mote named 28-F3 was found to be down at 1.15 pm, three minutes after the discovery of 27-65 being lost. It was reported live again after five minutes. An interesting observation is that the network is now actually improving in performance, both in terms of latency and reliability. An explanation for this behaviour might be the new links available for the other motes, more stable paths and 1/12 less traffic in the network. It all points in the direction of the report interval being too rapid for the installed network, there is just not enough bandwidth. See the performance statistics for more details.

Step 3: Rejoining a mote
The disconnected mote is turned on again at 2.19 pm. After approximately five minutes it is discovered and rejoined in the network. The new paths in the network can be viewed in figure 9.11. For more detailed event descriptions during the joining process, see the attached CD.

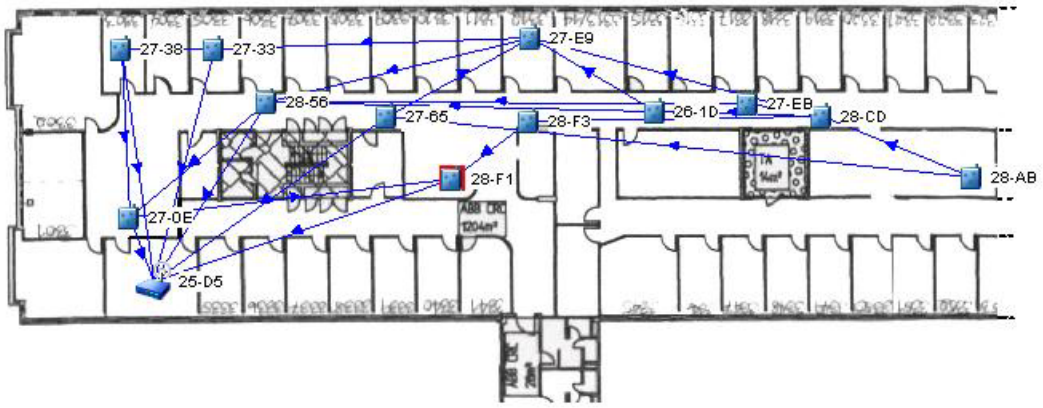


Figure 9.11: The network after the rejoining of mote 27-65

Reliability Statistics

The average reliability graph in figure 9.12 shows a clearly overloaded network. Between 5.45 pm and 0.30 am it is possible to see the Managers struggle to get a more reliable network. The sudden increases in reliability, observed at 9.00 pm and 0.15 am, is most likely due to better link assignments or the making of better and more stabile paths. If no retransmissions were necessary in the network we would achieve a much higher bandwidth, but this will seldom be the case in a real industrial environment. As with the first experiment, it is possible to see a change in statistics as the day begins. The main difference in this experiment compared to the first, is that the first experimental set could handle the extra bandwidth needed for retransmissions and maintained a reliable network. In this experiment, however, the reliability started to decrease at daytime.

The drop in reliability, observable at 1.00 pm, is due to the disconnection of mote 27-65. As encountered in the first experimental set, the packet loss from the disconnected mote is added even after the mote is found to be down. An interesting observation in this experiment is that reliability is actually increasing, even though we got the extra packet loss from the disconnected mote. The reason for this behaviour was mentioned in the previous section.

At 2.46 pm the report/sample interval was doubled. This event led to a steady increase in reliability until full reliability was achieved at 3.30 pm.

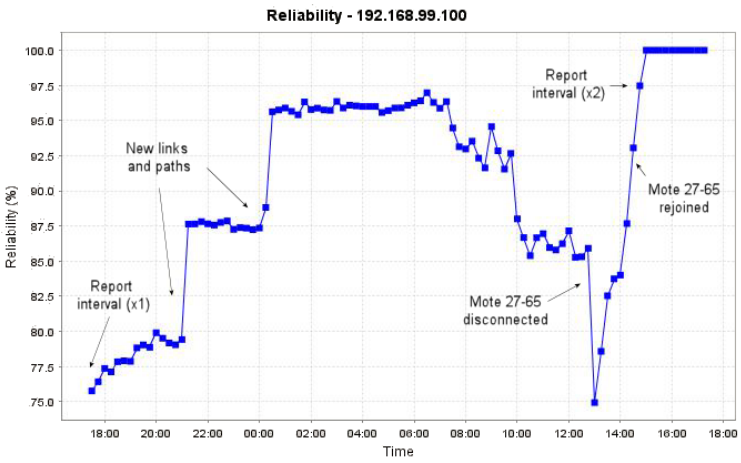


Figure 9.12: Average network reliability

The graph in figure 9.13 displays the packet loss for the different motes. Notice that the time scale is not continuous, but covers the most important stages during the experiment.

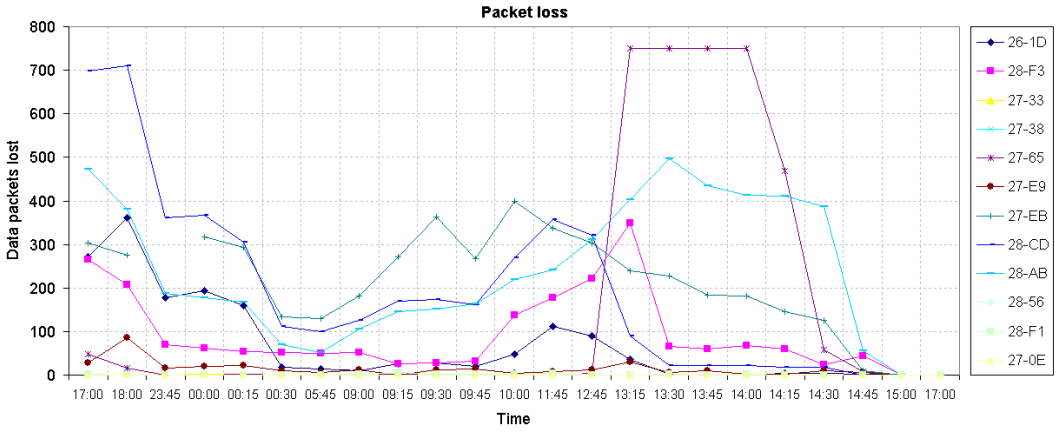


Figure 9.13: Packet loss for different motes

Latency Statistics

If a mote gets more data packets than it is capable of sending in one frame, it will have to deal with the unsent packets in the next frame. This will lead to an increase in latency for those packets. The latency will get even higher if the link quality is varying for some of the motes. Retransmissions will steal a lot of bandwidth which could have been spent sending packets. As mentioned earlier, a decrease in latency occurred when a mote was disconnected. The fact that the rejoining of the mote did not lead to a new latency increase could mean that better and more stabile paths were assigned. Normal average latency occurred when the report intervals were doubled. This seems to be the limit for the installed network.

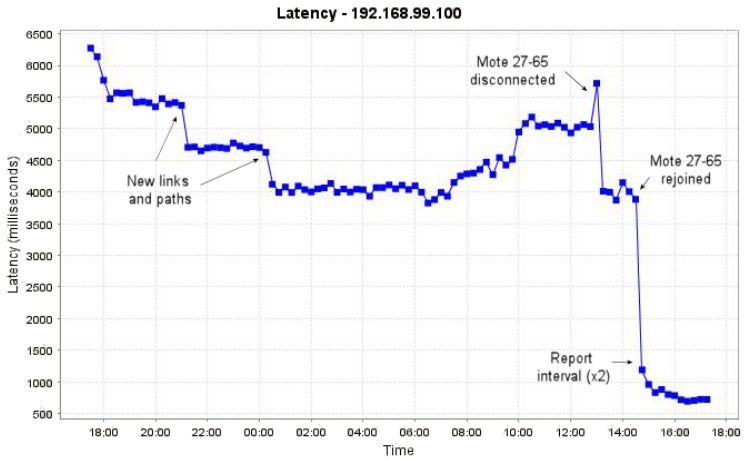


Figure 9.14: Average network latency

Latency for the different motes is depicted in figure 9.15. The two first values for 28-CD are 110 seconds and 164 seconds, these values are left out in the figure because they would ruin the scaling of the graph.

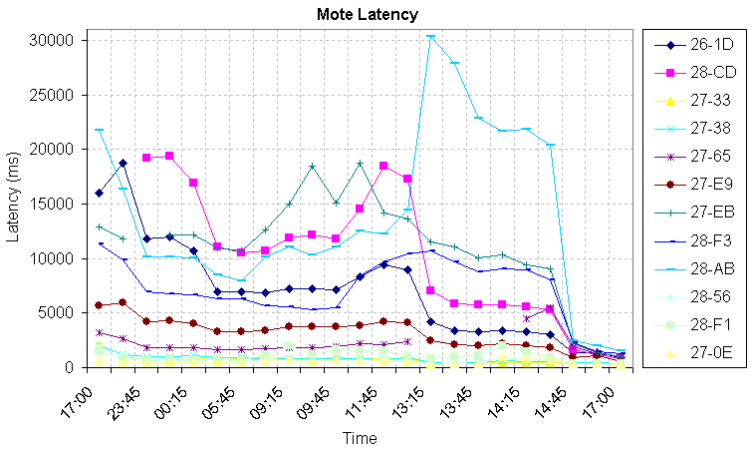


Figure 9.15: Mote Latency

Path Stability

The graph depicted in figure 9.16 shows average path stability varying between 88% and 98%. In contrast to the first experiment it is actually varying all the time, showing no smooth graph to analyze. The reason for this behaviour is network congestion. When a mote queue is full, holding 8 packets, it will stop generating samples and return NACKs to all transmitting motes. As explained in the ”transmission details” presented in section 3.2.2, NACKs are also counted as failed transmissions and will affect the average path statistics.

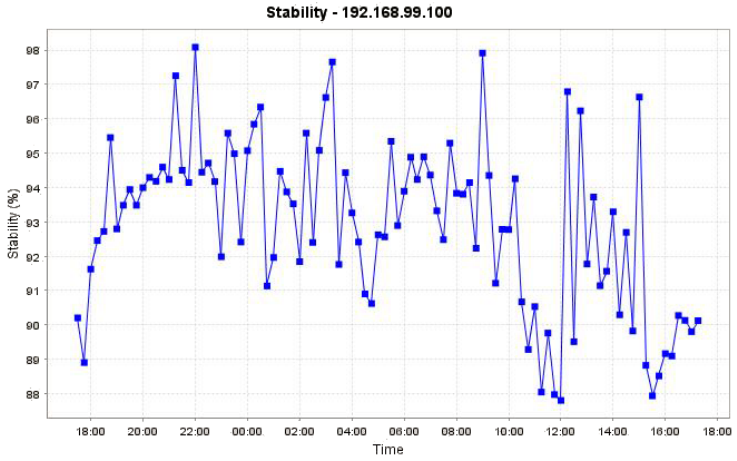


Figure 9.16: Average path stability indicating a strongly congested network

9.1.3 Experimental Set Three - Shorter Frame Length

This experiment was initiated with a frame length only twice the number of motes. When such frame is combined with a mesh topology, the number of links needed may easily exceed the number of available slots in the frame. To compensate for this possible lack of bandwidth/links it may be necessary to increase the report interval. In the current experiment however, a recommended report interval turned out to be sufficient.

Connectivity and Paths

The previous experiments have already covered network start-up, and joining and disconnection of motes. This experimental set will focus on the stable connected network, as depicted in figure 9.17.

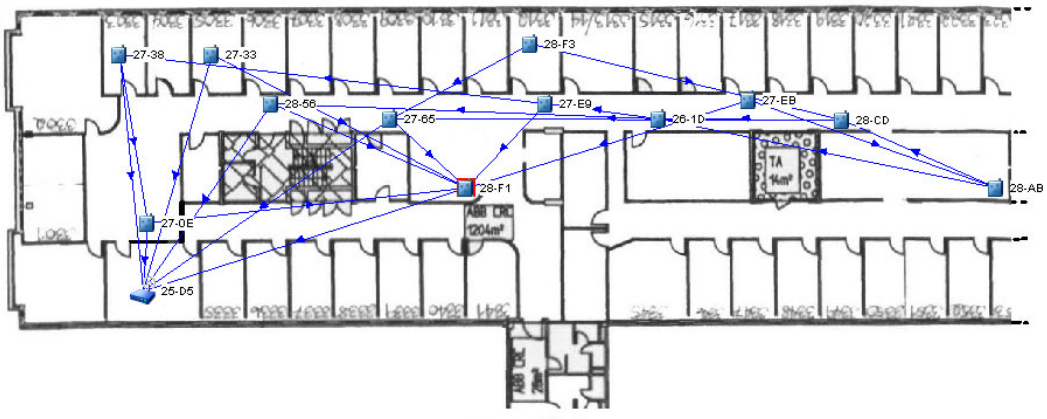


Figure 9.17: The connected network

Reliability Statistics

From a reliability perspective, the network seems unaffected by the shorter frame length as long as recommended report intervals were used. When the interval was reduced below the recommended threshold, the network showed some signs of instability. A few packets were dropped now and then, indicating a network operating on the edge of its capabilities. This can be viewed in the reliability graph given in figure 9.18. The report interval was changed to be two times the frame length at 9.00 am.

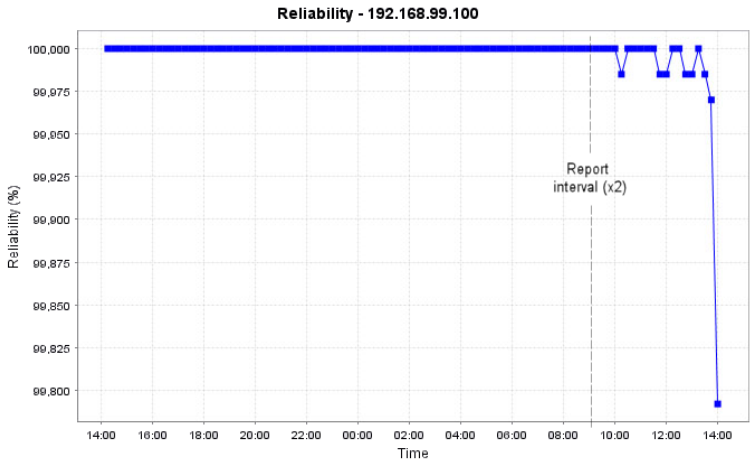


Figure 9.18: Average network reliability

The graph illustrating the packet loss of the motes, depicted in figure 9.19, shows only a few lost packets. All of them lost after the reduction in report interval. It should be mentioned that the same report interval gave a 100% reliable network in the previous experimental set. This indicates that the frame provides too few slots for the Manager to assign the correct number of links.

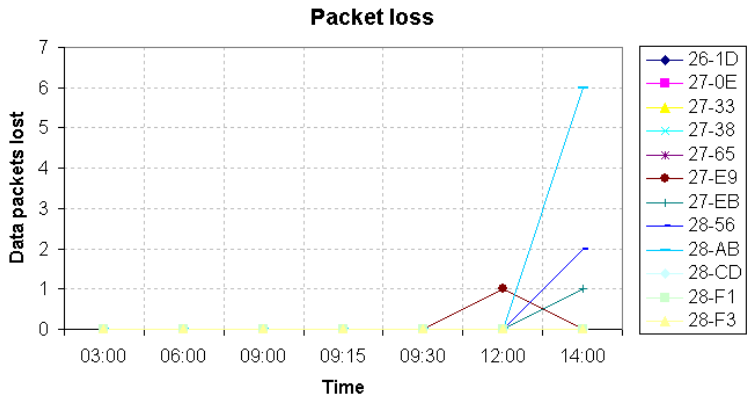


Figure 9.19: Packet loss for different motes

Latency Statistics

The average latency statistics did not agree with the expected results. Compared to the first experimental set, the average latency is actually higher with a shorter frame length. The reason may of course be that the frame length is lower than recommended, giving too few links to the network motes. This may also be the explanation for the increased latency received with the reduced report interval (at 9.00 am). Viewed in combination with path stability and compared to the congested network in experimental set two, it is obvious that the network becomes congested when the x2 report interval is initiated.

