Master's thesis

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

Sigurd Hellesvik

Testing of mesh and cellular wireless communication in IoT sensor networks

Master's thesis in Cybernetics and Robotics
Supervisor: Geir Mathisen
Co-supervisor: Jørgen Holmefjord
June 2021

NTNU
Norwegian University of Science and Technology

Sigurd Hellesvik

Testing of mesh and cellular wireless communication in IoT sensor networks

Master's thesis in Cybernetics and Robotics
Supervisor: Geir Mathisen
Co-supervisor: Jørgen Holmefjord
June 2021

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

-

MASTER THESIS DESCRIPTION

Candidate: **Sigurd Hellesvik**

Course: **TTK4900 Engineering Cybernetics**

Thesis title (Norwegian): **Testing av trådlause mesh- og mobilnett for bruk i IoT sensornettverk.**

Thesis title (English): **Testing of mesh and cellular wireless communication technologies in IoT sensor networks.**

Thesis description:

Low power wireless communication is key in most IoT networks, as it enables flexible communication for low power devices. At medium ranges, it is unclear whether mesh or cellular wireless communication is more effective.

The four properties: current consumption, reliability, throughput and latency are important in wireless sensor networks. Based on these properties, this thesis should evaluate the following wireless communication technologies: ZigBee, Thread, LTE-M and NB-IoT.

The tasks will be:

1. Conduct a literary study of the proposed communication systems, applied on medium range IoT sensor networks.
2. Propose a system for testing medium range IoT sensor networks with respect to power consumption, reliability, throughput and latency
3. As far as time permits, implement and perform the suggested tests.

Start date: 04. January, 2021
Due date: 13. June, 2021

Thesis performed at: Department of Engineering Cybernetics
Supervisor: Professor Geir Mathisen, Dept. of Eng. Cybernetics
Co-supervisor: Jørgen Holmefjord, Nordic Semiconductor ASA

Preface

This master's thesis is taken in the final semester of my 5 years masters program Cybernetics and Robotics at the department of Department of Engineering Cybernetics, Norwegian University of Technology and Science (NTNU), Trondheim, Norway.

Sigurd Hellesvik was the student taking the masters thesis. Professor Geir Mathisen was the supervisor. The thesis is done in cooperation with Nordic Semiconductor ASA, who provided most of the hardware for the thesis. Jørgen Holmefjord, engineer at Nordic Semiconductor ASA support department, was co-supervisor. Raspberry Pies used in the thesis were on loan to this thesis from multiple different friends of mine.

Summary

Wireless IoT sensor networks often have nodes spanning distances over 1000m. Some alternatives that can cover such distances ar are Low Power Area Wireless Networks (LPWAN) or wireless mesh networks. LPWAN uses low frequencies to make signals travel longer, while mesh can use relays to send the signal via multiple nodes to cover a larger distance.

Cellular LPWAN protocols NB-IoT and LTE-M were compared against mesh protocols ZigBee and Thread, using a test system designed in this thesis.

A literary study was performed to learn more about the used protocols, as well as to help in the design of the system test. The four properties: power consumption, reliability, throughput and latency, are important for wireless IoT sensor networks, and therefore the focus of the system test.

The test system was implemented, and produced measurements for power consumption, throughput and latency. Reliability was a part of the test system design, but due to time restrictions, reliability tests were not implemented.

It was found that ZigBee and Thread mesh networks have considerably higher power consumption than NB-IoT and LTE-M, since ZigBee and Thread mesh relays can not enter sleep mode.

The test system, applicable for any low power wireless system, is the main product of this thesis.

Samandrag

Trådlause IoT sensornett har ofte noder som må kommunisere over avstander på over 1000m. Enten Low Power Area Wireless Networks (LPWAN) eller meshnett kan bli brukt for å dekke større avstander. LPWAN bruker lave signalfrekvenser for å sende signalene langt, mens meshnett bruker relenoder for å sende signalet langt ved å sende via flere noder.

NB-IoT og LTE-M er mobilnett LPWAN protookoller, som ble sammenlignet med meshnetta ZigBee og Thread ved å bruke et testsystem desgna i denne masteroppgaven.

Eit literaturstudie blei gjennomført for å lære meir om dei brukte protokollene, og for å forberede design av testsystemet. Strømforbruk, pålitelighet, datakapasitet og forsinkelse er fire viktige egenskaper for trådlause IoT sensornett, og vil difor vere i fokus for design av testsystemet.

Etter implementasjonen målte testsystemet straumtrekk, datakapasitet og forsinkelse. Pålitelighet var ein del av systemdesignet, men på grunn av tidsmangel, ble ikkje pålitelighetsmålingene ein del av den ferdige implementasjonen.

Testene viste at ZigBee og Thread har mykje høgare straumtrekk enn NB-IoT og LTE-M. Dette er grunna at ZigBee og Thread mesh releer ikkje kan aktivere sleep mode.

Testsystemet som kan brukes for dei fleste straumgjerrige trådlause systemer, kan sjåast som hovudproduktet til denne oppåva.

Contents

Acronyms

IoT	Internet of Things
LTE-M	LTE-MTC
LTE-MTC	LTE-Machine Type Communication
LTE	Long-Term Evolution
NTNU	Norwegian University of Technol- ogy and Science
LPWAN	Low Power Area Wireless Networks
BLE	Bluetooth Low Energy
LoRa Wan	Longrange Radio Wide Area Network
DK	Development Kit
SoC	System on a Chip
PSM	Power Saving Mode
eDRX	extended Discontinuous reception
RAI	Release Assistance Indicator
RTT	Round Trip Time
SIM	subscriber identification module
RSRP	Reference Signal Received Power
LoS	Line of Sight
dBm	decibel-milliwatts
OSI	Open Systems Interconnection
MAC	Medium Access Control
LLC	Logical Link Control
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
WPAN	Wireless Personal Area Network
IEEE	Institute of Electrical and Electronics Engineers
kbps	kilobits per second
6LoWPAN	IPv6 over Low -Power Wireless Personal Area Networks
CoAP	Constrained Application Protocol
MQTT	Message Queuing Telemetry Transport
BR	Border Router
DNS	Domain Name Server
RRC	Radio Resource Control
M2M	Machine to Machine
HTTP	Hypertext Transfer Protocol

Table 1: Acronyms table 1

SDK	Software Development Kit
NCS	Nordic Connect SDK
RTOS	Real Time Operating System
GPS	Global Positioning System
API	Application Programming Interface
CLI	CLIent
NCP	Network Co Processor
SSID	Service Set IDentifier
VM	Virtual Machine
CSV	Comma Separated Values
PPK	Power Profiler Kit
GPIO	General Purpose Output Input
UART	Universal Asynchronous Receiver-Transmitter
TTM	Thread Topology Monitor
RTC	Real-Time Clock
CSMA/CA	Carrier sense multiple access with collision avoidance
APS	Application Support Sublayer
ACK	ACKnowledge
CON	CONformable
NON	NON-conformable
TX	Transmit
RX	Receive
QPSK	Quadrature Phase Shift Keying

Table 2: Acronyms table 2

1 Introduction

A wireless IoT sensor network is a network of sensors that monitor physical conditions, sending data to a central location. Sensors are typically distributed at multiple locations in an area. Wireless IoT sensor networks are useful for applications where it is impractical or costly to use wired connections. Typical uses can be monitoring of agricultural plants, environment or industrial processes. In applications where wireless is used instead of wires, power cords are also not usable, and devices have to run on batteries. In addition, these devices can be distributed more than 1000m apart.

While many wireless personal area networks are power efficient enough to run on battery, such as Bluetooth Low Energy (BLE), they do not have the range to support communication over a kilometer. There are two ways to cover this larger range: Either lower sending frequency to make signals travel a larger distance, or use relays to cover the distance using multiple wireless "hops".

Using a mesh network, wireless devices can connect to multiple other devices, and there is logic in the protocol to route a signal from a device in the mesh to another. Therefore, mesh networks could be ideal for creating a multi-hop line of wireless devices to cover longer distances. Popular mesh networks include ZigBee, Thread or Bluetooth Mesh. Low Power Wireless Area Networks (LPWAN) typically use low frequencies to transmit data over long range, at the cost of bit rate. Popular LPWAN include SigFox, LoRa Wan, LTE-M and NB-IoT.

As wireless IoT sensor network devices often run on batteries, it is important that devices have low power consumption. In addition, data from the sensors should not get lost; It is important that the protocols have a high reliability. For wireless IoT sensor networks with real-time demands, high throughput and low latency would be important.

To test these four properties: power consumption, reliability, throughput and latency, a test system will be created. This test system should be a general test system, able to test and log results for different low power wireless protocols.

This thesis is written in cooperation with the company Nordic Semiconductor ASA [1], who will provide equipment for the test system. For testing the test system, a selection of low power wireless protocol will be tested. Nordic Semiconductor ASA will provide nRF52840 Development Kits (DK) and nRF9160 DKs for this purpose. Four different protocols will be tested: ZigBee and Thread as wireless mesh protocols, and NB-IoT and LTE-M as cellular LPWAN protocols. The nRF52840 supports ZigBee and Thread, while nRF9160 supports LTE-M and NB-IoT.

In order to better implement a test system, a brief literary study will be performed prior to the design.