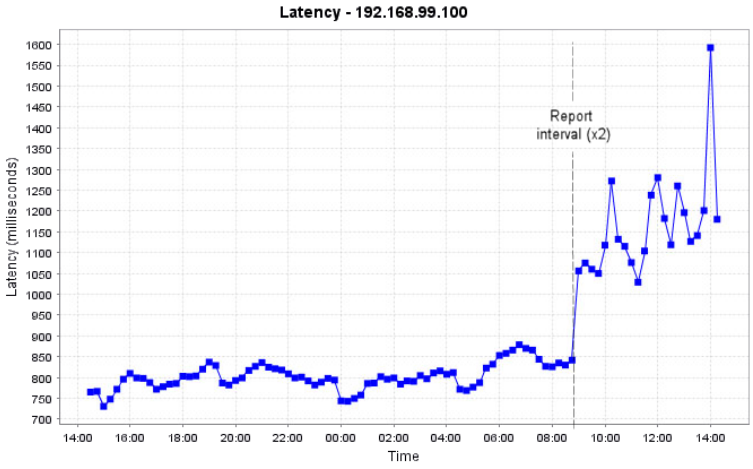


Figure 9.20: Average network latency

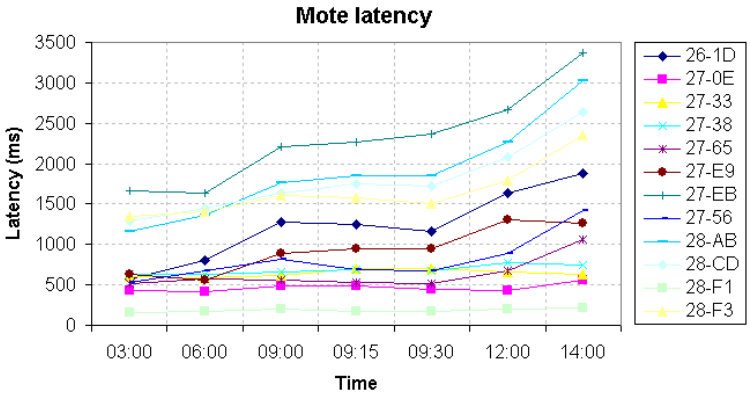


Figure 9.21: Mote latency

Path Stability

The path stability shows natural behaviour with a decrease in stability at daytime. But as with the second experimental set, the stability starts to vary when the network is operating on the edge of its capabilities. The reason is again network congestion.

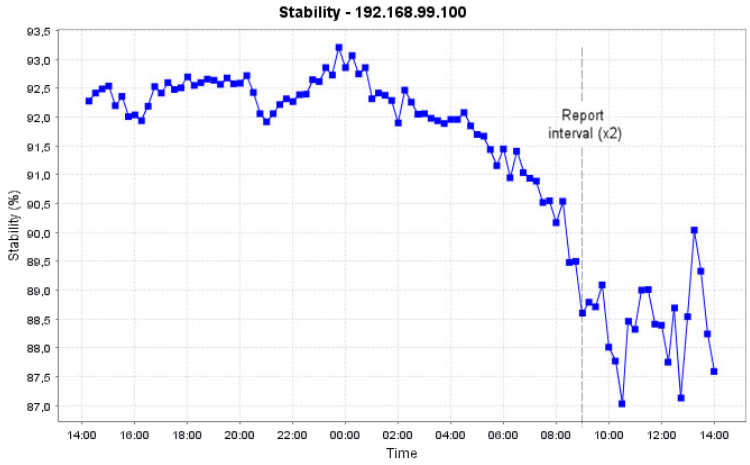


Figure 9.22: Average path stability

Network Lifetime

To collect statistics concerning the remaining network lifetime, the network has to be left running for more than 24 hours. The first and only time it was succeeded to collect and view this data was in this experimental set, where the network was left running for an entire weekend. Exactly why it takes this long to collect lifetime data is unknown. Perhaps it is necessary to get accurate calculations, or just a way to reduce network traffic.

The Battery life is calculated based on the most recent activity of the mote and is thus depending on profile settings, frame length and the number of children the mote has. When a mote joins the network an assumption is made that the battery was fresh, and from that moment on, all activity is logged and integrated into units of mA-hour. The remaining life is calculated by taking the remaining mA-hour divided by the most recent level of activity.

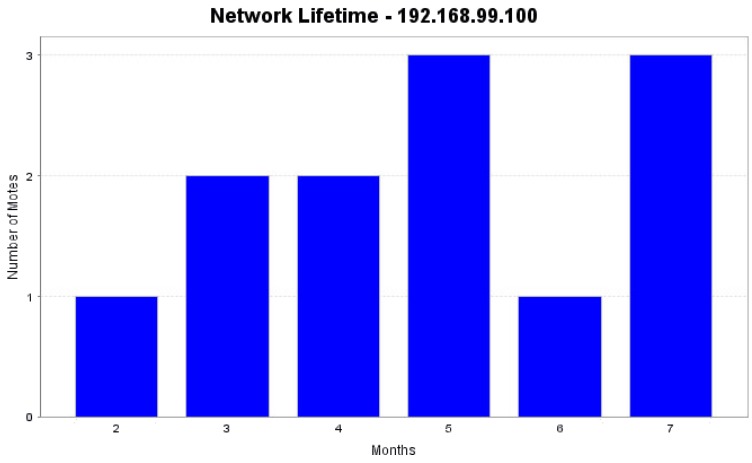


Figure 9.23: Remaining battery life

Knowing this, it is interesting to notice that with the current configuration, some motes will only last for a few months (illustrated in figure 9.23). The motes with the longest remaining battery life, probably leaf motes, may actually only last for 7 months. This observation, if the graphs can be trusted, indicates that when used in a network with rapid reporting and a short frame length, like in this experiment, at least some of the motes must be powered by wire.

9.1.4 Experimental Set Four - Longer Frame Length

Previous experimental sets have comprised both reduced and recommended frame lengths. This leaves only a frame length that is longer than the recommended minimum. The current experiment is initiated with a frame length four times the number of motes.

Connectivity and Paths

A reliable network was formed as illustrated in figure 9.24. As with the previous test there will be no new tests on disconnection and joining of motes in this experimental set. All events and alarms during the set can be found in the event log on the attached CD.

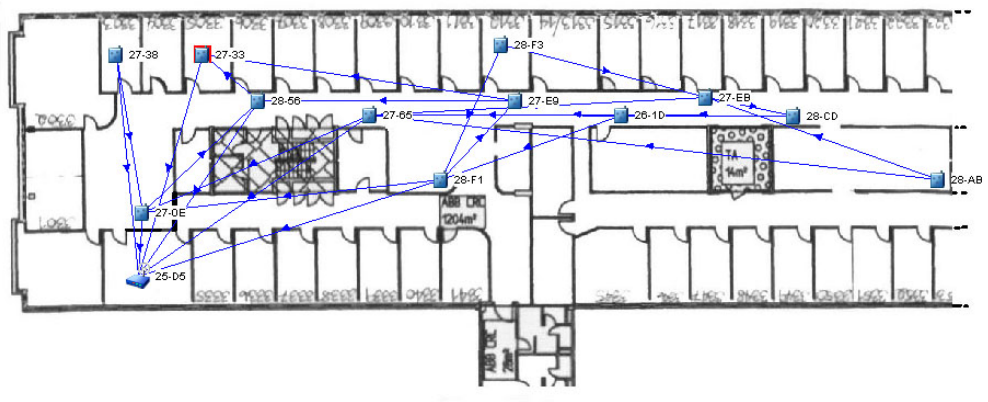


Figure 9.24: The connected network

Reliability Statistics

The network reached 100% reliability when all paths and links were created. With a frame length four times the network motes, there is no reason to believe that the network is short on bandwidth and thus, an x2 report interval can be initiated without risking reduced reliability. This was tested in experimental set 2, and since there have been no environmental changes the results will be equal.

Latency Statistics

The average latency, depicted in figure 9.25, never goes below 800 ms. This shows a slight increase in latency compared to the first experimental set. Hence, it is true that a longer frame length leads to increased latency.

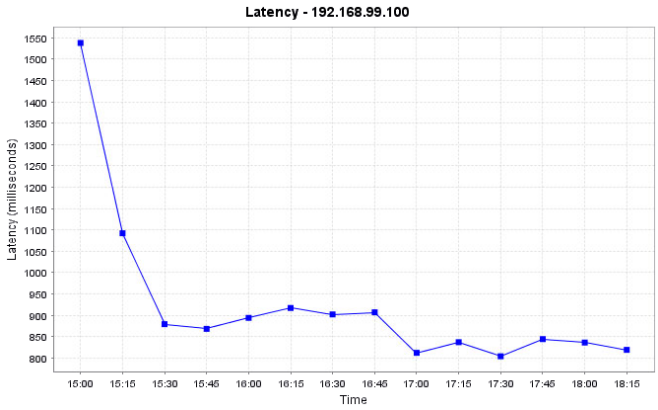


Figure 9.25: Average network latency

Path Stability

At the end of the work day, the path stability shows the same characteristic increase as the other experimental sets. It is easy to see the connection between path stability and latency when the two graphs are compared. Increased path stability leads to decreased latency.

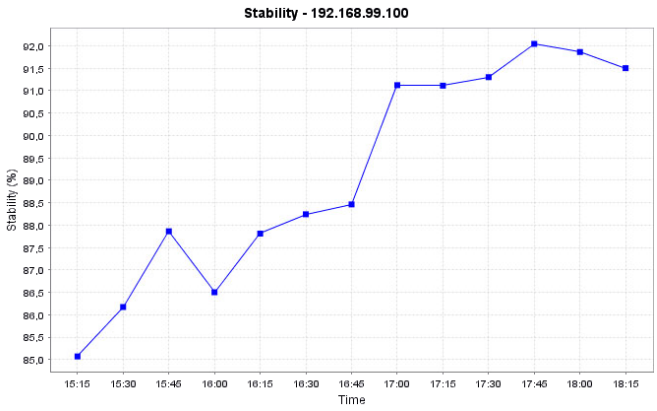


Figure 9.26: Average path stability

9.1.5 Experimental Set Five - Linear/Multi-Hop Topology

This experiment is performed under optimal conditions, using CLI commands to form the linear/multi-hop topology without motes meshing together. The results from this experiment must therefore be viewed as a best case scenario. With this in mind, the retrieved latency and reliability still gives a strong indication on what to expect from networks of various size. To point out the obvious, a linear topology is not recommended in most cases. Only one route exists from a mote to the Manager, and if a mote close to the Manager should go down, all of its children will also be lost.

Connectivity and Paths

When this experiment was initiated, it was assumed that the minimum frame length consisted of N+4 slots, independent of network topology. During the experiment, however, it was found that only a star topology can make use of this rule. In a linear/multi-hop topology, it is the mote closest to the Manager that is in need of most links. This mote will not only need upstream links to carry data from its children to the Manager, but also links to receive this data from its children. Because of this, we will need 2N+3 (23) slots to cover 10 multihop motes, and not N+4 (14) slots as assumed in the previous chapter. The discussion chapter explains this in detail.

During the experiment it was managed to connect all 10 motes, even with a frame consisting of 14 slots. This was a time consuming challenge, and when the last mote got connected after 4 hours, the network was congested to the point where statistics were no longer retrieved. It was tried to manually add more bandwidth/links with CLI commands, but because of the limited frame there were no links available.

When it was found that 10 motes resulted in too few links, four motes were disconnected to increase bandwidth and ensure faster start-up times. Still, this was not enough to provide all motes with the correct number of links. To get enough slots for six motes, it is necessary with 15 slots. Hence, the network frame was one slot short. The network still managed to maintain full reliability due to the x3 report interval, and it was decided to keep the six mote network throughout the experiment. Instead of mote removal, the problem could of course be solved by extending the frame length. But at the time it was assumed that the problem was somehow caused by the use of the ppath command.

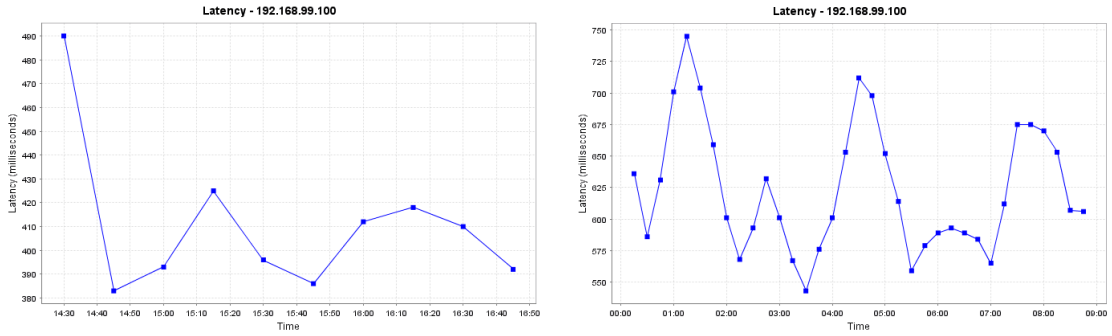
Figure 9.27: Connected linear/multi-hop topology

Reliability Statistics

Full reliability was achieved even with the shortest frame length. During the experiment all motes were set to send data at a rate three times the frame length. Thus, even with too few links, motes were able to forward data before new packets were generated. The fact that all motes were placed only centimetres apart made the signal conditions excellent, and additionally increased both path stability and reliability statistics.

Latency Statistics

The linear topology gave some interesting results. With the minimum frame length, consisting of 14 slots, the average latency varied between 380- and 430 ms. When the frame length was increased (with additional 14 slots), so was the latency. It now started to vary between 540- and 750 ms. Notice that not only are these numbers greater in value, but also farther apart.



(a) Minimum frame length

(b) Minimum x 2 frame length

Figure 9.28: Average latency with different frame lengths

The x4 frame length was initiated with a huge increase in average latency. By accident, one of the disconnected motes tried to rejoin the network a few minutes past 01.00 pm. This event led to the discovery of the multiple frame feature of the SmartMesh protocol. A second frame, frame 1, is created if frame 0 exceeds a certain length. In the software version used during the experiment, this limit is set to 50 slots. Frame 1 is a fast frame (31 slots) that is used to enable faster joining and hence faster network formation. It is initiated at start-up and by events like a new joining mote or an existing mote resetting and trying to re-join the network. Once active, the frame remains on for 1 hour. The average latency graph, depicted in figure 9.29, further illustrates this fact. When the faster frame is initiated by the joining mote, latency is decreased and do not increase again until one hour has past. The latency then vary between 1200- and 1250ms.

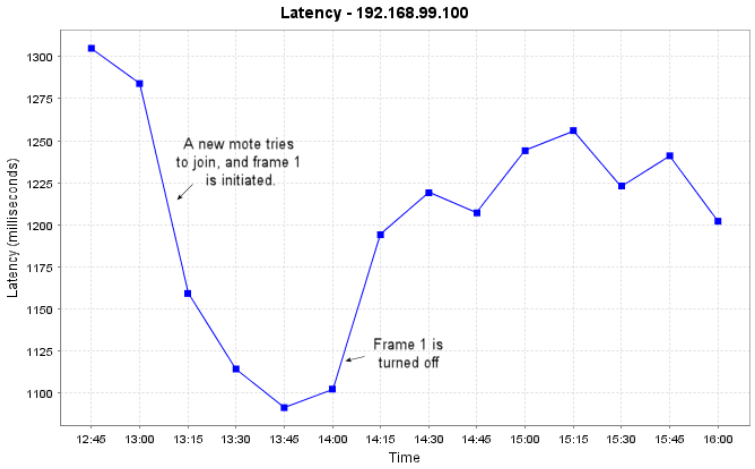
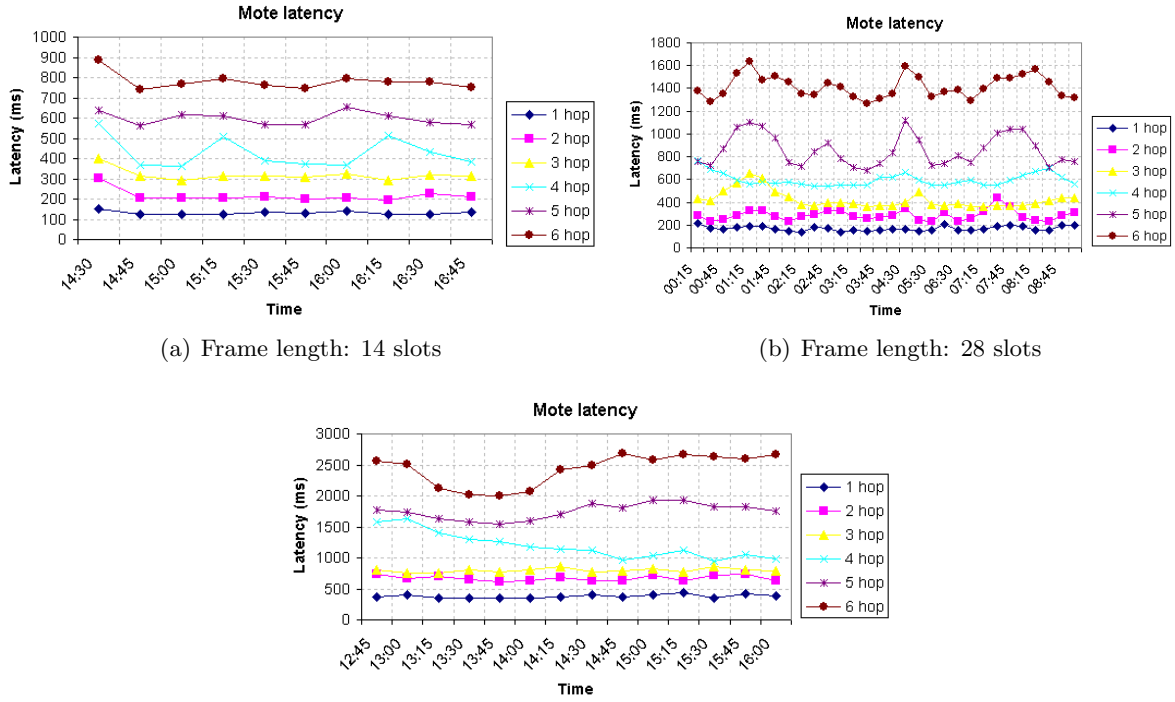


Figure 9.29: Latency with an x4 frame length

As mentioned earlier, a linear topology makes it possible to check the hop related latency of the different motes. The below figures describe this in terms of frame length and number of hops to the Manager. It is obvious that number of hops has great impact on the overall latency. See the discussion chapter for further comparison graphs.



(a) Frame length: 14 slots

(b) Frame length: 28 slots

(c) Frame length: 56 slots

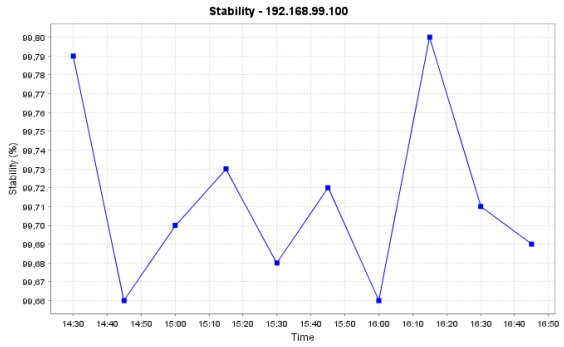
Figure 9.30: Mote latency for different frame lengths

Path Stability

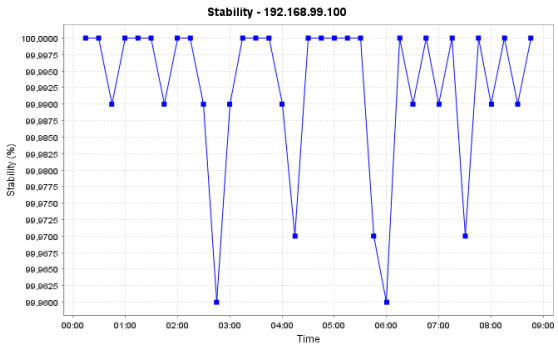
Because the six motes are placed only centimetres apart, full path stability was expected during this experiment. The reason why this is not the case in the first graph, figure 9.31(a), is due to lack of bandwidth. With a 14 slotted frame, the three motes closest to the Manager were missing one or more links. Still, path stability never goes below 99.66%. In the next picture, figure 9.31(b), the frame is extended to 28 slots. This is enough to provide the correct number of links to all motes. Hence, we get better path stability (less NACKs[3] caused by full buffers).

The last graph depicted in figure 9.31(c), shows the path stability when a frame consisting of 56 slots is used. Recall from the previous section that this will create an additional frame, frame 1, which is turned on automatically when a qualifying event happens. Frame 0 has priority over frame 1. This means that if two different motes are scheduled to talk to the same receiver in the same slot, but within two different frames. The receiver will choose to listen to the frame 0 transmitter, regardless of whether the transmitter sends anything or not. Hence, if the frame 1 transmitter sent something at that time, then that packet will get lost (not get an ACK) resulting in decreased path stability. Figure 9.31(c) illustrates this concept.

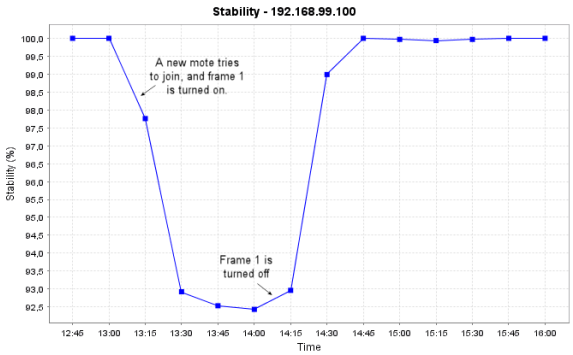
[3]Negative acknowledge (NACK) is sent by a mote if it received a broken or incomplete message, or if it for some reason is unable to receive the message at the time.



(a) Frame length: 14 slots



(b) Frame length: 28 slots



(c) Frame length: 56 slots

Figure 9.31: Path stability for different frame lengths

9.1.6 Experimental Set Six - Star Topology

Like the linear topology, this experiment is performed under optimal conditions. CLI commands are again used to force network motes to connect in the wanted topology. A star network should provide faster reporting and less latency than any of the other topologies. The drawback is that such network only covers a limited area, and since no meshing is allowed, motes are lost if their only path is blocked. But in contrast to the linear topology, a blocked mote will suffer alone.

Connectivity and Paths

In a star network the minimum frame length equals N+4, where N is the network size in number of motes. When the network was restarted with this configuration, eight motes connected the Manager within 2 minutes. After the joining of these motes the joining process seemed to come to a halt. The CLI command line showed that error 34 was received periodically. This error message means that there is not enough available bandwidth for the Manager to assign links to a joining mote. The same error message was received during the linear topology, but in this experiment the motes connected eventually. In the star topology, however, no new motes connected the network within the following hour. It should be mentioned that the star motes are not pure leaf motes, but still carry open listen links and downstream broadcast

links. There is no way to turn this off in the current software version.

The additional links could be one of the reasons to why the Manager is having trouble re-arranging links in a collision free way. Hence, the network may eventually form if it is just given enough time. However, a longer waiting period than one hour was considered too time consuming for the experiment. Instead, different frame lengths were tested until reasonable joining times were achieved. During these tests it was found that the frame length had to be extended with two slots to join one mote. To join the remaining four motes this meant a final frame length consisting of 24 slots. That's eight slots more than expected. But in return, we achieved faster network formation than in any of the experiments. All motes were connected within two minutes.

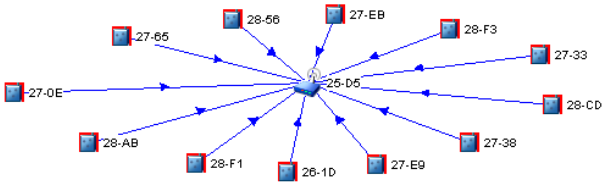


Figure 9.32: Connected star topology

Reliability Statistics

The star topology provided 100% reliable communication even with a report interval identical to the frame length. This is probably due to the optimal mote placement. In a real industrial environment 50% path stability should be expected, and thus an x1 report interval will lead to lost packets or a stop in packet generation.

Latency Statistics

Latency varies with frame length and number of hops to the Manager. In our 12 mote network, 24 slots turned out to be the minimum frame length. This resulted in an average latency varying between 390 ms and 425 ms. When the report interval was lowered to an x2 interval, latency decreased to about 350 ms. The latency was even further decreased when an x1 report interval was initiated. At this point the graph in figure 9.33 shows that the average latency is almost half the initial value. Dust Networks has been confronted with this behaviour and it turns out that it is caused by a bug in the current software version (1.5). The bug is likely to appear when the report intervals are "integer" multiples of the frame size, meaning that the x1, x2 and x3 intervals utilized throughout this entire project, may have been an unfortunate choice. In the other experiments, however, the differences in latency have not even been close to that of the star topology.

As a final test, the report interval was extended to 6 seconds. This increased the latency to 460 ms in mean value, which is an unsignificant change compared to the x3 report interval. The graph illustrating this step can be found on the attached CD, while graphs for average and mote specific latency (with the other report intervals) are illustrated in figure 9.33 and

figure 9.34 respectively. The two graphs look almost identical because all motes in the network seem to follow the same curve when it comes to latency.

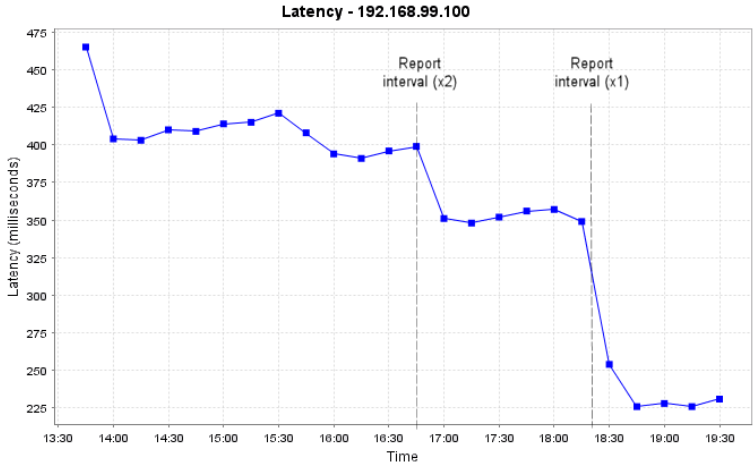


Figure 9.33: Average latency

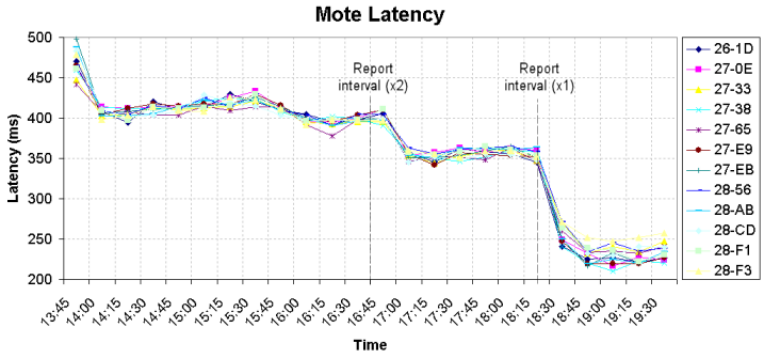


Figure 9.34: Mote latency

Path Stability

The graph depicted in figure 9.35 shows path stability close to 100%. For the x3 and x2 report intervals, full reliability is maintained even with a few failed packets[4]. For the x1 report interval the situation is a little different. If a mote fails to deliver a packet, that packet has to be sent in the next frame. But by the time that packet is to be sent, a new packet is generated. The newly generated packet now has to be sent in the next frame. This will continue until eventually the buffer is filled with packets. The mote will then stop generating packets until there again is available space in the buffer. With this in mind, it is possible to draw the conclusion that the network is not 100.000% reliable at all times. The graph shows

[4]Recall that failed packets are defined as sent packets without ACKs in return.

a decrease at 7.00 pm, meaning that a few packets is ”lost”, or not generated at all. However, this didn't show in the reliability graph because it was just a matter of 10 packets out of 100,000.

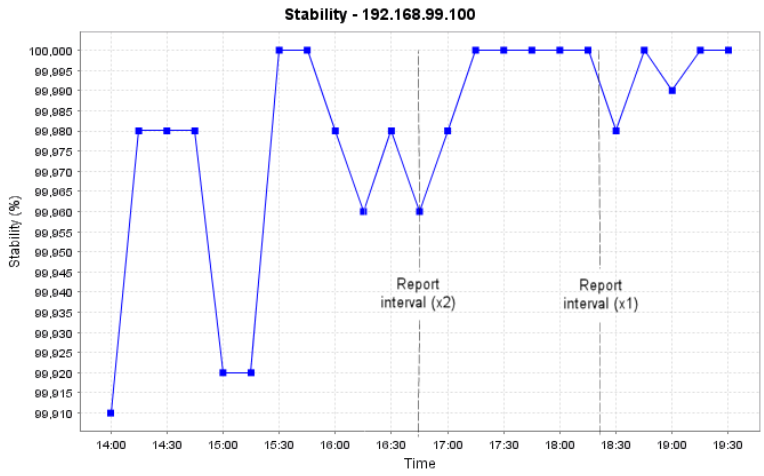


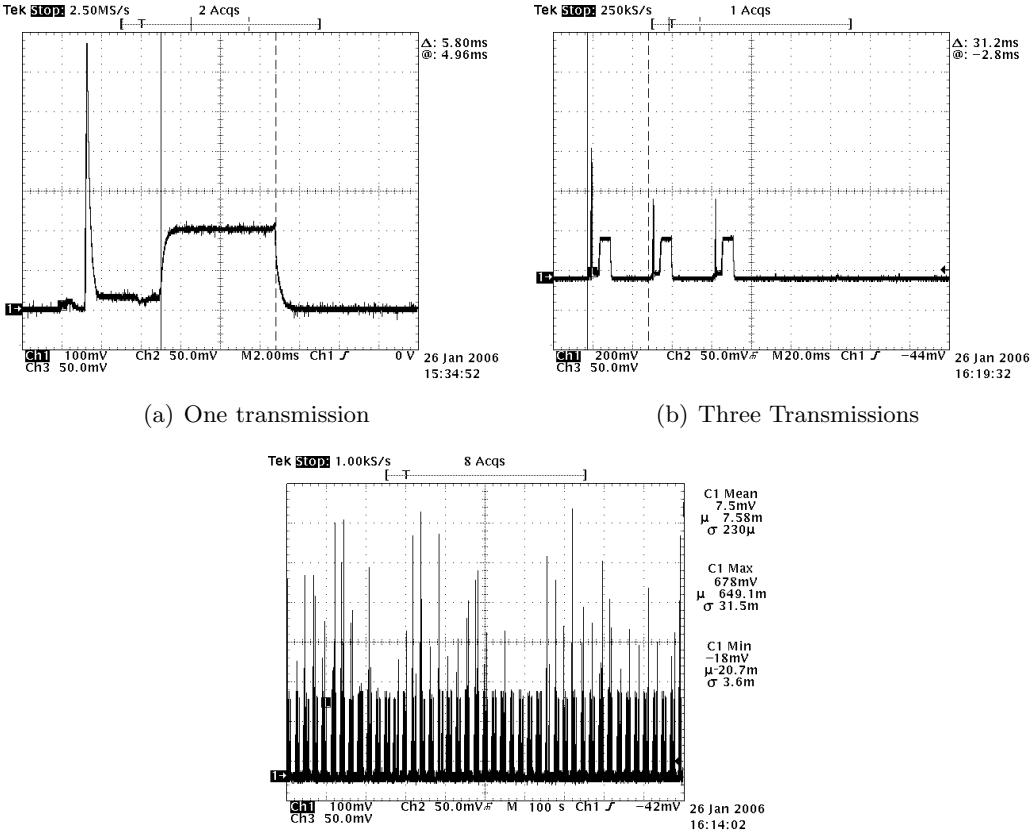
Figure 9.35: Average path stability

9.1.7 Experimental Set Seven - Power Consumption

The power consumption measurements were performed along with experimental set five. A linear topology made it possible to find the average power consumption for motes with various numbers of children and grandchildren. The oscilloscope was connected to the first mote in the topology, and measurements were taken every time a new mote was connected to the network. Doing it this way ensured that the motes had the correct number of links to their parents. It was tested to do it the other way around, letting the whole network form and take measurements as motes were disconnected. In some cases these measurements deviated from what was expected. The reason was found with help of the CLI command, *show mote*. With this command it was discovered that links from child to parent was not always deleted even though grandchildren were disconnected. As a result parent motes had to listen on more links than necessary, burning more power than they should. Hence, power measurements had to be performed during network formation.