In short, the goal of this thesis will be to design a test system to test low power wireless systems, and use said system to find if wireless mesh or cellular LPWAN is preferable for use in wireless IoT sensor networks.

2 Previous Research

2.1 The experience of using particle mesh for large IoT projects[...]

Nassr Myasar Abdullah and Rabee Hagem [2] discuss the use of an OpenThread mesh network for agricultural projects. They use the Particle Xenon microcontrollers for a mesh network. The Particle Xenon uses a nRF52840 System on a Chip (SoC). The goal of this project was to measure the power usage of a mesh network used for agricultural automation. To do this, a layout for a mesh network was proposed as seen in Fig. 1. Abdullah and Hagem report no losses in communication when nodes were placed 70m apart. They propose to install repeaters 100m apart, but state no loss-rate at this distance.

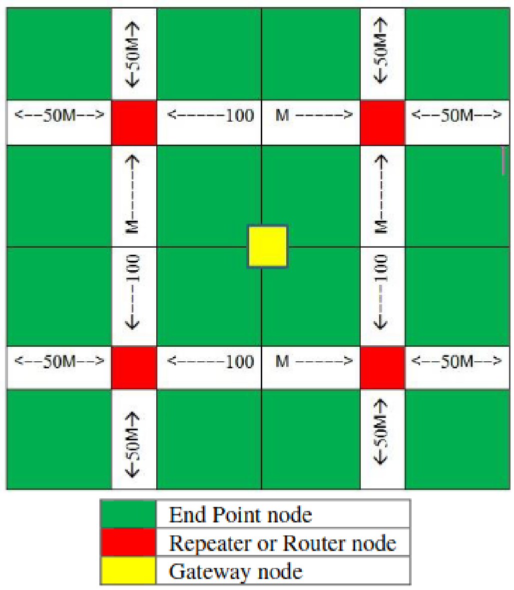


Figure 1: Layout for mesh network in an agricultural use case, from [2].

Abdullah and Hagem measure the current consumption of the Xenon for different operational states. The results from these measurements can be seen in Table 3. The measurement interval was 1 minute.

Node state	**Current [mA/h]**
Xenon in idle state without mesh	4.8
Xenon as a member in Particle mesh	5.5
Xenon when sending and receiving information in mesh	9.6
Xenon when sending information to cloud every 20s	13.5
Xenon when receiving information from cloud every 20s	13
Xenon in the Sleep mode	0.06
Xenon in the Deep Sleep mode	0.04

Table 3: Current consumption measurement for Xenon, from [2].

2.2 An empirical study of Cellular-IoT

In *An empirical study of Cellular-IoT* [3], Tim Ekman and Samuel Jönsson explains in depth how LTE-M and NB-IoT works before testing the protocols for use with IoT on the nRF9160 Development Kits(DK), with a Telia cellular connection. First, the current consumption of the DK is measured while the kits are stationary. Different settings for power saving are toggled to see how they influence the power efficiency. These are power save mode (PSM), extended discontinuous reception (eDRX) and release assistance indicator (RAI). It is found that NB-IoT uses less current than LTE-M, as seen in Fig. 2. LTE-M did not support RAI or eDRX in these tests. The report concludes that the lack of RAI in LTE-M was most likely the reason for the higher current consumption.

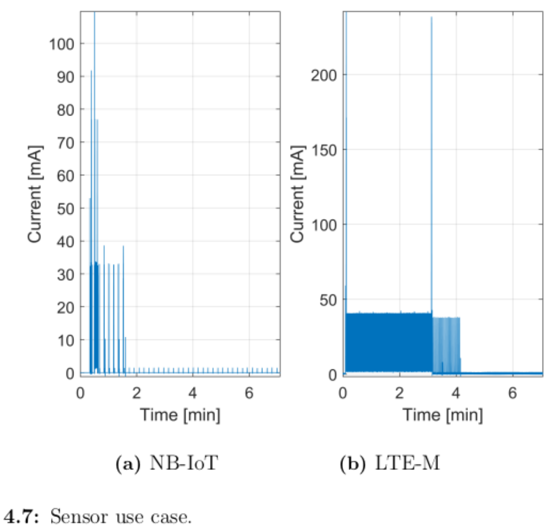


Figure 2: Power measurement during realistic example to compare LTE-M with NB-IoT. From [3].

After the stationary tests, latency during cellular handover is tested by testing the DKs in motion. Handover is when the DK changes which cell tower it is connected to, a process that will stall communication while it is ongoing. This latency was measured in Round Trip Time (RTT). This was tested with both TCP and UDP. It was found that only the combination of LTE-M and UDP was not slow, with RTT of 20s-40s, while the rest of the handover tests had RTT of around 100s. The report points out that low handover latency would be important for wearable applications.

2.3 Streetlightning: Cellular vs RF mesh

Interact Lightning is an IoT lightning brand from Signify. They specialize in city projects. In the blog post [4] by Interact Lightning, the advantages and disadvantages of cellular and RF mesh technologies are compared based on logistic factors.

The lifetime of standards are discussed; New and improved cellular technologies are released approximately every ten years, like 3G, 4G and 5G. It is costly to upgrade the system to a new technology, as hardware must be replaced. For mesh networks, the technologies in the field have proven to be long lasting, such as ZigBee which is 18 years old as of 2021, and are still widely used. It is remarked, however, that older cellular technologies are still supported in 2021, such as 3G.

Cellular networks are licensed, and a part of existing infrastructure. Because of this, failures of the communication network will be fixed by the owner of the cellular network. The cellular network has been around for a long time, and will be robust and have proven procedures for fixing failures from years of experience. On the other hand, the hardware of a mesh network has to be implemented and maintained by the developer. This will result in higher maintenance costs and risks more downtime. Mesh networks communicate over unlicensed frequencies while cellular networks use licensed frequencies. This can cause mesh networks to have downtime due to an abundance of traffic on the frequency.

Cellular operators charge for the use of their networks, often per traffic. This will increase the costs of cellular networks, while the mesh network traffic is self hosted, and therefore does not need a subscription.

For network planning, mesh nodes can often not simply be plugged into the network. Therefore, inserting a new node or changing an old one will need planning and technical skill. A cellular node on the other hand, is connected individually, and will therefore be simple to connect to the rest of the system. The blog post [4] does not conclude which technology is better, but rather that the tool should be chosen according to the use case.

2.4 Nordic Semiconductor tests of Cellular and Short range communication

On the online forum Nordic DevZone by Nordic Semiconductor multiple blogs can be found. A blog by Petter Myhre shows results from testing LTE-M over different distances from a base station. [5] The nRF9160DK is used with a Telenor SIM. Based on the distance from the base station, LTE-M will change output power, modulation and repetitions. In addition, it is possible to measure incoming signal signal quality by measuring reference signal received power (RSRP). These properties were measured each kilometer away from the base station. The results can be found in Fig. 3. The tests were done in an area with little interference from other sources.

Distance (km)	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Output power (dBm)	4	6	12	10	23	16	18	23	23	23
RSRP (dBm)	-86	-87	-90	-87	-95	-95	-97	-101	-112	-117
Modulation UL	16QAM							QPSK		
Modulation DL	QPSK									
Repetitions UL									16	
Repetitions DL								8		

Figure 3: Measurements on LTE-M using a nRF9160 DK at different distances from a base station.

Another DevZone blog post test the range of Bluetooth, ZigBee and Thread on the nRF52840DK. [6] Nodes for testing were placed in Line of Sight(LoS) from each other, 1 meter above ground, in an area with low 2.4GHz interference. The tests were performed for transmission powers of 0dBm, 4dBm and 8dBm. This showed that ZigBee has a somewhat lower range than Thread and Bluetooth. The results can be seen in Fig. 4.

Protocol / TX power output	0dBm	4dBm	8dBm
BLE @ 1Mbps	248m	276m	345m
BLE @ 2Mbps	238m	273m	333m
BLE @ 125kbps (long-range)	N/A	N/A	756m
ZigBee	196m	231m	280m
Thread	209m	270m	328m

Figure 4: Measuring distance form Bluetooth, ZigBee and Thread for Transmission output power levels.

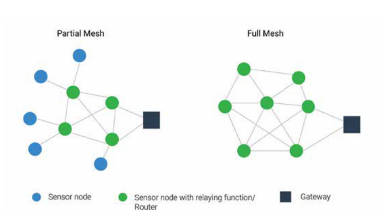
2.5 Behrtech: Power Mesh Networks vs. LPWAN

Behrtech is a IoT-technology focused company, which main product is the new Low Power Wide Area Networks (LPWAN) technology Mioty. [7] In the article *Power Mesh Networks vs. LPWAN: What You Need to Know*, Behrtech discuss differences between low power mesh networks and LPWAN. [8]

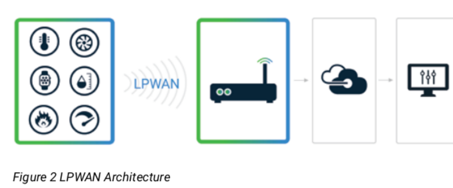
If a node in a mesh network is broken, the network will re-route the signal through the other nodes, making the mesh network more reliable.