The typical power consumption during a single transmission starts with a high peak followed by the actual transmission, as depicted in figure 9.36(a). A typical transmission is about 6ms in length, depending on the packet size. Figure 9.36(b) shows three timeslots in series, each with its own transmission.

Mean power consumption was found by averaging at high intervals. The oscilloscope calculates mean values of the intervals currently on its screen, which is why a high interval is needed. The first tests where carried out with a 50 second interval, this was broadened to a 3 minute interval in the succeeding tests. At least 6 minutes where spent on each measurement. This step can be viewed in figure 9.36(c).



(a) One transmission

(b) Three Transmissions

(c) Average power consumption

Figure 9.36: Power consumption measurements

The three tables below, table 1, table 2 and table 3, summarize the different measurements based on mote location and frame length. Mote 1 is the mote closest to the Manager, and is thus also the mote with the highest number of links and highest power consumption. Mote 6 on the other hand is a leaf mote with no children. Data from this mote must travel through all the other motes until it ends up at the Manager.

According to the three tables there is no noticeable difference between a 28 slotted frame and a 56 slotted frame. This is of course wrong. Recall from experimental set five that frame 1 is initiated when the frame length exceeds 50 slots. During network formation both frames are running simultaneously, increasing the power consumption for all motes. When the last measurement was done, all motes were connected and had been running for one hour. This was long enough for frame 1 to turn itself off. Hence, mote 1 has a much lower mean value at the time of the measurement than it would have only a few minutes earlier.

	Mean voltage (μ)	**Standard deviation (σ)**	**Mean current consumption**
Mote 1	27.8 mV	30 μV	2.8 mA
Mote 2	26.3 mV	240 μV	2.6 mA
Mote 3	24.6 mV	180 μV	2.5 mA
Mote 4	20.0 mV	0 μV	2.0 mA
Mote 5	15.5 mV	50 μV	1.6 mA
Mote 6	10.7 mV	70 μV	1.1 mA

Table 9.3: Power consumption with a frame consisting of 14 slots

	Mean voltage (μ)	**Standard deviation (σ)**	**Mean current consumption**
Mote 1	17.8 mV	110 μV	1.8 mA
Mote 2	15.6 mV	50 μV	1.6 mA
Mote 3	13.1 mV	0 μV	1.3 mA
Mote 4	10.3 mV	100 μV	1.0 mA
Mote 5	7.9 mV	550 μV	0.8 mA
Mote 6	5.1 mV	540 μV	0.5 mA

Table 9.4: Power consumption with a frame consisting of 28 slots

	Mean voltage (μ)	**Standard deviation (σ)**	**Mean current consumption**
Mote 1	9.6 mV	240 μV	1.0 mA
Mote 2	12.5 mV	60 μV	1.3 mA*
Mote 3	11.5 mV	110 μV	1.2 mA*
Mote 4	10.1 mV	210 μV	1.0 mA*
Mote 5	9.3 mV	300 μV	0.9 mA*
Mote 6	6.8 mV	140 μV	0.7 mA*

Table 9.5: Power consumption with frame 0 consisting of 56 slots (*two frames).

9.2 Results from the Industrial Experiment at Statoil

Before the results are examined there are some minor issues that should be mentioned. Recall from the previous chapter, that a misunderstanding led to the use of an N+4 frame length instead of the actual minimum. However, since there are more routers than sensing motes in the network, this will only affect the results in a positive way. The discussion chapter addresses this topic in section 10.3. Another factor that may affect the results compared to that of the IZT, is the lack of formation control. The IZT documentation [33, 34] does not describe any paths other than the ones planned in section 5.3.2, and it must therefore be assumed that the network was restricted to these paths. Without CLI commands it was impossible to make restrictions like these in the SmartMesh network and the paths may be a little different from the IZT even though the physical installation of the motes are identical.

9.2.1 Connectivity and Paths

The previous chapter explained the installation procedure in detail. Before collection of statistics and data was initiated, all statistics was cleared and correct profiles where applied to the motes. With recommended frame length and report intervals the network formed as illustrated in figure 9.37. Notice that the two motes close to the Manager have additional paths to the mote farther away. These paths had not been possible with a minimum frame length because too few slots would be available to carry the links. A more detailed discussion is carried out in the next chapter.

Figure 9.37: Network paths (recommended frame length)

Recommended frame length made it possible to form a network with only four hops (best case) from the wheel to the Manager. With the minimum frame length the network formed as depicted in figure 9.38. The best case route is still four hops, but the limited number of slots has forced the Manager to close the path between 28-F3 and 26-1D, and redirect the paths of the closest motes. During the night the path between 27-E9 and the manager were lost and the best case number of hops changed to five hops. That means a configuration almost identical to the IZT. As long as the motes were installed at identical locations as the IZT, the network was allowed to form without interference. Since no path-commands were available at the time, it was considered an ”impossible” task to force formation.

Figure 9.38: Network paths (lowest possible frame length)

9.2.2 Reliability Statistics

The experiment performed with 100% reliability during all test configurations. This further proves that router motes are the key to getting the fastest and most reliable network possible.

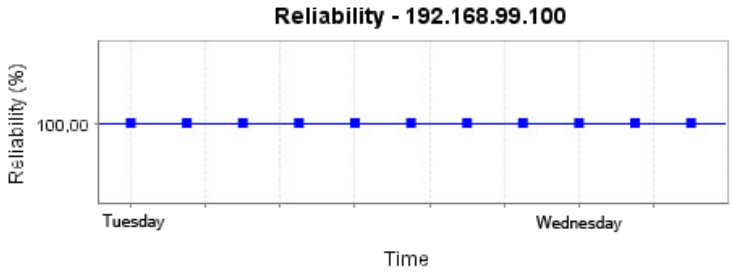


Figure 9.39: Average network reliability

9.2.3 Latency Statistics

The latency statistics is divided into sections for recommended and minimum frame length. This makes better scaling on the graphs and makes it easier to compare the different results.

Recommended Frame Length

With the door still open and the wheel in a locked position (stopped), the average latency is varying around 450 ms. When the door is closed the statistics is seemingly unaffected. This indicates that the path through the concrete wall is the one most frequently used. It is first when the wheel is started, setting the mote in motion, that a significant change can be viewed in the statistics. The latency increases up to the point where an x2 report interval is initiated. Right there, the latency drops with 70 ms compared to the previous 15 minute statistics. This behaviour is probably caused by a bug in the current software version (1.5), as described in section 9.1.6.

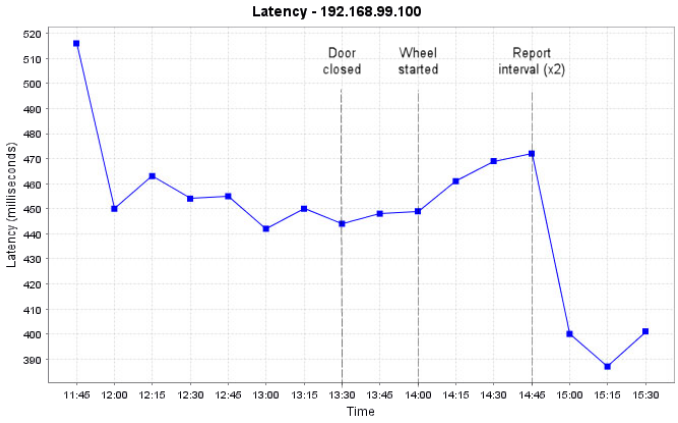


Figure 9.40: Average latency (recommended frame length)

The mote specific latency illustrated in figure 9.41, shows that the two closest motes have different latency even though they both share the same path configuration. In fact, one of the motes has twice as high latency as the other. The only reasonable explanation is that the Manager has assigned links in such way that data from one of the motes (27-38) is routed the longer route through the router mote (27-E9) further away. As explained in section 3.2.2, a mote sends its data in the first available slot. This means that the timing of data sampling is responsible for the chosen transmission link.

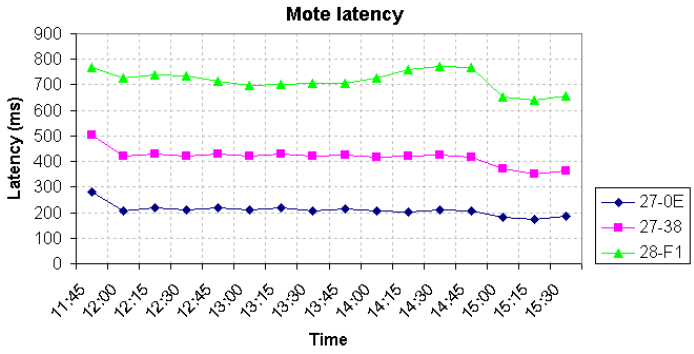


Figure 9.41: Mote latency (recommended frame length)

Minimum Frame Length

As expected the minimum frame length resulted in a much lower latency. When the x2 report interval was introduced, the latency decreased to an even lower value. During the next thirteen hours the average latency never exceeded 335 ms. This is the minimum latency possible for the network. The x1 report interval was successfully initiated, but added some extra milliseconds to the average latency. Still, the x1 report interval makes it possible with more rapid reporting even when latency is added.

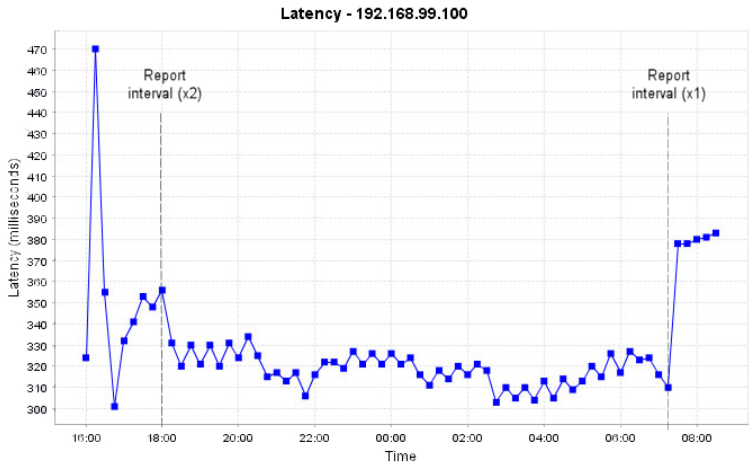


Figure 9.42: Average latency (minimum frame length)

The latency for the two closest motes is now almost identical. It is possible to notice that the mote (27-0E) with an additional path to its neighbours (27-38) has latency slightly higher than its neighbour. The obvious reason is that data is sometimes routed through this mote.

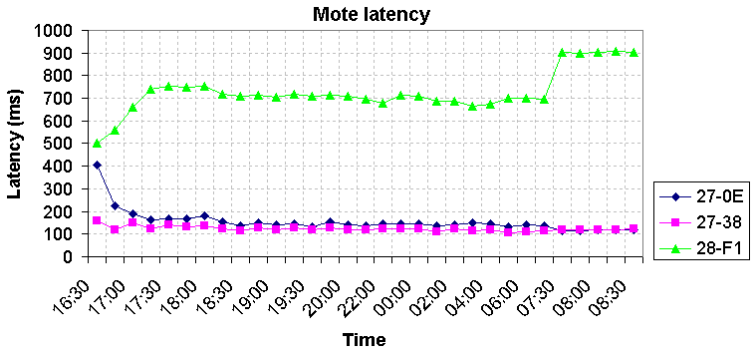


Figure 9.43: Mote latency (minimum frame length)

Latency per hop

It is quite obvious that latency varies depending on number of hops from the sending mote to the Manager. Latency per hop is summarized in table 9.6. The cells only contain approximate

values, retrieved over a 15 minute period. To get more accurate results this period should be extended to a few hours (at least). The limited period was used due to limited lab access time. Statistics were collected using minimum frame length in combination with recommended report interval (x3). All motes were given reporting profiles during this test.

1 Hop	2 Hop	3 Hop	4 Hop	5 Hop	6 Hop
155 ms	205 ms	355 ms	440 ms	680 ms	1107 ms

Table 9.6: Latency per hop

9.2.4 Path Stability

All configurations tested during the industrial experiment resulted in much better path stability than achieved in any of the ABB experiments. Improved path stability in an industrial environment compared to an office environment is not expected in the overall situation. There are several factors in play, making it possible for the industrial experiment to gain the highest average path stability. For instance, there was more human traffic at the ABB facility. In addition the ABB environment was covered with several WLANs, decreasing network performance. Perhaps the most important difference is the lack of router motes in the ABB setup. This results in less available links/bandwidth and increased chances for network congestion. Remember that NACKs count as faulty transmissions.

Recommended Frame Length

The graph in figure 9.44 shows that only two out of one hundred transmissions are faulty in the best case. These numbers were achieved even with the explosion safe door closed. When the wheel is started, three additional transmission failures occur per hundred in the worst case. These faulty transmissions are easily handled by the redundant routes for the wheel mote, assuming that this is the location with the most unstable environment.

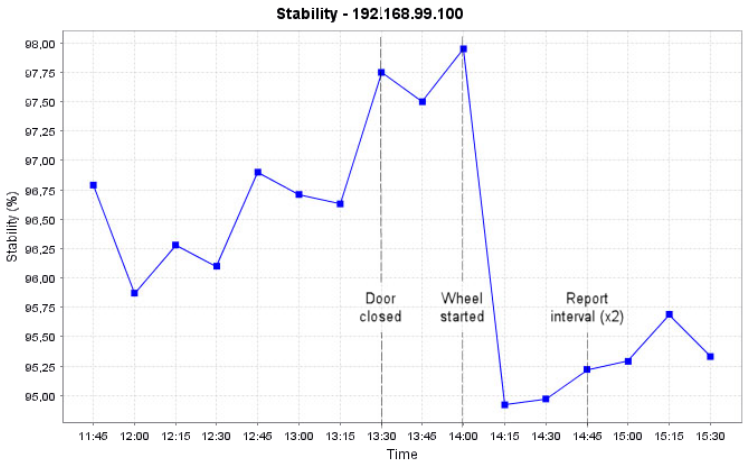


Figure 9.44: Average path stability (recommended frame length)

Minimum Frame Length

The path stability was slightly increased during the configuration with minimum frame length. The most reasonable explanation is that the RF conditions are improved during the night, probably caused by less human traffic in the area. Even with an x1 report interval the stability is never below 98%.

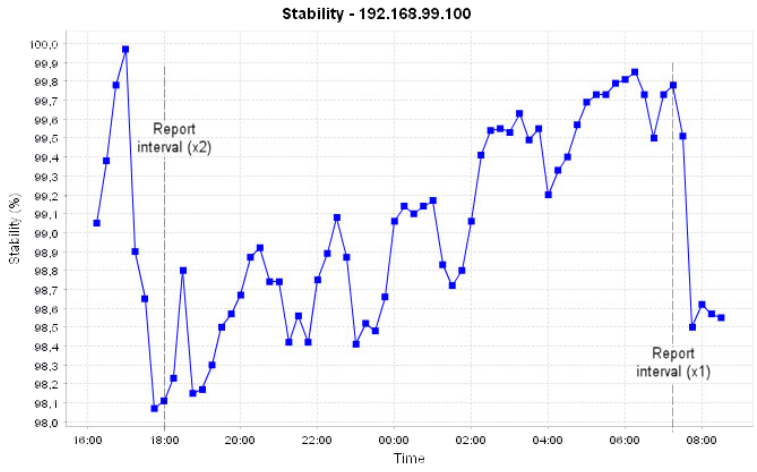


Figure 9.45: Average path stability (minimum frame length)

9.2.5 Starvation

While the network was running with an x1 report interval, all motes were given the reporting profile. In addition to a huge increase in latency, the event also led to the loss of all basement motes. As explained in chapter 3, a mote is reset if the retries/timeout limit is breached. This is what happened when the basement motes were lost. After a reset the motes were able to re-join the network and continue their tasks. But if no extra bandwidth/links are added, there is no guarantee that the behaviour won't repeat itself.