Mesh networks typically use the unlicensed 2.4GHz band frequencies, which have a lot of traffic. This can make it harder for signals to arrive. The 2.4GHz signal is also poor at penetrating walls, which can decrease range in environments with many barriers. The article also points out that in fill mesh networks, nodes with a lot of traffic will use more power than can be ideal for low power applications. Mesh networks can be complex due to a possible need for manual setup of a custom system. Costs can also be high, as multiple routers may be needed if there is need for communication over large distances.

LPWAN signals use frequency ISM bands below 1 GHz. These signals travel longer and penetrate walls better. Due to the increased range, LPWAN often uses star topology, with a single base station which can reach all nodes. The lack of mesh infrastructure lowers complexity, allowing nodes to save more power. Fewer nodes also lower cost.



(a) Behrtech figure of mesh topologies. [8]



(b) Behrtech figure of LPWAN network. [8]

However, many LPWAN networks can not guarantee reliability. In addition, the lower wavelength lowers data rates. The article also remarks that LPWAN in the licensed frequencies has better reliability, but will have a higher cost and power consumption.

Behrtech concluded that mesh networks are more capable at medium range where high data rates are important. LPWAN on the other hand, will be more effective for long range use cases, where high latency can be tolerated.

2.6 Takeaways

Section 2.4 test max signal range for the mesh protocols. This can be used as an indicator for how far apart nodes can be placed apart when tests for this report are planned. Section 2.4 find max signal range for the mesh network to be around 300m. Section 2.1 on the other hand, used 100m between nodes. The takeaway from this is that testing in this project could be done for distances between 100m and 300m.

Section 2.4 also points out that the Cellular LPWAN modem will change transmission power automatically during transmission. Because of this, power consumption may vary based on distance from the cell tower. This will be important to keep in mind during testing of the Cellular protocols.

In Section 2.2, NB-IoT performs better than LTE-M, since LTE-M was missing support for eDRX and RAI. It is important to try to implement both cellular protocols with PSM, eDRX and RAI in this project. also point out big latencies during handovers. Because of this, the sensors for the tests of this project will remain stationary. It is important to remark on the observation from Section 2.2 that sometimes handovers occur during stationary communication in certain areas. It would be wise to choose a test location where this will not happen.

As Section 2.3 and Section 2.5 discuss mesh and cellular networks from a logistics perspective, it will not impact the testing. However, it will be useful insights for a conclusion of this project.

3 Theory

In this section, important technological concepts related to Cellular LPWAN and mesh networks will be explained.

3.1 The OSI model

The Open Systems Interconnection(OSI) model describes how network communication can be split into seven different layers. This can be seen visualized in Fig. 6. Seven different layers make up the OSI model: Application, Presentation, Session, Transport, Network, Data link and Physical. The Physical layer is the physical medium the signals travel through, such as wires or wireless senders. The Data link layer is how data is transferred between different devices in a local network, such as information packages and addresses. Medium Access Control (MAC) and Logical Link Control (LLC) is used for this layer. The Network layer is how devices can communicate over a global network, usually IPv4 or IPv6. Low-level communication between devices is covered in the Transport layer. This can be protocols such as UDP or TCP. In the Session layer, a temporary connection is made between devices communicating is created: A session. The presentation layer is how data is formatted, and often includes security, such as Transport Layer Security (TLS). The Application layer is where the application either the user or the machine is using is. This can for example be a web browser.

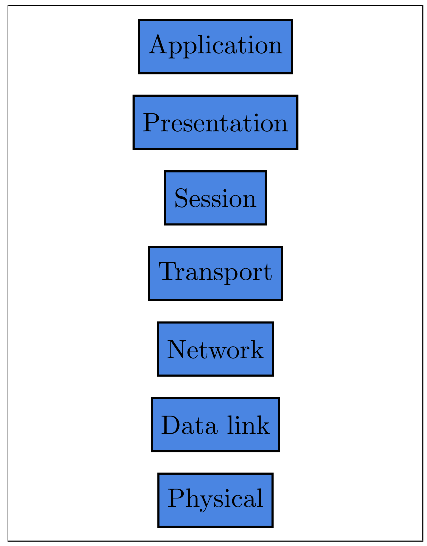
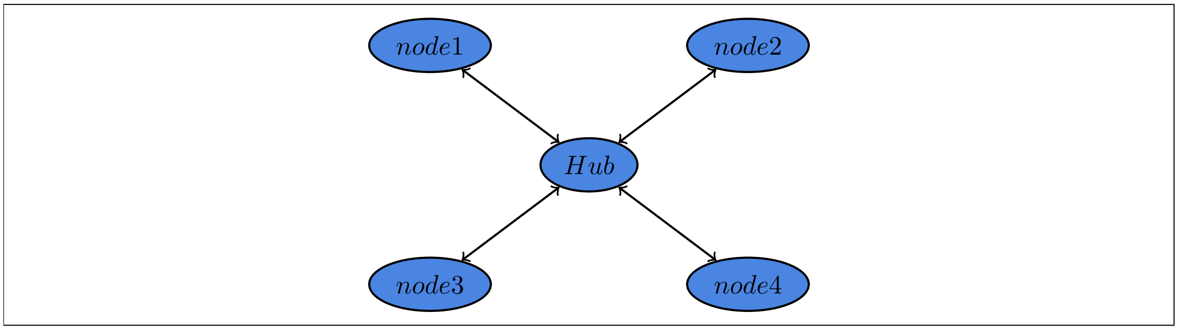


Figure 6: The Open Systems Interconnection(OSI) model, visualized.

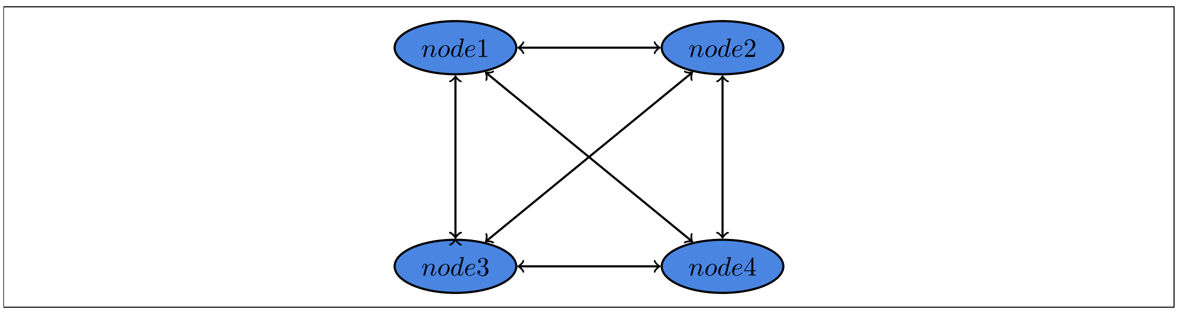
3.2 Wireless Network Topologies

There are many different network topologies. For modern wireless network protocols, star topology or mesh topology are normally used, as they scale well. [9] Star topology is when a single node, often called a hub, is connected to all the other nodes in the network. Edge nodes can not communicate directly with each other, and must instead communicate through the hub. Star topology is illustrated in Fig. 7a.

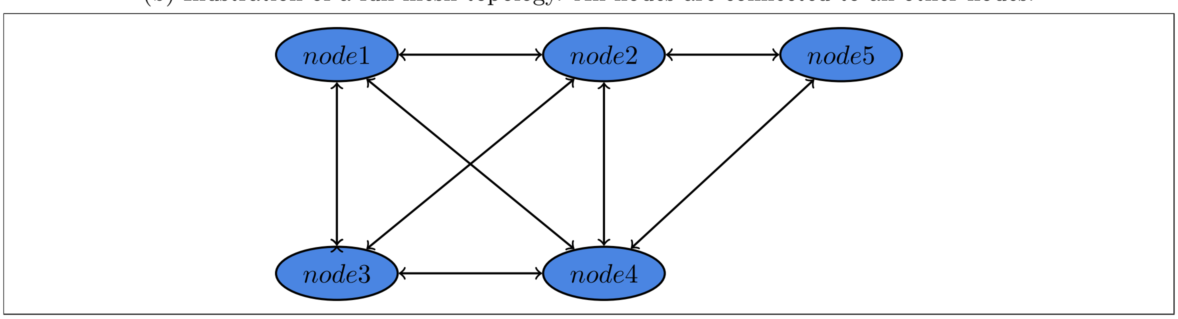
In mesh topology, nodes are interconnected: Any node can connect to any other node. Sending a message from one node to another in a mesh network will require the nodes to know what path the message shall take through the mesh network. Messages can be sent via multiple nodes from a sender node to a receiver node. Different methods are used for such routing, depending on what mesh protocol is used. Mesh networks can either be full or partial. A full mesh network is when all nodes in the mesh are connected to all other nodes, as illustrated in Fig. 7b. In a partial mesh network, nodes can be connected to multiple other nodes, but are not required to be connected to all other nodes, as illustrated in Fig. 7c.



(a) Illustration of star topology.



(b) Illustration of a full mesh topology. All nodes are connected to all other nodes.



(c) Illustration of a partial mesh topology. Nodes are connected to all nodes in range. Notice that node5 is only connected to node2 and node4.

Figure 7: Network topologies

3.3 ZigBee

ZigBee is a Wireless Personal Area Network (WPAN) specification. ZigBee use the 2.4GHz frequency band, and uses the IEEE 802.15.4 standard for physical and data link layers. According to the ZigBee alliance, the specification supports a data rate of 250 kbit/sec and range of 300m/100m line-of-sight/indoors. [10] It is important to state that this throughput is the theoretical maximum for the IEEE 802.15.4 physical layer. ZigBee adds overhead to communication sent, which will lower the throughput. ZigBee were standardized in 2003, and has since then been widely used, notably in home automation. The physical and data link layers of ZigBee use the IEEE 802.15.4 standard. ZigBee has its own implementation of the rest of the OSI model. Even the application layer is defined in ZigBee, such that all ZigBee devices should be interoperable. See Fig. 8a for an illustration of ZigBee set in the OSI model.

ZigBee networks consist of multiple nodes, connected as a partial mesh. There are three types of ZigBee nodes: End Devices, Routers and Coordinators. End Devices only connect to one other node in the network, and can not forward traffic through the network. End devices can enter sleep mode to save power. Routers are nodes that forward traffic through the mesh network. Routers can be connected to multiple other nodes. Routers can not enter sleep mode. A Coordinator is a router which in addition to forwarding traffic through the network, must also form the network. A ZigBee network must have one Coordinator.

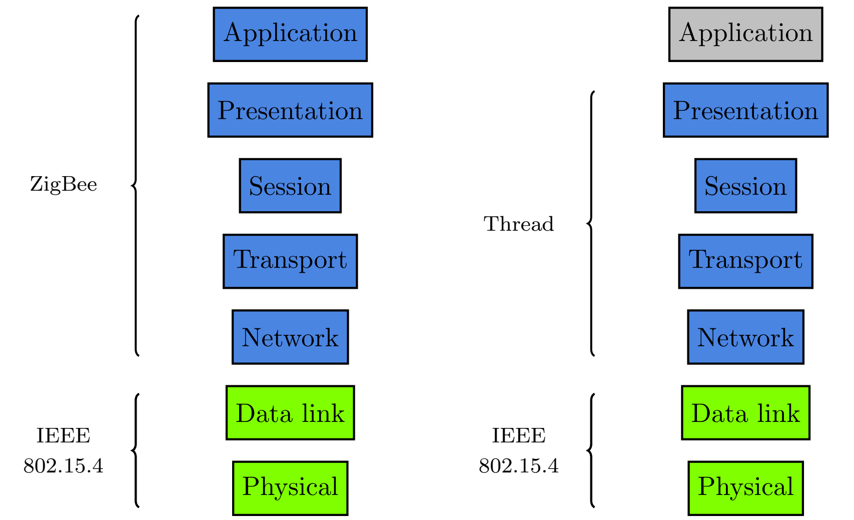
3.4 Thread and OpenThread

Thread is a WPAN specification, which uses the 2.4GHz frequency band and the IEEE 802.15.4 standard for physical and data link layers. For addressing, Thread uses IPv6 over Low-Power Wireless Personal Area Network, known as 6LoWPAN. As seen in Fig. 8b, the Thread protocol defines layers from network to presentation. However, thread does not define the application layer. It is up to the developer what application to use with thread, as long as the application can support 6LoWPAN. Typical applications on top of Thread can be UDP, CoAP or DHCPv6. The Thread protocol is from 2014, which is new for a communication protocol. Thread has seen a lot of use, and companies such as Apple, Google, NXP and Nordic Semiconductor are members of the Thread Group. [11]

Thread devices can have two roles, either End Device or Router. End Devices are only connected to a single router, and can not forward traffic through the network. Thread End Devices can disable the transceiver, to save power. Thread Routers forward traffic through the mesh network. Routers can be connected to multiple other devices. The transceiver of Thread Routers can not be disabled.

The Thread network can be connected to the internet if one of the routers is a Thread Border Router(BR). The Border Router will perform routing and DNS services to allow nodes to access the internet and vice versa.

OpenThread is an open source implementation of Thread by Google. "OpenThread implements all features defined in the Thread 1.1.1 Specification." [12]. This project will use OpenThread.



(a) OSI model for ZigBee (b) OSI model for Thread

Figure 8: The Open Systems Interconnection(OSI) model, visualized.

3.5 LTE-M and NB-IoT

LTE-M and NB-IoT are cellular Low Power Wide Area Network (LPWAN) protocols. LTE-M is short for Long Term Evolution Machine Type Communication (LTE-MTC(LTE-M)), and NB-IoT short is for Narrow Band Internet of Things (NB-IoT).

LTE-M and NB-IoT are both part of the Long Term Evolution (LTE) standard, also known as 4G. LTE devices connect to base stations in star topology. Base stations are operated by cellular operators: Companies who run the network, and charge fees to use the LTE network. Different cellular operators use different frequency bands for the LTE network. A list of frequency bands can be seen at [13]. Cellular operators often have multiple base stations in the same are. In this case, the LTE devices will typically connect to the base station with the strongest signal. Then the cellular device will send authentication data provided by a SIM, and will be connected. This process is called *Attach*.

The LTE network have mostly been for sending large amounts of data, for use with mobile phones. LTE-M and NB-IoT are recent adds to the LTE network that aim for low data rate but longer range and less power consumption. The biggest difference between these two is that LTE-M has higher throughput while NB-IoT has longer range. LTE-M has download/upload rates of 300/375 kbps and a range of up to 11km, while NB-IoT has download/upload rates of 30/60 kbps and range of up to 15km.

As LTE-M and NB-IoT are intended for low-data applications, sleep mode is important. For LTE, this is done by Connection Modes. There are three different Connection Modes: RRC Connected, RRC Idle and PSM.

In Radio Resource Control (RRC) Connected, the cellular device can transfer data, but will have high power consumption.

The device can enter RRC Idle to sleep for shorter intervals between communicating, to save power. This will result in latency for download, as messages will first arrive after the sleep interval. LTE-M sleep interval can be up to 44min, while NB-IoT can be up to 175min. RRC Idle is also known as extended Discontinuous Receive (eDRX).

For Power Saving Mode (PSM), the cellular device can sleep for much longer, up to 413 days. There is however a minimum sleep time of 10min. After the PSM sleep, the device must go to RRC Connected mode to receive and/or send data.

For both eDRX and PSM, the cellular device can wake up at any time to send data. For PSM however, it is a lot more power efficient to also send in RRC Connected between PSM sleep, since the device is already in the RRC Connected state. eDRX use less power to go from sleep mode to receive mode then PSM, while PSM can sleep for longer intervals.

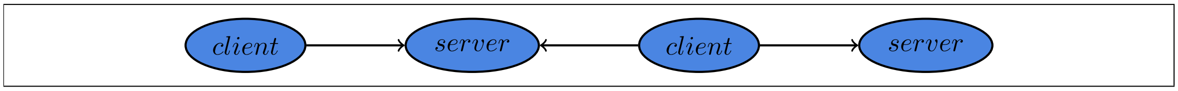
More information on the difference between LTE-M and NB-IoT and Connection Modes can be found in this Nordic Webinar [14].

For eDRX and PSM, the cellular device configure its sleep intervals, whereas the base stations that are responsible for the timing. In RRC Connected Mode, a cellular device will first reenter after a preconfigured timeout period, without any activity. If the cellular device does not expect to receive data, and has finished sending data, it is power inefficient to still stay in RRC Connected mode until the timeout expires. Using the Release assistance indication (RAI), a cellular device can signal the base station that it is ready to reenter PSM sleep mode. [15]

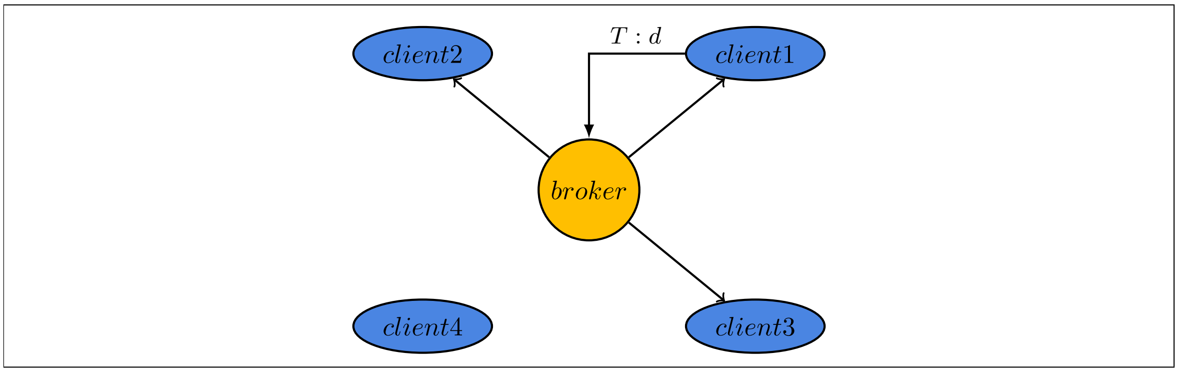
3.6 CoAP vs MQTT

The Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport (MQTT) are lightweight Application layer protocols for Machine to Machine (M2M) communication. CoAP use server-client communication, which is practical for one to one connections. In server-client communication, nodes communicate in pairs, between server and client. It is possible, however, for a client to communicate with multiple different servers, and vice versa. See Fig. 9a for an illustration of how CoAP nodes communicate. CoAP use the same addressing as HTTP, where the server is connected to an IP address, and has resources open on this address. The client can then access there resources by using defined methods, such as GET or PUT. For example, an server may be connected to the IP address 123.456.654.321, with the resource "time". The client can then send a GET request to 123.456.654.321/time, and the server will send data containing the time to the client.