9.2.6 Mote Range

The above results indicate that the SmartMesh motes have approximately the same range as the IZT nodes. When good mote positions were eventually found, no problems occurred during the experiment.

9.2.7 Throughput

Maximum throughput was achieved with minimum frame length and an x1 report interval. This resulted in a 344 ms report interval with average latency varying around 380 ms.

Chapter 10

Discussion

The results from the various experiments illustrated and revealed many interesting features and properties for the SmartMesh protocol and the technology in general. One of the most surprising discoveries was the fact that the office environment at ABB turned out to be a lot more challenging for the network than the industrial environment at Statoil. Hence, it is proven that RF barriers like thick concrete walls, engines and rotating wheels can be a "small" challenge compared to WiFi jamming and multipath fading.

The goal of this chapter is to provide an overview of the properties and capabilities of the SmartMesh network. When the reader has finished this chapter, he or she should be able to fully understand all behaviour observed in the previous chapter. It is also the intention of this chapter to provide readers with the means to determine whether or not the protocol can meet the requirements of different applications.

10.1 Sources of Errors

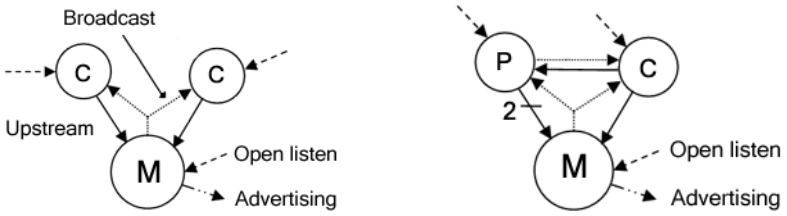
Before further discussion is carried out, it is appropriate to point out a few sources of errors that may have influenced the collected statistics. The reader should be aware that all statistics deviates from time to time even with identical configurations in identical environments. Still, there are a few factors that can make the results even more fluctuating than they should.

The first and perhaps most important source of error is caused by the changing environment. Each test in the different experiments was performed over relatively short periods and in the order of hours. Ideally, each test should have continued for days or perhaps even weeks. For instance, there is a huge difference between performing a test at night and redoing the same test at daytime. Only in a static environment without RF interferers will the statistics remain unchanged.

The second source of error is caused by a bug in the protocol. When the results are examined it is possible to notice a decrease in latency every time the report interval is decreased. This behaviour is caused by a bug that is likely to occur when integer multiples of the frame length is used as report intervals. The faulty behaviour can be avoided by using non-integer multiples of the frame, but unfortunately this was not discovered until recently. However, not all experiments show this faulty behaviour and in most cases the deviations are insignificant.

10.2 Network Formation

There are two major factors that affect network formation; the frame length and the RF conditions at the time of network installation. The former is only a factor if the network is configured below the recommended settings to achieve higher report rates. As mentioned in previous chapters, the shortest possible frame length is N + 4 slots, where N is the number of network motes. A frame of this length reduces the meshing abilities of the network and must be handled with care. It is recommended to perform a complete analysis the network before a choice is made regarding a frame length below N x 3. The below figures illustrates the difference between a direct connected network and a network making use of the mesh topology. Notice that the meshed network in figure 10.1(b) exceeds the N + 4 frame length.



(a) 6 slots (N+4)

(b) 7 slots

Figure 10.1: Minimum frame lengths

The RF environment plays an important role in network formation. Criteria for path creation were explained in Chapter 3. To refresh, suitable parents are chosen based on hop depth as long as RSSI values are better than -87dBm. When a path is created it is not deleted throughout network lifetime unless the retries/timeout threshold is breached for that path. As a result, the network may in some cases get poor path stability if it was installed at a time when the RF conditions were exceptionally good. The experiments showed that even if all motes had two parents the first time the network was formed, this was not necessarily the case after a restart. To get the best network possible, it should be installed at the time when the worst case conditions are expected.

The time it takes to find all network motes may vary. Network topology plays an important role in this process. It is obvious that it takes more time to find motes farther away, than motes close to the Manager. The longest start-up time during all experiments was found in the congested network in experimental set two. With this configuration as much as 20 minutes past until all motes had paths assigned. This is partly caused by unstable connections and limited bandwidth, causing motes to reset before the Manager has time to assign more paths and links. In comparison, the industrial experiment spent 18 minutes searching for motes until the discovery process completed. This network has only seven motes, but the number of hops in the linear topology increase the start-up time considerably. The best start-up times for a mesh-connected network were observed in the non-congested experiments at ABB, varying around 11-12 minutes. With a pure star topology, however, it is possible to achieve start-up times about 2 minutes (12 motes).

In larger networks, longer frame lengths must be used. This will result in increased start-up

times in addition to higher latency. To speed up the formation process in such networks the SmartMesh protocol includes a multiple frame feature, made possible by frequency hopping and 16 available channels per slot. When a network is started with a frame that exceeds a certain length in slots, an additional shorter and thus faster frame is initiated simultaneously with the main frame. This extra frame ensures faster formation, and will remain on for one hour or until the ”maximum number of motes” setting is reached. When stopped, it may again be initiated by certain events, like a joining or resetting mote etc. Dust Network has planned to make this feature available for other tasks as well, making it possible to use the network in a range of new applications. Rapid reporting is then not only restricted to small networks, but may also be achieved in larger networks.

In networks with nodes several hops away, involving start-up times greater than 18 minutes, a restart or shutdown of a Manager may have huge consequences depending on the applications it is running. To prevent expensive downtime in case of Manager failures, SmartMesh-XR supports redundant Managers.

Dust Networks is also planning to halve the current time slots. This will improve both formation times and latency and make faster reporting possible.

10.3 High Report Rates versus Reliability

To achieve the highest possible report rates, it is necessary to reduce the frame length to a minimum. The minimum frame length is in turn dependent on the network size and topology, which may differ depending on the RF environment. In a typical RF-environment, 50% path stability should be expected. There are two approaches that may be used to ensure full reliability in the presence of failed transmissions; either to add more links in the frame, or to increase the report intervals (assuming initial x1 report intervals). This of course includes retransmissions, which should be implemented in all communication protocols where reliability is of the essence. However, if it was possible to achieve 100% path stability, an x1 report interval may be used without adding extra bandwidth/links. Available bandwidth/links are in turn depending on the length of the frame and we are back where we started. In networks with many router motes, however, bandwidth/links are added implicitly. The Manager adds links to motes independent of their profile, and thus a router mote will have additional links to forward data. A real world example will now follow.

During the industrial experiment an x1 report interval was successfully initiated even without 100% path stability. The reason is obvious; a quick look at the connectivity map shows that with a minimum frame length, one of the router motes connects directly to the wheel mote. As explained in Chapter 3, the Manager adds links depending on the number of children a mote has. In this particular case, the child was originally meant as a router and not the other way around. Hence, the wheel mote was given an additional link to send its data and as the test results shows, this was enough to maintain reliability even with the high pace. The below figures, figure 10.2 and figure 10.3, illustrate the different link assignments for the recommended and minimum frame length. Only upstream links are indicated to give a clear overview. In addition, four slots must be added to cover, neighbour discovery, open listen, advertising and broadcasting links (seen from the Manager’s perspective).

With a recommended frame length, the network forms as illustrated in figure 10.2. Links are

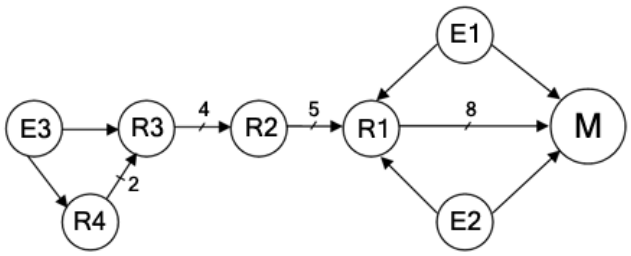


Figure 10.2: Recommended frame length (7 x 3 = 21 slots)

assigned as the numbers indicate. Since the wheel mote, E3, is the only reporting mote more than two hops away from the Manager, it has a lot of bandwidth/links on its upstream route. When data reaches R2, as much as five slots are available to forward its data. This should make it pretty obvious why it is recommended to insert router motes throughout a network.

The recommended frame length results in 21 available slots. However, only 19 slots are necessary to form the network shown in figure 10.2. The linear/multi-hop sub-network connected to R1, makes this the mote in need of most slots. Seven slots are needed for it to receive data from its children, while eight slots must be used to forward the data to the manager (upstream links). It will also need one slot to hold the downstream/broadcast link from the Manager, and one slot to forward broadcasts to its children. In addition, one slot is needed to listen for joining motes (open listen). The neighbour discovery slot further extends the minimum frame length, but since it is shared by all motes it will not make a huge difference. To achieve the highest possible report rates, while maintaining the most reliable topology, such analysis can be done before a final frame length decision is made.

At the time of the industrial experiment, a misunderstanding led to the assumption that the minimum frame length was N+4, as long as the network was kept small in size. Extensive analysis has proven this assumption wrong. With a frame length consisting of 11 slots (N+4), it was possible to achieve the topology depicted in figure 10.3. However, for the number of links to be correct, the frame length must be slightly extended. Additional 4 slots ¨must be provided to handle all links needed by mote R1. In the industrial setup, 11 slots were sufficient because only three motes were sending data. Notice that R4 is now connected to the wheel mote, giving the wheel mote one additional link for data transmissions. The upstream links assigned for R1 is randomly placed. The important thing here is to give motes enough bandwidth to forward the data from its children, grandchildren and so on.
The starvation experiment further illustrated the benefit of router motes. When all motes in the industrial experiment were given reporting profiles, a recommended report interval (x3) was necessary to insure a reliable network. It is possible to add extra links via the command level interface (CLI), but this would result in a lot of work for the network integrator.

10.4 Latency

There are several conditions affecting the latency in a network. The RF environment may vary, causing increased or decreased latency depending on the number of retransmissions

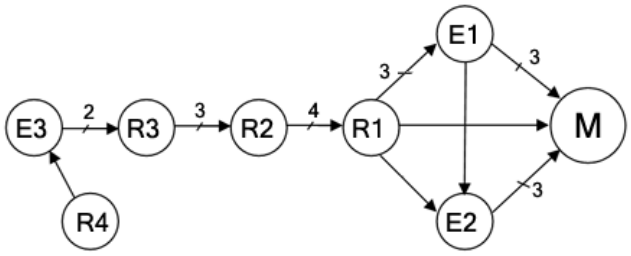


Figure 10.3: Minimum frame length (15 slots)

necessary. Intelligent mote placement and increased number of router motes are methods to reduce the environmental influence. These methods may have undesirable effects to the overall latency, because latency is influenced by the network topology as well. For instance, if the number of hops from a mote to the Manager is increased, the latency is also increased.

The final and perhaps one of the most important factors influencing network latency is the frame length. To illustrate this, a scenario with three motes (A, B and C) is introduced. Their communication links in the frame are shown in figure 10.4, along with the latency added for each hop. As the picture illustrates, the longest latency periods occur if a transmission fails. The worst case scenario occurs if a mote has only one available upstream link, as for mote B. This will not be the case in a real life network because the Manager will add an extra data carrying link for mote B to forward the extra data from mote C. In fact, only leaf motes will apply to the present scenario, but the principle remains the same. If the frame length is extended, so will the intervals, L1 and L2. Obviously, latency will vary depending on the number of links and their placement in the frame. This makes it hard to make more accurate calculations about expected latency for different frame lengths.

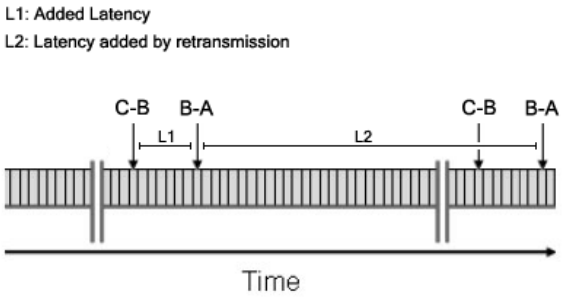


Figure 10.4: Latency due to link placement in the frame

Even though exact latency is hard to predict, it is important to have some knowledge of approximately what latency to expect when different frame lengths are used. For some tasks, like hard or immediate real-time applications, time is of the essence. In the worst case scenario, such applications may suffer a critical failure if their time constraints are violated. Thus, it is important that the chosen frame length is short enough to make latency oblige to the

application requirements.

Latency data has been collected in all experiments, involving different environments, various topologies and different frame lengths. As expected and as showed by the results, latency strongly varies depending on these factors. In the experiment with linear/multihop topology, presented in section 9.1.5, three different frame lengths were tested in a six mote networks. The experiment was performed under optimal conditions and with all motes placed just a few meters away from the Manager. In a real world experiment additional latency is expected, most likely to be caused by failed transmissions. Note that if a mote close to the Manager has low path stability, all motes connected to this mote will experience decreased latency. The comparison of three different frame lengths is illustrated in figure 10.5.

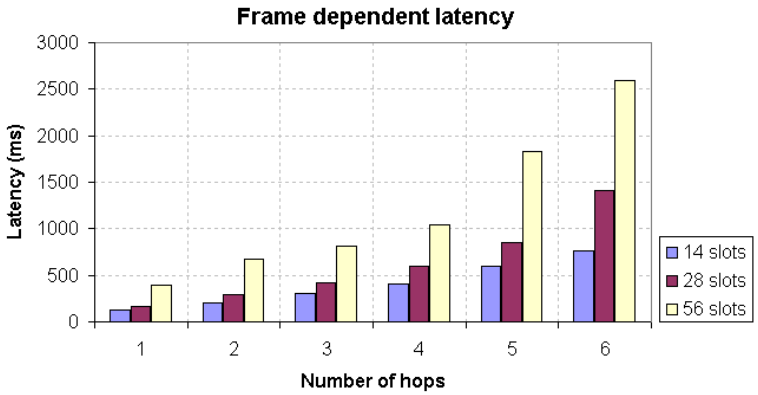


Figure 10.5: Latency in linear topology

Recall that Dust Networks are planning to reduce their current slot size with 50%. This will reduce latency considerably and possibly halve the values of the graph illustrated in figure 10.5. More rapid reporting will also be possible with the same number of slots.

10.5 Power Consumption

Power consumption is affected by the same factors as the performance metrics already discussed. The frame length is one of the most important factors in this matter because it decides how much time a mote may spend in sleep mode. Recall that a mote only has to be powered on in the time slots it is assigned for communication. More time slots results in more sleep and thus also conserved power.

Another important configurable parameter is the rate at which data is reported. A long report interval results in more time between each transmission and thus a longer period in sleep mode, which in turn results in lower power consumption.

Finally, the power consumption is also affected by the RF environment. In a harsh environment including signal interference and RF barriers, a mote is likely to experience failed transmissions at a higher rate. This result in more retransmissions and less time spent in sleep mode. Hence, the power consumption is much higher than it would have been in an

environment with path stability close to 100%.

Measurements on power consumption were carried out along with the tests on the linear topology. When these measurements were performed, the multiple frame capabilities of the network were still not discovered. Some of the measurements were therefore very confusing, since a long frame length sometimes, but not always, resulted in higher power consumption than measured with a shorter frame. The reason for this behaviour is now clear. When the frame exceeded a certain length, the measurements actually included two simultaneous frames, and not one long frame as was assumed. Sometimes when new measurements were performed, the network had been running long enough for the second frame to turn off again, resulting in much lower power measurements than earlier.