MQTT use publish-subscribe communication, which is practical for many to many communication. Here, a server called a *broker* will host *topics*, that clients can subscribe or publish to. When a client publishes a message, the broker will forward the message to all clients who have subscribed to the topic. All clients need to know the address of the broker for this to work. For example, a broker host the topic "time". Then client2 and client3 subscribe to "time". When client1 then publishes data containing the time to the topic "time", both client2 and client3 will receive this message. The MQTT protocol is illustrated in Fig. 9b.



(a) CoAP network structure. Clients make a connection to the server. In the same connection, data can be returned.



(b) MQTT network structure. Client1 publish a message with topic T containing data d to the broker. The broker then forward the message to clients that subscribe to T: 1,2 and 3.

Figure 9: CoAP and MQTT network topologies

3.7 nRF development

There are three main steps for running code on a microcontrollers: Writing the code, compiling the program and flashing the microcontrollers. Nordic Semiconductor provide Software Development Kits (SDK) to help with all three of these steps. A SDK provides readymade functions for different uses, such as drivers or libraries. In addition, finished examples can be found, to jump-start development. For developing for the nRF development kits used in this thesis, there are two different alternatives for SDK.

First, the new Nordic Connect SDK(NCS) [16] is built around the Zephyr Real Time Operating System (Zephyr RTOS). To start development with NCS, it is recommended to use the nRF Connect for Desktop. [17] NCS is the newest SDK, and was therefore the most used for this thesis.

Second, the nRF5 SDK [18] is older, but has more functionality, as not all have been moved to NCS yet. To get started with development on the nRF5 SDK, see its product page [18]. The nRF5 SDK was only used in this thesis where functionality was not found in NCS, for example for the mesh Benchmarking test.

Both these SDK also provide solutions for compiling code and flashing microcontrollers.

4 Overview of test concept and scenario

4.1 Objective

A test system will be design to measure the four properties, for either mesh or cellular LPWAN networks: Current consumption, Latency, Throughput and Reliability.

Throughput is a measure of the amount of data a device is able to send over a time period, and is measured in kilobits per second ($kbps$). Latency is the amount of time a message uses when traveling from one device to another. This may be measured either one-way or for a round-trip, and is measured in milliseconds(ms). Power consumption is how much power a device use to operate, and is a function of voltage and current. The nRF52840DK and nRF9160DK used in this thesis, run at an internal voltage of 3V. Therefore, current measurement will be done to get the power consumption. Current consumption is a measure on how much current a device uses, and is measured in either micro- or milliampere ($\mu A/mA$). Reliability is a measure of how much of the data is lost, and is measured in the percentage of messages arrived successfully.

4.2 Testing scenario

The testing will be performed in a realistic environment. A use case is selected where it is not clear which wireless technology would be better. The task will be to send measurements from battery powered dummy sensors to a cloud server. There are four such sensors, and the range in between each are defined as D, as seen in Fig. 10

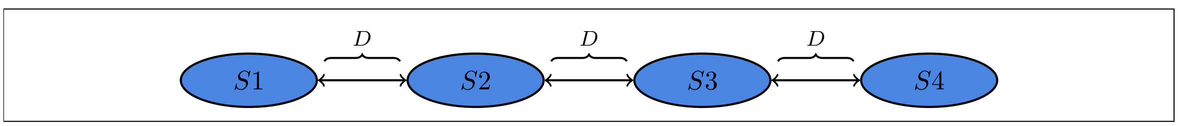


Figure 10: Four sensors, distibuted over a horizontal line, where the distance between each sensor is D, and the distance between two outer sensors is $3D$.

For cellular communication, each of the sensors communicate directly through the internet, to a server. The server will log the data reported by the sensors. This is illustrated in Fig. 11

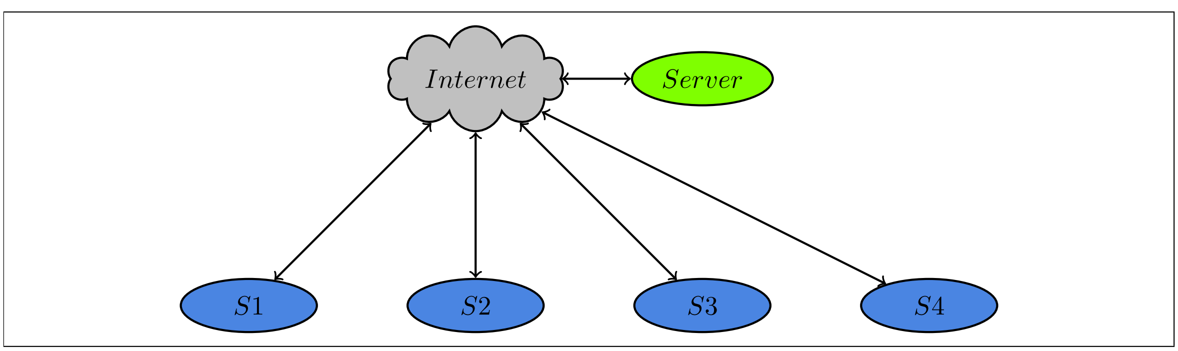


Figure 11: The four sensors, connected directly to the internet. The server is also connected to the internet.

In a mesh network, the devices are interconnected, and get internet access from a border router connected to one of the nodes. The sensors will be placed a distance D apart, such that nodes only connect to neighboring sensors, so that all will be connected in series. This connection can be seen in Fig. 12. Traffic from all sensors have to travel through S1. S2 will forward data from S3 and S4. S3 will forward data from S4. The edge node S4 will only send traffic. This way, tests can be done for different traffic conditions.

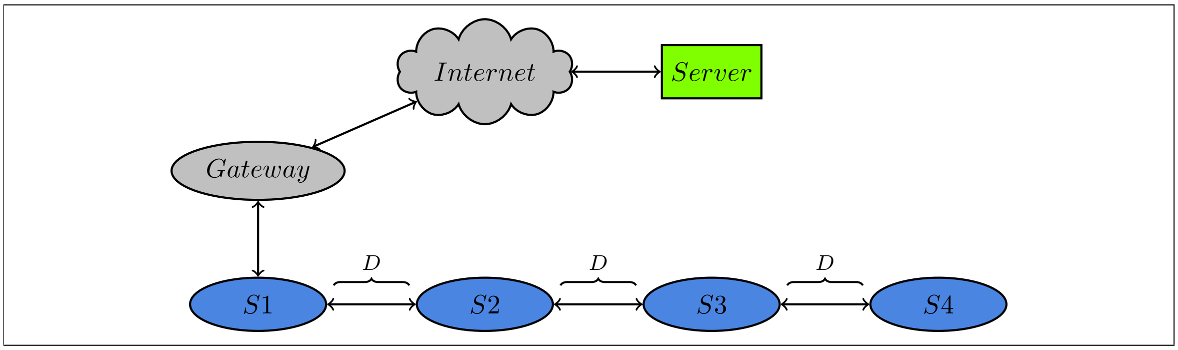


Figure 12: The sensors are communicating wirelessly with each other. Sensor 1 is connected to a border router to get internet access, with one sensor connected to it on one side, and two in series to the other.

To measure on nodes that only forward messages, relays are used to forward the signal, as seen in Fig. 13. The distance D will span between a node and a relay, such that all are still connected in series.

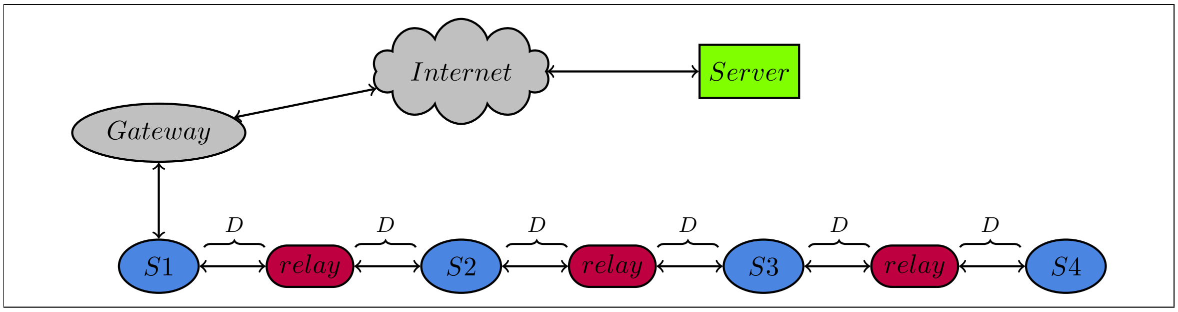


Figure 13: A mesh network, with relays to enable a signal to travel greater distances.

To measure results from the tests, a measurement module will be connected to all nodes in the network. An example of this in the mesh network can be seen in Fig. 14. In a cellular network, measurement nodes would be connected to sensors S1, S2, S3 and S4.

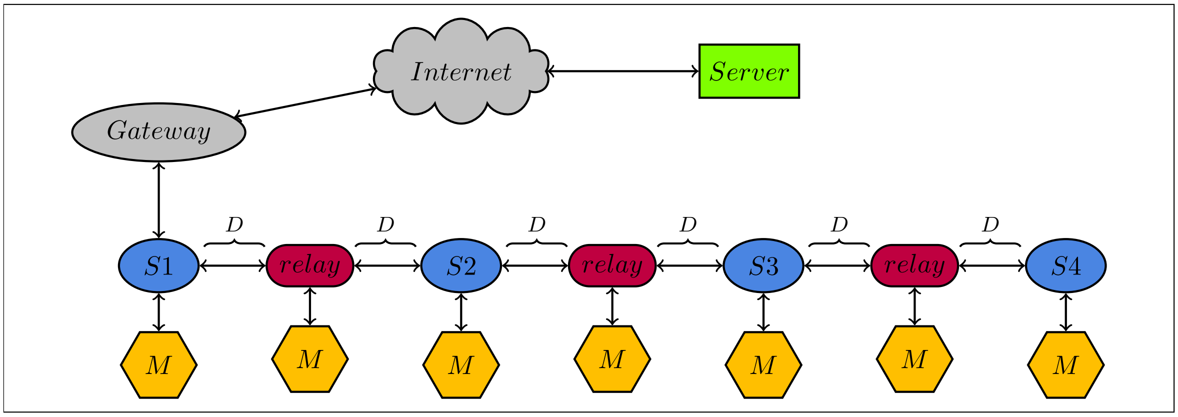


Figure 14: Measurment modules M connected to a mesh network.

4.3 Mesh networks

The tests for mesh networks will first be done with nodes S1, S2, S3 and S4. Then another test will be performed with relays between each sensor in the series. Relays are named after nodes they span, R1-2, R2-3 and R3-4. The use of relays in a mesh network is illustrated in Fig. 13.

These tests will be performed using first ZigBee mesh, then the OpenThread.

4.4 Cellular communication

After the tests for mesh networks have been performed, similar tests will be performed on the cellular communication protocols LTE-M and NB-IoT. These tests will be done similarly to the mesh tests, with a few differences.

As the cellular devices communicate directly to a server, the range between them will not influence the testing. To test the nodes for similar environmental noise, the cellular nodes will be placed in the same spots as the mesh nodes.

4.5 Introducing Interference

Other wireless signals may interfere with the communication of the test nodes. An interference module should be added to intervals of the tests, to find how interference will affect the results.

5 Physical Implementation

In this section, The test system described in Section 4 will be implemented.

5.1 Gitlab Repository

A public Gitlab repository containing code that was used in this project can be found at [19].

5.2 Equipment types

The tests will use different kinds of hardware, listed in this section. In the PDF version, the section headers below will act as links to the product web pages.

5.2.1 nRF52840DK

The nRF52840 Development Kit(DK) is a DK for the Nordic Semiconductor nRF52840 system on a chip (SoC). It supports Bluetooth Low Energy, Bluetooth mesh, Thread, ZigBee, 802.15.4, and ANT. [20] Development on this kit will be done using both nRF5 Software Development Kit (SDK) for Thread and ZigBee version 4.1.0 [21] and nRF Connect SDK (NCS) version 1.4.2. [16]

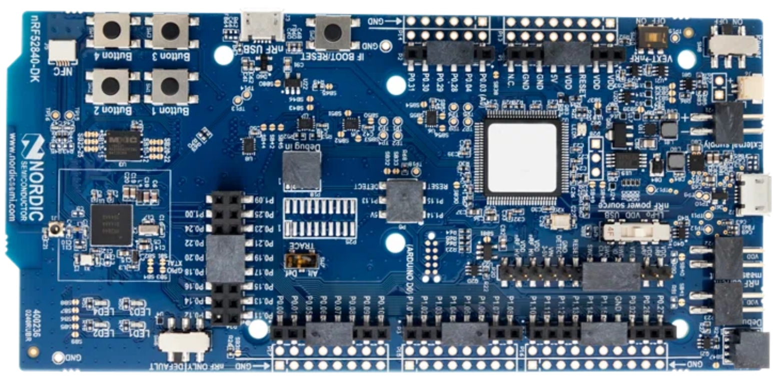


Figure 15: The nRF52840 DK

5.2.2 nRF9160DK

The nRF9160 DK is a DK for the Nordic Semiconductor nRF9160 SoC. It supports LTE-M, NB-IoT and GPS. [22] Development on this kit will be done using NCS version 1.4.2. [16]

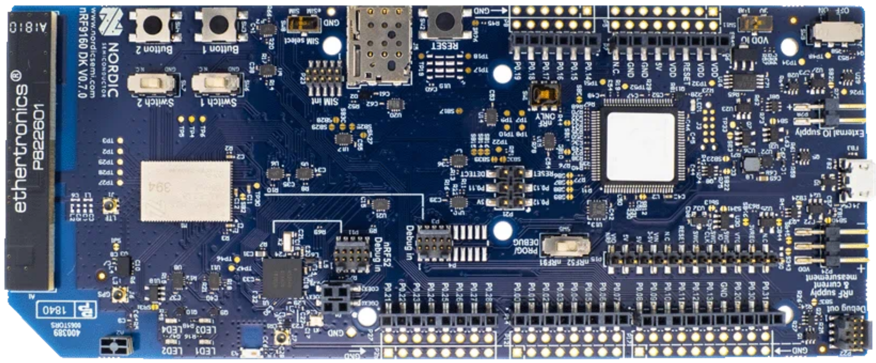


Figure 16: The nRF9160 DK

5.2.3 Power Profiler Kit II

The Power Profiler Kit II (PPK2) is a Nordic Semiconductor standalone kit for measuring and supplying currents for the Development Kits of Nordic Semiconductor. [23] The PPK2 will be interfaced using nRF Connect for desktop [17] and an Python API by IRNAS [24]

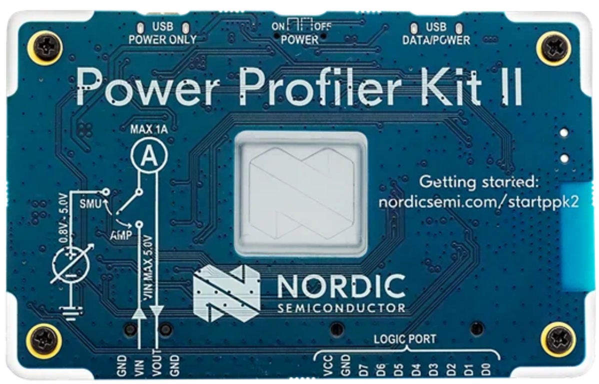


Figure 17: The Nordic Semiconductor Power Profiler Kit 2

5.2.4 Raspberry Pi 3B+

The Raspberry Pi 3B+ is a single board computer featuring 1.4GHz 64-bit quad-core processor, dual-band wireless LAN and Bluetooth 4.2/BLE among other things. [25] The Linux operating systems Raspbian [26] and Ubuntu for Raspberry Pi. [27]

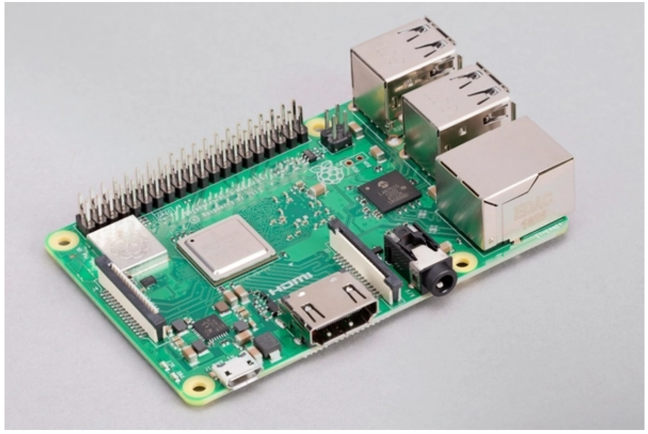


Figure 18: The Raspberry Pi 3B+

5.3 Description of test nodes

The implementation of the different nodes from Fig. 14 will be described in the following sections. The Nodes are: sensor nodes(S1,S2, S3 and S4), relay nodes, a border router, measurement modules and a web server. The hardware of each module, except the virtual cloud server, can be seen in Fig. 19.

5.3.1 Sensor Nodes

Sensor nodes will be part of both the mesh test and the cellular test. The job of this node is to send dummy data to the web server, either over the mesh or the cellular network. This node will consist of a nRF52840 DK for mesh, or a nRF9160 DK for cellular. The kits will run edited versions of the *Thread CoAP client* [28], *ZigBee Light Switch* [29] and *nRF9160: nRF CoAP Client* [30] samples for the different tests. Hardware listed in Fig. 19a.

Sensor nodes will act as Thread Routers or ZigBee routers.

5.3.2 Relay Nodes

Relay nodes will be part of the mesh test. These nodes will relay communication between sensor nodes when distances are too long for the radio signals to travel in a single hop. This node will consist of a nRF52840 DK using a modified version of the NCS *Thread: CLI sample* [31] or the NCS *ZigBee: Light bulb* sample. Hardware listed in Fig. 19b.