The power consumption measurements are illustrated in figure 10.6. Recall that the linear topology was installed in an optimal configuration, which should result in lower power consumption than in the average network. Also note that the first light coloured bar, drawn with a solid line, is the only measurement that consists of a single 56 slotted frame. The other bars, drawn with dotted lines, indicate measurements collected while two frames are running simultaneously (56 slots and 31 slots). Still, the two frames results in lower power consumption than the 28 slotted frame in the cases of the four closest motes. The report interval during all measurements was three times the number of motes.

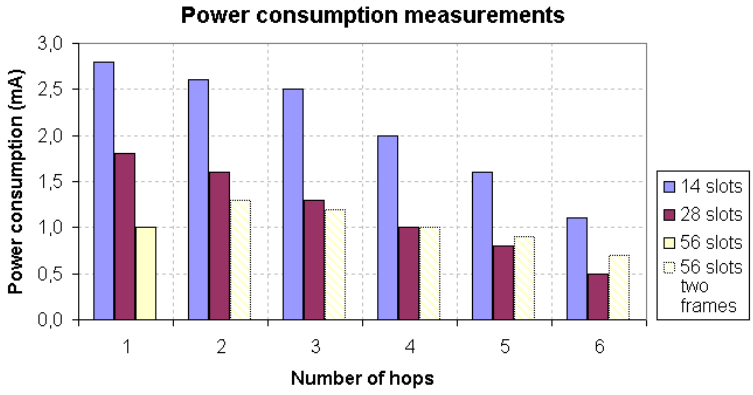


Figure 10.6: Power consumption measurements

During a web seminar [12] held by Kris Pister, it was mentioned that Dust motes consume as little as 25 μA of current and that even heavily-burdened routing motes, handling traffic from dozens of neighbours, typically consume less than 200 μA. In the experiments performed in this project, however, only a leaf mote consumed less than 200 μA of current when configured with a 56 slotted frame (roughly calculated). Still, when a network with several hundred motes is considered, it is likely that the mentioned current consumption can be achieved. But networks with frames as large as these are limited to slow reporting and high latency. The solution to this problem already exists and will be made available for the end user in the near future - multiple simultaneous frames will make it possible to ensure fast reporting for some motes, while other motes may conserve power utilizing the longer main frame.

10.6 Network Topology Calculations

Based on knowledge about link assignment in different topologies, it is possible to make general equations for minimum frame length and minimum report intervals. The general network illustrated in figure 10.7 can function as a basis for this matter.

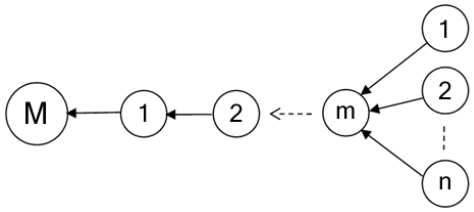


Figure 10.7: General simple network

Motes which are placed on close locations will try to mesh together. There are two ways to avoid this; either by using CLI commands or by trying to physically block the path between the two motes. The latter is a hard task because RF signals may be reflected and twisted in ways that are hard to foresee. In such scenarios it is recommended to notice all motes with more than one parent and restart the network with an extended frame length. One extra path can be viewed as one additional mote. With m greater than zero (m>0), mote 1 will require more links than the Manager. The exact number of links can be calculated by adding the links from the Manager to mote 1 with the links from mote 1 to its child. Mathematically this is given by the following: (n + m) links from mote to manager + (n + m - 1) links from child to mote + 2 broadcasts + 1 open listen + neighbour discover + 2x number of extra paths. Letting n+m equal N, leaves us with equation 10.1. The equation assumes that no extra paths are connected directly to the Manager. In the case where m=0, the Manager will be the device in need of most slots. Equation 10.2 represents this scenario.

$$MinimumFrameLength_{m>0} = 2N + 3 + 2x \tag{10.1}$$

$$MinimumFrameLength_{m=0} = N + 4 + x \tag{10.2}$$

The general network can be rearranged in a number of configurations. For example, a linear topology is obtained by inserting n = 0, while a pure star network is achieved by setting m=0. To attain more advanced topologies it is possible to use the same formula and connect an equal general network to any of its motes. Letting N represent all motes in the network makes equation 10.1 valid for such network as well. If more than one general network is connected directly to the Manager, the Manager may be the device in need of most links. This occurs if two or more equally[1] sized networks are connected directly to the Manager. When the minimum frame length is found, it is an easy task to find the minimum report interval. As the experiments have shown, the minimum report interval is dependent on topology, number of routers and RF environment. To keep the network reliable it is recommended to expect a path stability of 50%, resulting in a report interval about twice as long as the frame length. Recall that Dust Networks recommend a report interval with a length three times the frame

[1]The connected networks only have to be equal in size if they are two.

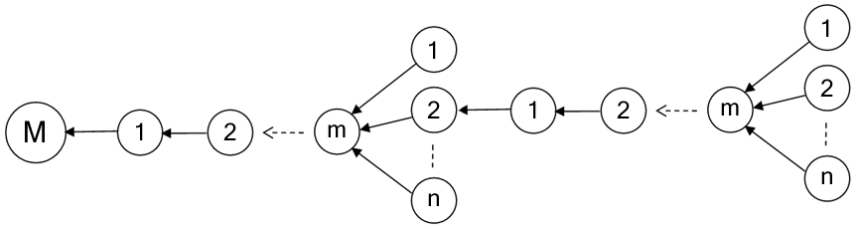


Figure 10.8: Connected general networks

length. In a network were no routers are present and no extra bandwidth is added this might be necessary unless great RSSI signals are obtained.

$$MinimumReportInterval = 2xMinimumFrameLength \tag{10.3}$$

10.7 SmartMesh-XR versus ZigBee

Based on knowledge of the medium access methods used the in the two protocols, ZigBee is expected to achieve higher throughput and less latency as long as the network is small. On the other hand, a CSMA/CA based protocol can give no guarantees to any of these parameters. With SmartMesh-XR, everything is a matter of configuration. Assuming correct settings and installation it will collect data both reliable and deterministic. The results retrieved from the Industrial Experiment do not prove theory wrong.

Throughput and Latency

The IZT defined a packet as lost if no ACK was received from the gateway within a specified amount of time. The project report[33] does not, however, explain if or how retransmissions are implemented in the application. To compare the results of the SmartMesh network and the IZT, it is important to ensure that conditions are identical. This includes identical definitions of packet loss and other performance metrics. Recall that a packet in the SmartMesh protocol is defined as lost if it is not received within the expected report interval. Hence, the two definitions are at least based on a similar strategy.

In the discussion chapter of the IZT report[33] it is written that packet loss was observed when the report interval was reduced to about 400 ms. Recall that the lowest possible report interval of the SmartMesh Network was found to be 344 ms, maintaining 100% reliability (no lost packets). The IZT kept reducing the report interval even though packets were lost. However, a reduction in the report interval from 400 ms to 100 ms only resulted in approximately 25 additional lost packets. From this point, further reductions resulted in a much higher number of lost packets. A climax occurred when a report interval of 55ms was reached. From 55 ms to 20 ms the packet loss was increased with 400 lost packets.

The above section leaves no room for doubt. For a network with tasks in need of high reliability, ZigBee cannot use a report interval below 400 ms. SmartMesh-XR on the other hand, maintained a 100% reliable network even with a 344 ms report interval. However, if time is of the essence, latency must be added to these numbers. Latency per hop is illustrated

in figure 10.9, making it obvious which network makes the fastest data collection. At least if a multihop network is installed. The IZT did some throughput calculations regarding both

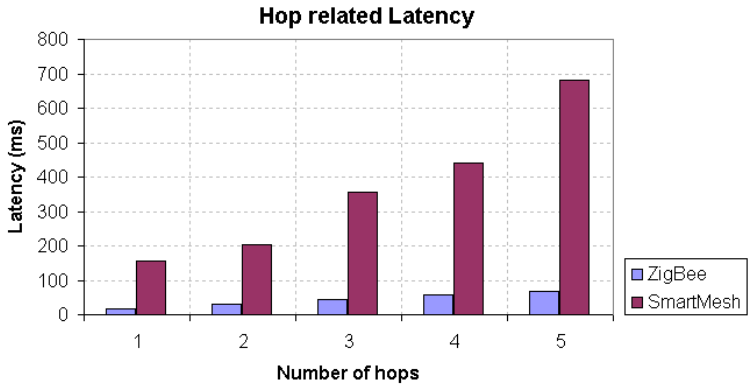


Figure 10.9: Comparison of hop related latency

complete packets and actual data payload. Calculations were done with a report interval of 100 ms, meaning that a few lost packets were accepted. With both aps-ACKs and mac-ACKs included, it was calculated that the gateway must handle data traffic of at least 18k bit/s. The actual payload that reached the gateway was found to be approximately 2.2k bit/s, pointing towards the fact that 88% of the data flow in the network is overhead. In the SmartMesh network multiple transmissions may occur simultaneously due to frequency hopping. To make accurate calculations, it would also be necessary to add the traffic caused by neighbour discovery, health reports and the like. The below equations are identical to the ones in the IZT and do not include this traffic. Calculated values are only meant to give a rough idea of the different network throughput. As mentioned in chapter 3, a data packet consists of at most 128 bytes with 80 bytes of payload. In the experiment the analog inputs were used, limiting the packet size to 66 bytes (18 bytes payload). ACK packets are 12 bytes long. Hence, the Manager must handle a data flow of at least,

$$TPraw = 3motesx(66 + 12)bytesx(1000/344)Hz = 5.4kbit/s \tag{10.4}$$

Compared to the ZigBee gateway the Manager has much less data to handle per second. However, if no packet loss is allowed, a 400 ms report interval must be used instead of the 100 ms interval in the IZT equations. This results in a data flow of about 4.5k bit/s for the ZigBee gateway. The fact that the 3 SmartMesh motes are actually sending 6 samples while the IZT nodes only sent one, additionally levels the difference. With just one sample per packet the Manager would receive a data flow of about 4.7k bit/s, leaving the two gateways with approximately equal data flows to absorb. The actual payload that reaches the Manager consists of 18 bytes (6 analog samples). Actual payload throughput to the Manager is then,

$$TPpayload = 3motesx18bytesx(1000/344)Hz = 1.2kbit/s \tag{10.5}$$

To make the ZigBee motes report at a reliable rate, a 400 ms interval must be switched with the 100 ms interval used in the equation. This will result in a 0.5 kbit/s payload throughput.

Mote Range

Detailed experiments concerning mote range were not performed due to lack of LQI measurements. However, the SmartMesh network with identical mote placement as the IZT indicated equal, if not better mote range. As an example the concrete wall was easily penetrated when good mote positions were eventually found. Recall that centimetres can make a huge difference, and there can only be assumed that the IZT also strived to find the best mote positions possible.

Starvation

The previous chapter revealed that starvation is indeed possible even in a deterministic network like SmartMesh-XR. This is, however, only achieved with unintelligent network configuration. In the experiment, all motes where given reporting profiles with x1 report interval. This assumes 100% path stability or plenty of bandwidth for all motes. At the time of initiation, neither of these requirements was fulfilled in the network. The result is congestion and a stop in generated packets. To explain exactly what happens, it is convenient to take a closer look at the farthest motes. These motes are the ones most affected by network congestion.

The two motes farthest away from the Manager is the wheel mote with its connected router mote. When the router is given a reporting profile, it starts sending one packet to the wheel mote in each frame. The wheel mote now has to use both its links per frame to forward data from its child in addition to its own data. When the first packet failure occurs, three packets will be available in the mote's queue during the next frame. There is not enough bandwidth to send more than two packets per frame, resulting in one additional packet in the queue each time packet failures are encountered. Eventually, the queue is filled up and the mote will stop generating samples. If the child tries to send a data packet at this time, it will receive a NACK instead of an ACK, filling its own buffer. Because this mote only has one link, packets are also added to its queue each time packet failures are encountered. In a multihop network, this may result in huge delays, and like in our case, congest the network to a degree where the retries/timeout limit is breached. Hence, motes are reset and must rejoin the network.

Summary

Even though the minimum report interval is 344 ms it is possible to sample data at lower intervals. This is achieved by data sampling at higher rates than the report interval, and transmission of multiple samples per packet. Six samples are the maximum when the analog inputs are sampled. Thus, the minimum interval at which data can be logged is every 57 ms (in this particular experiment). Table 10.1 summarizes the other performance metrics of the two experiments.

The latency provided in table 10.1 deviates from the latency illustrated in figure 10.9. This is because it presents the latency at the time of maximum report rate and only three reporting motes, while figure 10.9 is collected with an x3 report interval to provide reliable reporting for all motes. The throughput for raw data is calculated with reliable report rates and 6 data samples for the SmartMesh motes. This is also the case for the payload throughput. See the section about "Throughput and Latency" for more information.

Performance metric	SmartMesh-XR	ZigBee
Minimum report interval (reliable)	344 ms	400 ms
Minimum report interval (allowing a few lost packets)	344 ms	100 ms
Average latency	380 ms	44 ms
Max latency (5hops)	900 ms	68 ms
Minimum latency (1 hop)	120 ms	17 ms
TPraw (to gateway/Manager)	5.4 kbps	4.5 kbps
TPpayload (to gateway/Manager)	1.2 kbps	0.5 kbps

Table 10.1: Comparison table for the industrial experiment

10.8 Distributed Network Control

For the time being, a centralised unit, referred to as the SmartMesh Manager, is responsible for all control and management tasks in the SmartMesh Network. The concept of centralized control is also utilized by the SmartMesh competitors, but the topic of distributed control is still interesting because of the advantages it would provide. For one, it would relieve the centralized unit for a huge amount of work if motes were able to tend to simple management tasks. This would in turn make it possible to use a cheaper and less powerful Manager in terms of CPU and memory. Such solutions have been discussed by Sensicast and probably also by Dust Networks and other WSN providers. The solution Sensicast discusses is to provide certain motes with bandwidth of their own, which they may then distribute to other motes as they see fit. This will also result in less network traffic because all information no longer has to be routed all the way to the Manager. In the discussed strategy this only occurs when a mote needs additional bandwidth or is applied to a reporting task.

If distributed control is implemented based on the discussed strategy, the network is actually divided into self-contained sub networks. Further, if these sub networks are configured with frames of different lengths that applies to their tasks, many performance metrics can be extensively improved. Latency is one of these metrics that might be greatly reduced by distributed controllers (special motes), but then again, it all depends on the network applications. Not all networks and sites will benefit from a distributed strategy (e.g. small networks where the added complexity only increases cost and CPU usage).

10.9 SmartMesh-XR Compared to Other WSN Protocols

For industrial applications there are three requirements that are of particular interest. The most important one is reliability. Without reliability none of the other performance metrics matters, because data loss will make the network inadequate for most tasks anyway. Next we have the requirement for low power consumption. To collect data from rotating devices or from hard to reach locations without nearby wired power supplies, it is required that the motes can be powered by batteries (preferable in the order of years). This also makes the network more flexible and suitable for most sites. The last requirement is that of latency. In some applications, like actuation and control, it is important that the delay between a transmission and reception by the intended recipient won't exceed a certain limit. Data that are received too late may in some cases be discarded altogether. Based on these requirements, the SmartMesh protocol will now be compared to its competitors.

Reliability

To sidestep RF interferers and RF barriers, all the new protocols discussed in this report comprises frequency hopping. This also indicates that all protocols implement some form for synchronization. Both the SmartMesh-XR and the SensiNet protocol implements timed and slotted communication for all nodes in the network. This avoids collisions and ensures collision free transmissions. When it comes to network formation their strategies are slightly different. SensiNet utilize CSMA/CA to achieve faster joining times while the SmartMesh protocol uses special "listen" slots in the (TDMA) frame. In a dense network where many motes try to join simultaneously, this may results in collisions and slower joining in a SensiNet network than in a SmartMesh network. However, collisions may also occur during the joining process of the SmartMesh network if multiple motes try to send their join request in the same time slot, but recall that a slotted approach minimizes this risk considerably (slotted ALOHA). In addition, the SmartMesh protocol also utilizes multiple frames during the joining process to enable faster join times. The join sequence is mentioned in this section because it may be critical for performance that nodes re-join as fast as possible if they are somehow disconnected.