Relay nodes will act as Thread Routers or ZigBee routers.

5.3.3 Cloud Gateway

The cloud gateway will be part of the mesh test. It will connect the mesh network to the internet. A Raspberry Pi 3B+ connected to a nRF52840 DK will be used for this. For the Thread test, the Raspberry Pi will be configured as a Thread Border Router [32], with the nRF52 DK running the NCS *Thread: NCP* sample. [33] For the ZigBee test, the nRF52 DK will run the NCS *ZigBee: NCP* sample [34], while the Raspberry Pi will be configured as a *NCP Host for ZigBee* [34], with added CoAP functionality. For the ZigBee network, an additional nRF52840 DK will be added to this node, to act as a *ZigBee: Network Coordinator*. [35] The cloud gateway must be connected to the internet to function. For the Thread Border Router, the internet connection specifically must be a wireless network which can be connected to using only SSID and password. This can for example not be Eduroam. Hardware listed in Fig. 19c.

The cloud gateway will act as a Border Router for Thread, and a Coordinator + custom cloud gateway for ZigBee.

5.3.4 Cloud Server

The cloud server will be part of both the mesh test and the cellular test. It will log data from the sensors and be implemented as a CoAP server run on a Linux Virtual Machine(VM). The VM will run Debian [36], with a CoAP server built on libcoap [37]. The CoAP server will host a ”log” resource, which can be written to using the PUT method. When a node writes to this resource, the server will log the data received, and return the string ”Received”. The data from the nodes are expected on the format ”ID,data”, and the server will save ”timestamp,ID,data” to a Comma-Separated Values(CSV) file. The server will be associated with a static public IPv4 address. This address will be referred to as SERVER_IP. More info on CoAP can be found in Section 3.6.

5.3.5 Measurement module

To measure and log the results of the tests, an external module will be used. This module consists of the Raspberry Pi 3B+ and the Power Profiler Kit II (PPK2). The PPK2 will supply power to and measure current from nRF Development kits. The Raspberry Pi will log data from the PPK2. In addition, the Raspberry Pi will be connected to the nodes by wire, to be able to control or reprogram the Development Kits and collect information. Code running on the Raspberry Pi can be seen in [19]. A Deltaco PV-1050 Power Bank [38] will be connected to the Measurement module, to power all of the test. This Power Bank is able to supply 5V 2.1A. Hardware listed in Fig. 19d.

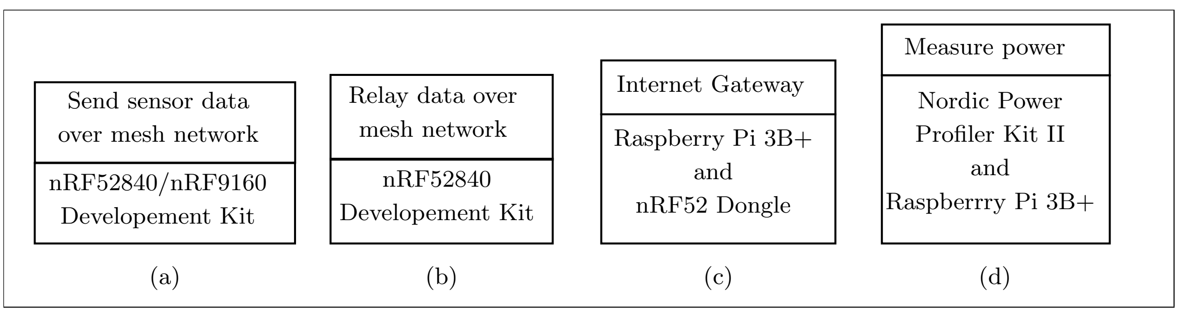


Figure 19: Nodes for the test network, and what hardware they will use.

5.4 Required amount of equipment for performing tests

The equipment needed for the different tests is listed in the following sections. The hardware needed per module is listed in Table 4. The cloud server is not listed as any specific hardware, as the Linux machine could be any computer with an internet connection.

Node	nRF52840DK	nRF9160DK	PPK2	rPi	Battery
Sensor(cellular)		1			
Sensor(mesh)	1				
Relay	1				
Cloud Gateway	2			1	
Cloud Server					
Measurement module			1	1	1

Table 4: Equipment needed per module. Hardware described in Section 5.2.

5.4.1 Equipment for Cellular Test

For the cellular tests, four sensor nodes and one cloud sensor is needed, as seen in Fig. 11. A measurement module will be connected to each of the four sensor nodes. This adds up to the hardware seen in Table 4.

Hardware	Amount
nRF9160DK	4
Raspberry Pi 3B+	4
PPK2	4
Battery	4

Table 5: Total hardware needed for the cellular tests.

5.4.2 Equipment for Mesh Test

The same equipment is used for both the Thread and the ZigBee test. As seen in Fig. 13, four sensors are needed. Three relays will be used, in between each node. A single cloud gateway will be used for the whole network. A measurement module will be connected to each node, except for the cloud gateway, totaling seven measurement modules. The sum of hardware needed for this test can be seen in Table 6.

Hardware	Amount
nRF52840DK	9
Raspberry Pi 3B+	8
PPK2	7
Battery	7

Table 6: Total hardware needed for the mesh tests.

5.5 Measurement of properties

5.5.1 Current Consumption Measurement

The PPK2 of the measurement module will be connected to the nRF52840DK/nRF9160DK nodes during all tests, and will measure current consumption continuously. The measurement function will return a current measurement every two seconds. This measurement will be the average of measurements done in the two seconds.

The PPK2 has a sampling rate of 100 samples per millisecond. The measure function in the PPK2 API returns 1024 samples. This way, measurements are done in 10 milliseconds. As tests are done over a long time, an average will be done over two seconds. The data read function will therefore be run 200 times, before calculating an average over 204800 samples. This average will be logged.

5.5.2 Reliability Test

In order to measure reliability, the sensors will periodically send dummy data to the cloud server. Measurements will be sent once every second. GPIO pins P0.07 and P0.08 of the nodes will be connected to the measurement module. Every time the node sends or receives a message, it will signal the measurement module using these pins. Errors in transmission will also be signaled using this method. While UART could have been used to communicate transmission status, GPIO is used to consume as little power as possible. The amount of unsuccessful transmission is often referred to as a *loss rate*. The term loss rate will sometimes be used as a substitute for the term reliability, as it is shorter.

P0.07	**P0.08**	**State**
0	0	Idle
1	0	Message received successfully
0	1	Message sent successfully
1	1	Transmission error

Table 7: States signaled from nodes to measurement module.

5.5.3 Send Test

To measure current consumption during realistic circumstances, a simple send test is performed. In this test, Sensors will send data to the Server every ten(10) seconds. Current will be logged during this test. For the mesh tests, nodes will be configured as routers.

5.5.4 Throughput and Latency Test

The Benchmark Application Example from the nRF5 SDK [39] is a test that will measure both throughput and latency. The test takes approximately two minutes to complete. During this time, the test will send 1000 messages at the size of 64 bytes, which is the total of 64 000 bytes. Two nodes are required for the test, one tester and one receiver. Sensor 1 will be the receiver, and sensors 2-4 will be testers. The test will be configured to *echo* mode: The tester sends data to a receiver, who will echo data back to the tester. By timing the round trip time for each packet sent, latency can be measured. Both average latency over all packets sent and the max latency will be logged. By timing the total time of sending data, throughput can be measured. The Throughput and Latency test will be referred to as the *benchmarking* test,

The sensor nodes will perform the benchmarking tests. The nodes can perform the benchmarking test at the same time, or time multiplexed. Both will be performed. Benchmarking tests will be started every 15 minutes. Each node will first perform a benchmarking test simultaneously. After this, each node will wait minutes equal to its node number times two. Node 1 will wait 2 minutes, node 2 will wait 4 minutes, and so on. The second test will be done by each node separately.

5.5.5 Introducing interference

Interference will be introduced during part of the tests, as can be seen in Section 5.6 A separate radio sender will be placed next to sensor node S2 and broadcast every second. A radio receiver will be placed next to this sender, to receive the messages. Both interference nodes will be nRF52840DK.

5.6 Test timeline

The test will be performed over three hours. Once for the Send test, and once for the Benchmarking test. After two hours, the interference node, mentioned in Section 5.5.5 will activate. Current measurement will run for the full tests. The timeline for tests can be seen in Fig. 20.

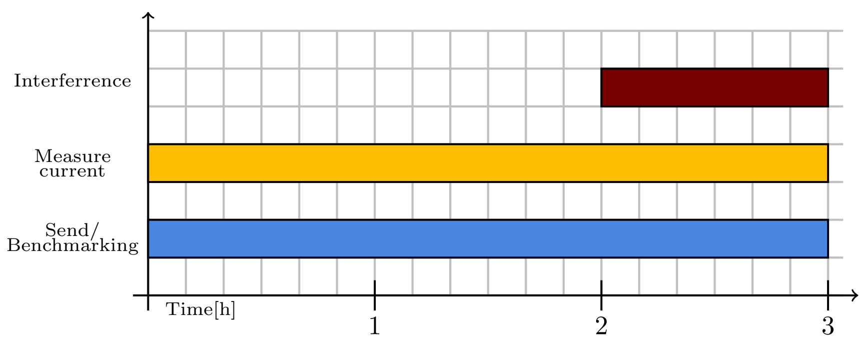


Figure 20: The timeline for the tests. The X axis show time in hours. The Y axis list tests, measurments and interferrence.