The Wavenis protocol has a different communication strategy than both SmartMesh-XR and SensiNet. During normal operation it makes use of CSMA/CA, only utilizing slotted communication after a polling request from the gateway. The operating procedure involves periodic awakening of all nodes. During this period a node typically listens for transmissions, or initiate communication if it has data to send. In a dense network with frequent reporting, this may result in an unacceptable number of collisions. To achieve deterministic communication for all nodes it is therefore necessary to poll for data, resulting in a lot of overhead in the network. Note that normal operation in a HART network is actually based on polling [22].

All protocols implements retransmissions and support a reliable mesh topology.

Power Consumption

The reason that all the discussed protocols implement synchronized communication is not solely due to frequency hopping, but also a means to enable duty cycling. This allows nodes to enter sleep mode when they are not assigned for communication and thus conserve a considerable amount of power. The SensiNet protocol is the only protocol that defines a wire-powered backbone structure in their network. This is implemented in order to increase the output power and thereby also the range of the router motes.

Since this project only involves physical experiments with the SmartMesh network it is hard to make detailed comparisons on power consumption. In general, however, TDMA is known as one of the best approaches to achieve a low powered network because all communication is scheduled prior to normal operation.

Latency

To achieve long-life battery powered nodes it is necessary to implement power saving schemes like duty cycling and reduced output power. The drawback with duty cycling and synchronized communication is that latency is sometimes increased to a level where the network might be inadequate for many control and actuation tasks. In section 10.7 the latency of SmartMesh-XR was compared to the latency of an identically installed ZigBee network, con-

figured without beacons and synchronization. The graph illustrated in figure 10.9 shows that the difference is huge.

There are a lot of trade-offs involved with the latency requirement. In both SmartMesh-XR and the SensiNet network, latency depends on the configured frame length, which in turn decides the power consumption of the network. The same concept applies to the Wavenis network where the configured wakeup period has the same function as the slotted frame. Knowing this, it is hard to determine which protocol has the lowest latency. Bear in mind though, that the SmartMesh protocol may have an edge over its competitors when its multiple frame capability becomes enabled for end users.

10.10 Portable Monitoring Equipment

In many scenarios it may come in handy to be able to use a portable utility to check network status and information while being in the field. The SmartMesh Network actually implements solutions for this even now. With the Manager connected to a wired or wireless LAN you can log in and check the network status with a laptop or PDA as long as there is a LAN network in range. However, this is hardly the case at all sites where the SmartMesh network is deployed. Hence, it is necessary to find a solution that covers all sites in the cheapest and easiest possible way.

The best solution would be to make special equipment that can connect directly to the network motes and by this collect information about the entire network. Since the network motes know nothing about the rest of the network, it is necessary to send a special command all the way to the Manager. In return the Manager could provide the mote with the necessary information. To speed up the process, a special fast (short) frame could be used for this purpose just like the one used for fast formation.

The other solution is to keep information about the entire network in all network motes and let the Manager update the motes every time a change occurs. This may require additional memory to store the information and thus increased costs per node. More protocol details must be provided to carry on with this work.

10.11 Is the SmartMesh Protocol Suitable for Wireless HART?

To make an assessment on the use of SmartMesh-XR as a basis for wireless HART it is important to have clear understanding of what HART really is, the properties it comprises and the requirements that are typical for the protocol. This section will thus start with a brief introduction of the HART technology.

10.11.1 HART - Highway Addressable Remote Transducer Protocol

The HART Protocol dates back to the early 80s where it was developed by Fisher Rosemount as a means to communicate with smart field instruments. The principle of the technology, and the property that has made HART as wide spread as it is today, is that it allows digital communication in parallel with an analog signal (4-20mA) and on the same pair of wires.

There are two different operating modes available; point-to-point and multidrop mode. In

point-to-point configuration, a traditional 4-20mA signal is used for the process value while other data and configuration parameters are sent via the digital HART signal. Since the 4-20mA signal requires a closed loop without interference from other analog signals, only one HART compatible device may be used in this configuration. In contrast, the multidrop configuration only utilizes the digital signal. This makes it possible to connect up to 15 addressable devices on the same pair of wires. Note that it takes approximately 500 ms to collect information from a device. Hence, if there are 15 devices in the network it takes about 7.5 seconds to contact and collect the primary process variables from all devices. Without going into the details, the ”long” delays are caused by a maximum transmission rate of 1200 baud (UART technology in the PHY/MAC layer).

In most applications the typical communication mode for HART is the request-response mode where a host device, often referred to as a master, sends a request to the device it wishes to communicate with. That device may then respond with the requested data or with an acknowledge message if it was ”told” to change its parameters etc. The other communication mode is burst mode, allowing more rapid transmissions (3-4 reports per second). In this mode the master sends a special command to one of the devices in the network, instructing it to continuously broadcast a certain HART message (e.g. the process value). The master can then receive the message at high speed until it asks the device to stop its transmissions.

In a project prior to this master thesis, ZigBee was used for wireless transmission of HART between a pressure transmitter and a laptop [22]. This project can be used as a reference for more information about HART and show how a wireless protocol can be implemented with the older technology.

10.11.2 Requirements for Wireless HART Compared to SmartMesh-XR

Reliability

The HART protocol has issues on its own, like timing and synchronization of messages, handling of multiple masters (two supported), transmissions ending with phantom bytes and sometimes, although rare, gaps and dribbles between bits and bytes [22, 30]. Solutions to all of these problems are implemented in the protocol. The big question, however, is whether or not it will stay reliable in the new wireless version. Without reliability, the wireless upgrade is to no use and might as well be excluded altogether.

The experiments of this project have shown that the SmartMesh protocol is highly reliable as long as guidelines are followed for installation and configuration. Frequency hopping makes it robust to RF barriers and RF interferers at a long-term basis. In contrast, a single channel network may experience decreased performance in the presence of future changes in the environment. Further, it includes a reliable mesh topology that ensures that data always has an alternative path if one path is blocked due to mote failure and blocked signals etc.

Scalability

The maximum number of instruments in a HART loop is achieved in multidrop mode where up to 15 devices makes use of the same pair of wires. When the devices are connected together like this, it is actually sufficient with just one wireless module in the loop. However, higher bandwidth is achieved if all devices are equipped with transceivers of their own. The trade-off

between bandwidth and cost is important and may be weighted differently for various systems. To further complicate matters, a site is likely to have more than one HART-loop, which in turn may have different requirements and priorities. Since the size and density of different sites may vary, it is important that the wireless protocol is both scalable and flexible. A basis requirement is that it can handle both small and large networks without having the combined cost of motes and Manager(s) exceeding the initial cost-savings.

The SmartMesh network can handle up to 250 motes per Manager. When more motes than this are required, frequency hopping makes it possible for several networks to coexist in the same area. In small networks that just need a few motes, the SmartMesh Manager may be a too expensive to benefit the installation. An ordinary mote may then be used in its place as long as the network size doesn't exceed 20 motes.

To cover large areas the SmartMesh network is capable of multi-hop routing to collect data from motes more than 10 hops away. When motes are located beyond this distance it is recommended to install multiple networks to ensure fast and reliable data collection. Recall that latency and formation times are strongly affected by the number of hops in the network.

Power Consumption

HART compatible devices are powered by an external power source just like traditional 4-20 mA equipment. When configured in multidrop mode the loop current is adjusted to a minimum, typically set to 4 mA. Motes that are placed close to or in this loop may steal power and charge their batteries utilizing some sort of interface. For other wireless motes we have the same trade-offs that have been discussed throughout the report and summarized in section 10.5. For all networks in general it is preferable with a battery-life in the order of years. If batteries must be changed frequently, the initial cost savings will be lost and we risk expensive downtime of the system (section 2.3.1)

With a TDMA based protocol like SmartMesh-XR, power consumption depends on network size (frame length), network topology and the report rates of the motes. The graph depicted in figure 10.6 shows power consumption in a linear topology consisting of 6 motes. Combined with the other results discussed in the chapter, this data can be used to illustrate other topologies and network sizes as well. In a star network for example, a frame consisting of 28 slots is capable of running 24 motes (in theory). The resulting power consumption and latency, assuming an x3 report interval, will then be 0.5 mA and 129 ms respectively (for all motes). Similarly, 56 slots in the frame results in 52 star motes, about 290 ms latency and less than 200 uA power consumption. In other topologies power consumption will increase drastically for motes close to the Manager, but with even larger frames it is still possible to avoid wired power.

Latency

The HART protocol is relatively slow (1200 baud) and has response times (latency) varying about 500 ms. In burst mode, where a HART device is continuously reporting data, latency is reduced and varies between 250- and 300 ms depending on the message length. Since the latency of HART is fluctuating and relatively high it is assumed that the protocol can tolerate some additional delays. The upper limit for these delays may vary depending on the application where the collected data is used. For this reason the wireless extension should

involve some kind of priority scheduling.

The latency of SmartMesh-XR was discussed in section 10.11.2. To summarize, there are two configuration parameters that have huge influence on latency; the network size (frame length) and network topology (number of hops from a mote to the Manager). This was illustrated in figure 10.5.

A typical operating system (OS) requirement is that the average latency experienced by the end user should vary around 2 seconds or less [29]. In an automation network this strongly depends on the network applications. It is obvious that a pure monitoring or logging application doesn't have the same requirement to latency as a hard real-time application (e.g. control and actuation). However, if logging or monitoring is considered using the OS requirements as a basis, the graph in figure 10.5 shows that the limit is breached for a mote 6 hops away when the frame is configured with 56 slots. When the latency of wired HART is included the average latency is actually above 3 seconds. Note that a frame consisting of 56 slots in theory may include 26 network motes (N = 56 - 3/2) when several hops are allowed.

Recall that it is planned to halve the current slot size and enable the multiple frame feature for the end user. The former will probably halve the current latency, but also increase power consumption in regards to the measurements of this report (if frames are re-measured with the identical number of slots). With multiple frames it will be possible to reduce latency and minimize report intervals for certain tasks in the network.

Report Rates

The maximum report rate for a HART compatible device is achieved in burst mode resulting in 3-4 reports per second. Thus, a wireless mote must be configured with the report interval set to 250 ms if instantaneous reporting is required. In a TDMA based protocol like SmartMesh-XR this would require a really short frame resulting in a small network only consisting of a few motes. Recall the minimum report interval (344 ms) achieved in the industrial experiment, discussed in section 10.7. Note that this report interval was possible solely due to "extra" bandwidth provided by router motes, and the fact that only three motes were reporting data. Under normal circumstances an x1 report interval is unlikely to be initiated without decreased reliability.

One possible solution to this problem is to allow motes to collect several samples before they initiate a transmission. Since the HART message may vary in size, typically above 20 bytes [22], it may be necessary to find a smart solution to reduce their length in order to send several samples in the same packet. A possible solution is to extract the actual payload data from the HART messages before they are included in "wireless" packets, but this is beyond the scope of this report.

In addition to frame length and number of reporting motes in the network, there is another important factor that limits the report intervals. Frequency hopping makes it possible for several motes to communicate simultaneously in the same time slot utilizing different channels in the 2.4 GHz band. In a frame with 20 slots assigned for communication, that means that there is actually 320 (20 x 16) available links. However, the Manager can only collect data from one mote in the slot at the time and thus limits the capacity of the entire network. The concept is most easily illustrated with the use of a star topology.

In theory, a star network with a frame consisting of 28 slots is capable of running 24 motes. When the 24 motes are connected, all slots in the frame are assigned with links involving the Manager. As a result, each mote can report data only once for the duration of the entire frame. The recommended report interval, three times the frame length (x3), is necessary to ensure high reliability in the presence of interference and multipath fading (since no additional bandwidth is available). In total this results in a minimum report interval of 2.6 seconds. The Manager is limited to the same number of slots regardless of the topology and the configuration of the connected motes. Hence, the only way to reduce the report interval is either to free bandwidth by reducing the number of reporting network motes, or to risk reduced reliability by reducing the report interval below the recommended threshold. Note that when the time slots are cut in halve, the report intervals will be equally reduced.

Formation Times

The join times of the network are important during start-up and if the network for some reason has to be restarted during normal operation. In the situation where a network reset is unplanned, but necessary to continue normal operation, the time it takes to rejoin all motes must be considered as downtime which can be expensive. To prevent unintended resets in the presence of Manager failure, the SmartMesh network supports redundant Managers.

Fast joining is important in other situations as well. Maintenance of motes, like change of batteries or complete replacement, may require that the mote is turned off during the repair. To keep the repair time as short as possible it is important that the repaired mote can rejoin the network and continue its task(s) as fast as possible. Equally important, if the network is going to provide support for portable monitoring equipment, as discussed in section 10.10, it is a criterion that the joining process won't exceed a certain limit (about 1-2 minutes). Note that a few minutes probably feels like an eternity for stressed maintenance personnel.

The joining and formation times of the SmartMesh network are strongly connected to frame length and network topology. To enable faster formation times, a short (fast) frame is initiated during start-up and in the event of joining motes etc. The fast frame and the main frame are then running simultaneously, but without degrading the performance of existing network applications. In cases where both frames are assigned to the same slots, the main frame is at all times prioritized over the shorter and faster frame.

During the experiments it was shown that different network topologies results in very different formation and joining times. The best joining times occur in a star network where all motes are likely to join within just a few minutes. In a linear/multihop topology, however, the joining process is a lot slower - close to 20 minutes. A compromise is made with the mesh topology, resulting in formation times between 10 - 12 minutes in the average case. More details were discussed in section 10.11.2.

Final Discussion and Conclusion

The different requirements that have been discussed in this section should make a good basis to determine whether or not the SmartMesh protocol is suitable for wireless HART. When it comes to reliability, the experiments and protocol analysis show that the SmartMesh protocol is highly reliable as long as the guidelines for installation and configuration are followed.

Power consumption and latency, however, strongly depends on the frame length in addition to network size and topology. In a network with applications that require little latency it may be necessary to configure the network with a short frame, and as a result, power consumption is increased for all motes. The possible frame reduction is in turn depending on the number of motes in the network. To complicate matters even further, there may be different sub-networks that have different requirements to the different performance metrics. The next version of the protocol will handle these tradeoffs better than the current version due to enabled multiple frame capabilities and halved slot size. Still, it is possible to achieve a low powered network with the current software - at least as long as the requirement for latency is relaxed and in the order of a few seconds.

The report rates of SmartMesh-XR are usually much lower than that of wired HART. In fact, it is actually impossible to achieve report intervals as low as 500 ms in a medium sized network with more than 12 reporting motes (and the current slot size). Recall that even though frequency hopping provides additional bandwidth, the Manager can still collect data from only one mote at the time. Hence, to utilize SmartMesh-XR to transfer HART messages, the system must either allow relaxed report rates or limit the network to a minimum of reporting motes. Adding distributed features to the protocol may help to relieve the Manager and provide additional bandwidth for reporting motes. Thus, make faster reporting possible.

The formation and join times in the SmartMesh networks vary depending on topology and frame length. In the average case, a mesh network with 12 motes connected within 10-12 minutes. The time it takes to connect a single mote varies depending on the frame size, the number of nearby motes and the number of hops these motes are from the Manager.

Apart from the limitations already mentioned, the SmartMesh protocol comprises a lot of good features and properties. Its competitors have not been tested during this project, but given the protocol analysis there is no reason to believe that any of these will perform better than the SmartMesh network. The final conclusion is therefore that if the limitations for latency and report rates can be accepted, SmartMesh-XR is a good and reliable alternative for wireless HART.

Appendix A

The Industrial Experiment

This chapter gives a brief description of the SmartMesh evaluation kit that has been used in all experiments carried out in this project. In addition, it will describe the industrial environment at Statoil and the interface cards that had to be made to connect motes and sensors.

A.1 The SmartMesh Evaluation Kit

The SmartMesh evaluation kit consists of twelve motes, one Manager and a CD containing the necessary client software to manage and control the network. To make the user able to connect a fully functional network without having to buy additional equipment, all necessary cables and tools is provided with the kit. In addition to the in-built temperature sensors of the motes the kit comes with three sensor plug-ins, each containing a temperature sensor, a light sensor, a 2-axis accelerometer, and an LED. Using the provided client software, the SmartMesh Console, it is up to the user to select which sensors to collect samples from. This task is carried out by making profiles as explained in section 3.3. To get the network up and running as fast and easy as possible, a guide [19] is provided which describes all details about installation and configuration of both motes and Manager.

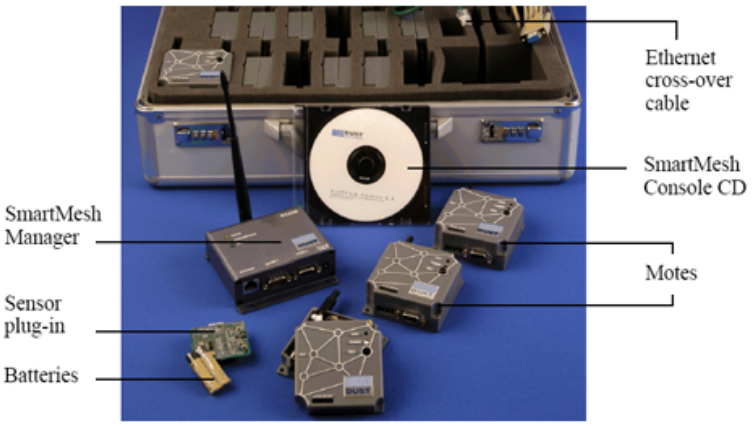
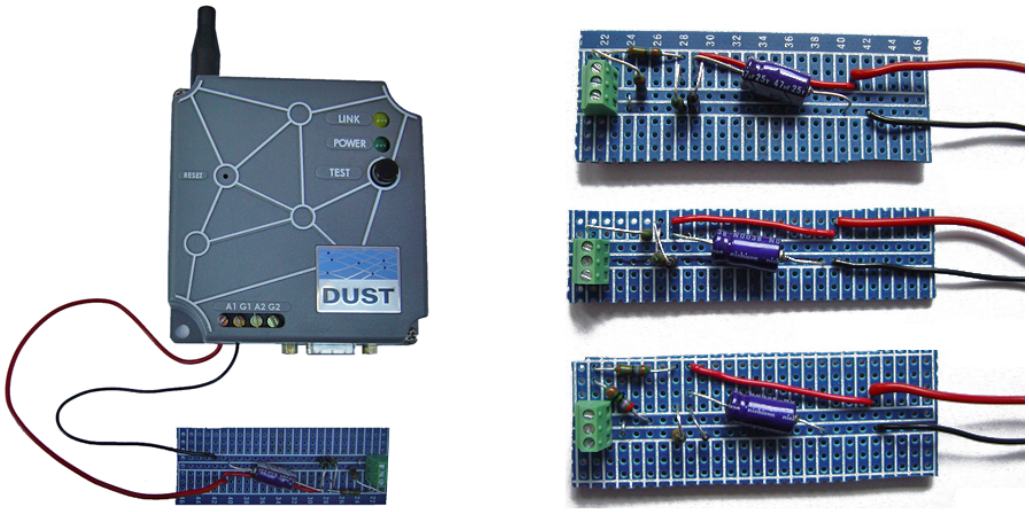


Figure A.1: The SmartMesh evaluation kit [19]

A.2 Sensor Interface Cards

In the industrial experiment at Statoil, data were collected from three external sensors in the environment. The different sensors included a torque sensor (0-30 mA), a Level transmitter (0-10 V) and a pressure transmitter (4-20 mA). To connect these sensors to the analog inputs of the motes (0-5 V), interfaces had to be made for each sensor. The different sensor interfaces and a mote connection is illustrated in figure A.2.

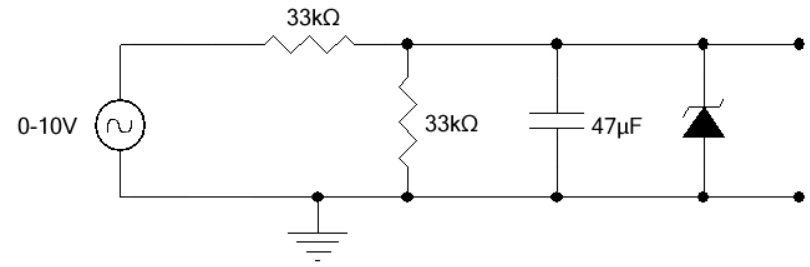


(a) Connected interface card

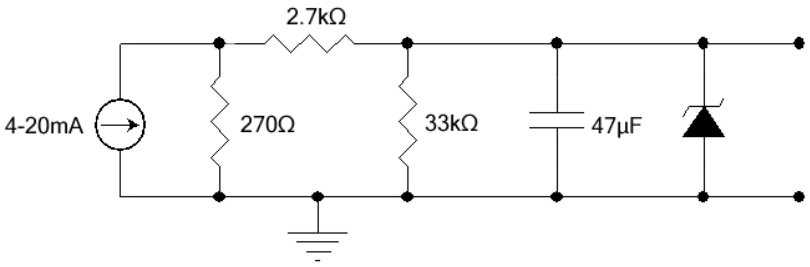
(b) The three interface cards

Figure A.2: Sensor interfaces

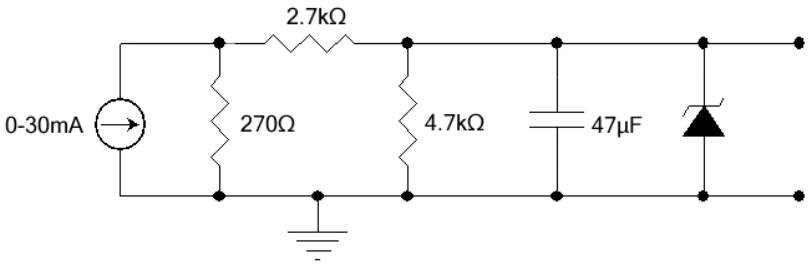
Some of the most important choices in regard to interface components were discussed in section 6.2.1. One of the most essential points of the section was that the combined resistance of the interface cards functions as a loop resistor when the card is connected to a sensor. The minimum value for a loop resistor is 250 ohm. All interface cards are designed to meet this requirement. In addition they also include zener diodes connected as shunt-regulators to protect the motes from high voltage peaks. The capacitor is connected in parallel to provide filtering of low frequency noise (-3db point just below 2 Hz). The different interface designs are depicted in figure A.3 (with component values indicated).



(a) Level transmitter



(b) Pressure transmitter

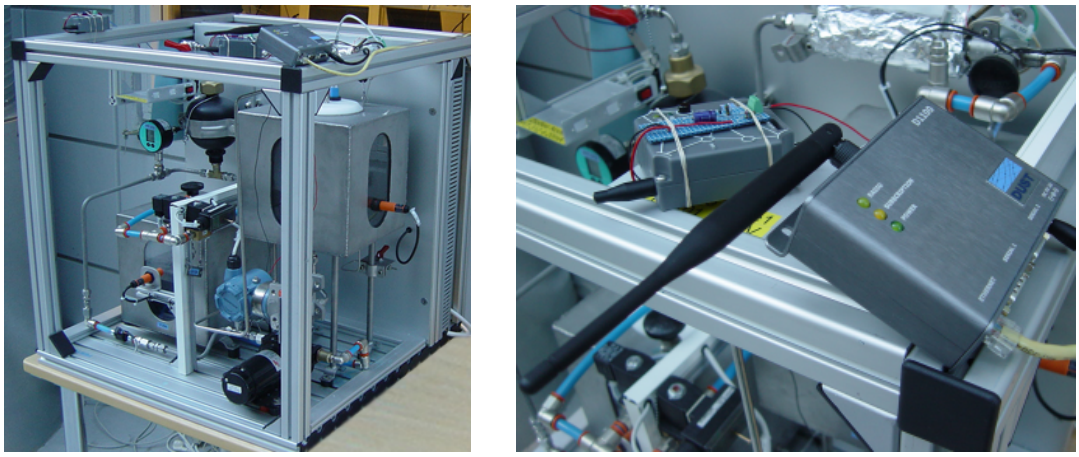


(c) Torque Sensor

Figure A.3: Interface designs with component values

A.3 Site Description and Mote Placement

The industrial experiment included three motes used to collect data from external sensors in the environment. Two of these sensors, the pressure- and level transmitters, were placed on a combined demonstration- and training construction named Ziggy. The Manager, like the motes, was placed on Ziggy as illustrated in figure A.4. A brief description about Ziggy and its functionality can be found in section 5.3.1.



(a) Ziggy (b) Manager and motes installed on Ziggy

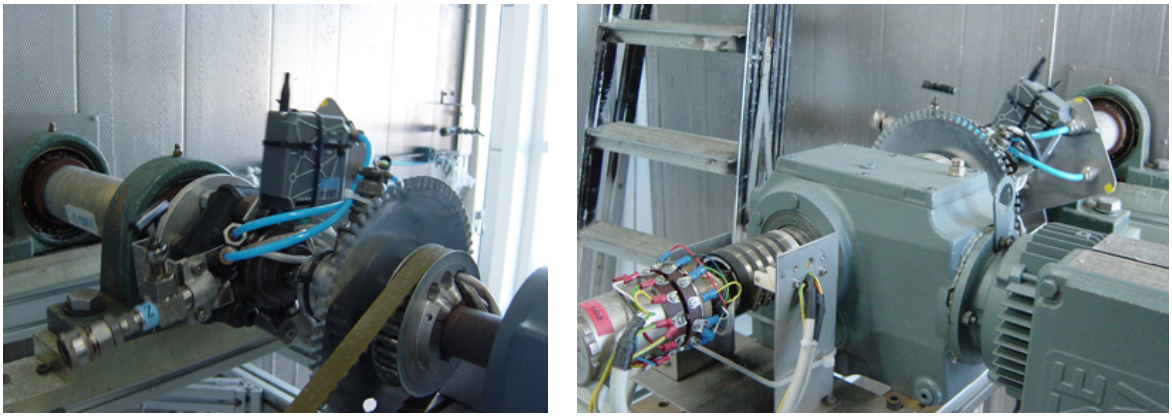
Figure A.4: Ziggy - combined demonstration and training construction

One of the greatest challenges in the industrial experiment was a safety cell with thick concrete walls and an explosion proof metal door as the only entrance. The metal door is illustrated in figure A.5(a), depicted from the inside of the safety cell. When the picture is examined closely, it is actually possible to see the first router mote through the glass of the door. The second picture (figure A.5(b)) takes a closer look at this mote. It is located in a room of its own, but with much thinner walls than the safety cell. The view from this room and in through the glass of the explosion proof door is depicted in figure A.5(c). In the centre of the round window it is possible to see the shaft of the rotating wheel where the ”wheel” mote is placed. A close-up of this mote can be viewed in figure A.6.

(a) Metal door (b) Outside the metal door (c) Inside the metal door

Figure A.5: The entrance and surroundings of the safety cell

In figure A.6 the ”wheel” mote is shown from two different angles. The placement on the wheel shaft resulted in a lot of problems, even before the wheel was set in motion. Recall from section 6.2.2 that the mote had to be placed with its antenna as far away from the metal as possible, like the picture illustrates. In the first installation, the mote was placed a few centimetres closer- and in parallel with the metal plate, resulting in a poor and unstable connection. Other engines and pumps are located in the room as well, but this is not shown in the pictures.



(a) Rotating mote (left)

(b) Rotating mote (right)

Figure A.6: Mote placed on a wheel shaft within the safety cell

Bibliography

[1] M. Ilyas and I. Mahgoub. 'Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems'. CRC Press LLC, Boka Raton, Florida, 2005.

[2] H. Karl and A. Willig. 'Protocols and Architectures for Wireless Sensor Networks'. John Wiley & Sons Ltd, The Atrium, West Sussex, England, 2005.

[3] S. Banarjee and A. Misra. 'Adapting Transmission Power for Optimal Energy Reliable Multi-hop Wireless Communication'. UMIACSTR. Technical report, 2002.

[4] Jr. Edgar H. Callaway. 'Wireless Sensor Networks: Architectures and Protocols'. CRC Press LLC, Boca Raton, Florida, 2004.

[5] Wikipedia Online Dictionary. http://www.wikipedia.org

[6] E. Undheim.'An Evaluation of the IEEE 802.15.4 and ZigBee standards for Industrial Applications'. Master thesis at NTNU, spring 2005.

[7] Beeby, S. P., Tudor, M. J., Koukharenko, E., White, N. M., O'Donnell, T., Saha, C., Kulkarni, S. and Roy, S.'Design and Performance of a Microelectromagnetic Vibration-powered Generator'. In Proceedings of The 13th Internation Conference on Solid-State Sensors, Actuators and Microsystems 1, pp. 780-783, Seoul, Korea. 2005.

[8] Chappell Brown. 'Endless energy is harvesting's promise'. EE Times, http://www.eetimes.com/. 27. February, 2006

[9] J. Werb, M. Newman, V. Berry, S. Lamb, D. Sexton and M. Lapinski. 'Improved Quality of Service in IEEE 802.15.4 Mesh Networks'. General Electric and Sensicast Systems. IWWIA, San Francisco, California, March 2005.

[10] IEEE Computer Society. 'Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)', IEEE standard, 2003.

[11] Earl McCune. 'DSSS vs. FHSS narrowband interference performance issues'. RF Design, http://rfdesign.com/mag/radio_dsss_vs_fhss/. September 2000.

[12] Kris Pister. 'Wireless Sensor Networks for Harsh Environments'. Web seminar, Dust Networks, http://www.dustnetworks.com. WINA, March 2006.

[13] Dust Networks. http://www.dustnetworks.com.

[14] Dust Networks. 'Sensor OEM Guide'. Document number: 040-0010-00C. Last Revised: 19. October, 2005.

[15] Dust Networks. 'SmartMesh Console Reference Guide'. Document number: 040-0013-01. Last Revised: 9. January, 2006.

[16] Dust Networks. 'System Integrator Guide'. Document number: 040-0011-01. Last Revised: 9. January, 2006.

[17] Dust Networks. 'SMM API Guide'. Document number: 040-0012-00B. Last Revised: 13. September, 2005.

[18] Dust Networks. 'CLI Commands Guide 1.6'. Document number: 040-0016-3 (Beta Draft). Last Revised: 28. March, 2005.

[19] Dust Networks. 'SmartMesh Evaluation Kit Guide'. Document number: 850-0009-00C (Tesla). Last Revised: 22. September, 2005.

[20] Texas Instrument. http://www.ti.com/

[21] Texas Instrument - Chipcon. http://www.chipcon.no/

[22] Nils Petter Eftedal. 'Wireless Transmission of HART via ZigBee'. Project report, NTNU, Autumn 2005.

[23] World Wide Web Consortium (W3C). http://www.w3c.org

[24] XML-RPC Home Page. http://www.xmlrpc.com

[25] Coronis Systems. http://www.coronis.com

[26] Coronis Systems.'Wavenis Wireless Technology Description'. v0.9 rev 1, February 2006.

[27] Sensicast Systems. http://www.sensicast.com

[28] Sensicast Systems. 'SensiNet Architecture Introduction'. Whitepaper, 2004.

[29] William Stallings 'Operating Systems - Internals and Design Principles'. Fifth edition. Person Prentice Hall, 2005.

[30] Analog Services. 'Online HART book'. http://www.analogservices.com/about_part0.htm, septemper 1999.

[31] ZigBee Alliance. '*ZigBee Specification*'. ZigBee Document 053474r06, Version 1.0, 14. Desember 2004.

[32] HART Communication Foundation. http://www.hartcomm.org.

[33] Dag Vegard Tveitå. 'Wireless Sensor Networks'. Master thesis, NTNU, Spring 2005.

[34] Cato Andre Jensen. 'Wireless Sensor Networks'. Master thesis, NTNU, Spring 2005.

[35] Tor Onshus. 'Instrumenteringssystemer'. NTNU, January 1997.