5.7 Test Variables and Output Format

The tests will be performed over time, with measurements made periodically, resulting in output in the form of time series. Multiple properties will be measured during each test. Tests will be done for four different protocols, Thread, ZigBee, LTE-M and NB-IoT. First, send tests and benchmarking tests will be performed for four nodes using all four protocols. Then the same tests will be performed for 7 nodes using mesh protocols. The tests can be seen in Table 8.

	Thread	**ZigBee**	**LTE-M**	**NB-IoT**
4Send				
7Send			X	X
4Bench				
7Bench			X	X

Table 8: Every test, as a table. The X indicates that no test are done for this combination. 12 Tests will be performed in total.

Test results will be logged in CSV files. These files will be named and sorted so that they can be located easily. Firstly, they will be sorted after test dates, then after type of test, and then named by the test variables. The folder structure with file names is shown in Appendix A.

5.8 Distribution of test nodes

In a mesh network, nodes will interconnect if possible. To achieve connection of nodes in series, as described in Section 4.2, the nodes will be located such that they can not physically make unintended connections. This is done by placing neighboring nodes just inside each others range. For example, S2 can be located at max range D from S1 on one side and max range D from S3 on the other side. This way, the distance between S1 and S3 is 2D, and these two nodes can not connect to each other. Signals from S1 to S3 will then have to travel via S2.

To verify that this placement is correct, the nRF Thread Topology Monitor(TTM) is used. [40]

For send tests, the mesh networks will be connected to a Cloud Router to obtain internet access, as mentioned in Section 5.3.3.

Cellular nodes connect to the best cell tower. The test nodes will be placed in an area with cellular coverage. Interconnection will not be a problem for cellular nodes. However, the cellular nodes will be placed in the same locations as for the mesh nodes, to test under similar environmental conditions. More specifically, Cellular test nodes will be placed as for the Send4 test.

5.9 Environment protection

To protect the nodes and measurement modules from hostile environment effects such as rain or wind, all electronics will be placed inside plastic boxes. As the test nodes will be located in the public domain, a note will be attached to each box, informing passerby people of the purpose of the nodes. The note will also contain contact information.

5.10 Original and revised tests

As found in Section 2.4, the nRF52840DK is able to send messages up to 300m. A linear mesh network with 7 sensors would cover 2100m. Due to the limited size of buildings, tests were originally performed outside. During the first outside test, it was found that the range of the test nodes was approximately 70m. There were probably two main reasons to why this range was much lower than found in Section 2.4. The test from Section 2.4 will be called the *range* test, while the one performed in this project will be called the *original* test.

Firstly, the range test were performed with Line of Sight(LOS), placed at 1 meter above ground. In the original test, nodes were placed with LOS, but inside plastic boxes. In addition, nodes were placed in somewhat hidden locations, and therefore not in ideal locations for wireless signals. Secondly, the range test were performed in an area with little wireless interference. The original test were located on a busy campus, with much wireless interference.

The actual range between the nodes is not too important though, as the main goal is to connect nodes linearly, not test range.

During the first original test, a middle aged woman was found trying to move one of the boxes. She did not want it to be stolen, as the contents looked valuable. In the dark she had not seen the note on the box, explaining its purpose. Due to this incident, the boxes were painted black, to draw less attention.

During the second original test, one of the boxes was vandalized in the middle of the day, at 12:37:05. When retrieved, the contents of test node 4 were found scattered on the ground, with the battery missing. As it was raining this day, equipment as part of node 4, Raspberry Pi, nRF52840DK and PPK2 were dried using a heat-gun. All equipment from this node were functional after drying.

Due to this, testing conditions were revised, and tests were moved inside a building. It was discovered that inside, node range was approximately 30m, allowing all seven nodes inside a single building while still linearly connected. These tests will be referred to as *revised* tests.

6 Results

Table 9 shows the success of the tests done. Due to time and hardware restrictions, tests could not be rerun multiple times. Therefore, not all tests were complete. Results were collected from incomplete tests, for example if only node 1 collected measurements, these measurements will be considered. Results can be found in the git repository [19], and the folder structure with filenames can be seen on Appendix A

	Thread	**ZigBee**	**LTE-M**	**NB-IoT**
4Send	All	All	All - 3	All - 3
7Send	Broken	Broken		
4Bench	All - 3+4(mA)	Only 2	Only 2	None
7Bench	All	Broken		

Table 9: A table listing how successful each test were. "All" is short for "All Nodes". "All -2" would indicate that all nodes except Node 2 was successful. "None" indicates that the test was not performed. "Broken" indicates that the test was unsuccessful.

6.1 Bugs and other Considerations

6.1.1 Timestamps

All tests were logged by the measurement module, as explained in Section 5.3.5. Measurements was timestamped by the measurement module. During the first tests, the time of the measurement module was initialized by connecting a Real-Time Clock(RTC) module to the Raspberry Pi at startup, then remove it after, and repeat with all measurement modules. This method was not stable, and for multiple tests the timestamps will not be synchronized for the nodes. In revised tests, a permanent RTC module was connected to node 1, 2 and 4. Node 3 will use wrong time, and therefore often not be logged in these tests. The measurement modules for router nodes did not have RTC modules, and will not yield synchronized measurements.

In addition to this, node 4 was for multiple tests set to a timezone one hour in front of the other tests.

Timing problems will be commented on for each result where relevant.

6.1.2 PPK2 bug

Multiple times during current measurements, the Power Profiler Kit 2 measured currents several magnitudes higher than the expected values. This was impossible, as the battery supplying the measurement module could not supply such currents. When the measurement module was restarted, the bug would disappear. This bug manifested with irregular intervals. The reason for the bug was not discovered. Measurements affected by this bug were discarded.

6.1.3 Raspberry programmer bug

The nRF development kits were sometimes programmed by a raspberry pi, as explained in Section 5.3.5. Sporadically this would corrupt the firmware running on the nRF52840DK, when reprogramming of the benchmarking sample. The benchmarking test program would be functional, but neither the Thread nor the ZigBee protocol were functional. Tests affected by this bugs produced no measurements, and were discarded.

6.1.4 ZigBee connections

For ZigBee nodes to join the ZigBee network, nodes had to be started in a specific order. First the coordinator had to be started, then each other node. For tests using the ZigBee to CoAP bridge, the bridge had to be started last. Multiple times nodes did not connect to the ZigBee network even with this startup order. The reason for this ZigBee behavior was not found. For the send tests, connection to the ZigBee network could be verified from the ZigBee to CoAP bridge.

For the benchmarking tests, connection to the ZigBee network could be verified from the benchmarking target node. Nodes would seemingly disconnect from the ZigBee network after approximately 5 minutes. The benchmarking test were to run every 15 minutes. For the ZigBee network to connect anew for every benchmarking test, all nodes were automatically reset before each ZigBee benchmarking test.

6.1.5 Developing costs

Wireless communication protocols are complicated, and developing tests from scratch would have required a lot of work and time. To lighten the workload, the test firmware in this project is mostly modified examples from the NCS, as mentioned in Section 3.7. With the examples, most of the protocols were more or less implemented with no problems.

In general, the cellular protocols required less setup, as interconnection did not have to be handled. In addition, no relays were needed for cellular protocols, and less nodes were easier to handle.

ZigBee, however, required much more time and work. In the end, ZigBee was the only protocol which were not fully functional, as seen in Section 6.1.4. ZigBee development costs were considered higher than for the other protocols.

6.2 Thread test results

All Thread tests were performed using revised conditions. Current consumption logged in the Thread send test with 4 nodes can be seen in Fig. 21.

Fig. 22 shows current consumption from sensor 4, during send test with four nodes. The figure shows a sample size of five minutes, from 16:00 to 16:05.

Current consumption logged in Thread benchmarking with 4 nodes can be seen in Fig. 23. Measurements for node 3 and 4 were discarded due to the PPK2 bug, as mentioned in Section 6.1.2.

Throughput, average latency and max latency during benchmarking with 4 nodes, for nodes 1, 2, 3 and 4 can be seen in Fig. 24, Fig. 25, Fig. 26 and Fig. 27 respectively. Remark that the axis of the graphs are not scaled equally. The time of node 4 is 1 hour offset due to being registered for a different timezone.

Current consumption logged in Thread benchmarking with 7 nodes, for sensor nodes 1, 2, 3 and 4 can be seen in Fig. 28. Node 3 is out of time synchronization due to an unsuccessful RTC connection at boot.

Current consumption logged in Thread benchmarking with 7 nodes, for router nodes 1-2, 2-3 and 3-4, can be seen in Fig. 29. All three nodes are out of time synchronization due to an unsuccessful RTC connection at boot.

Throughput, average latency and max latency during Thread benchmarking with 7 nodes, for nodes 1 and 3 can be seen in Fig. 30, and Fig. 31 respectively. Remark that the axis of the graphs are not scaled equally. Node 3 is out of time synchronization due to unsuccessful RTC connection at boot.

Send test Thread, 4 nodes

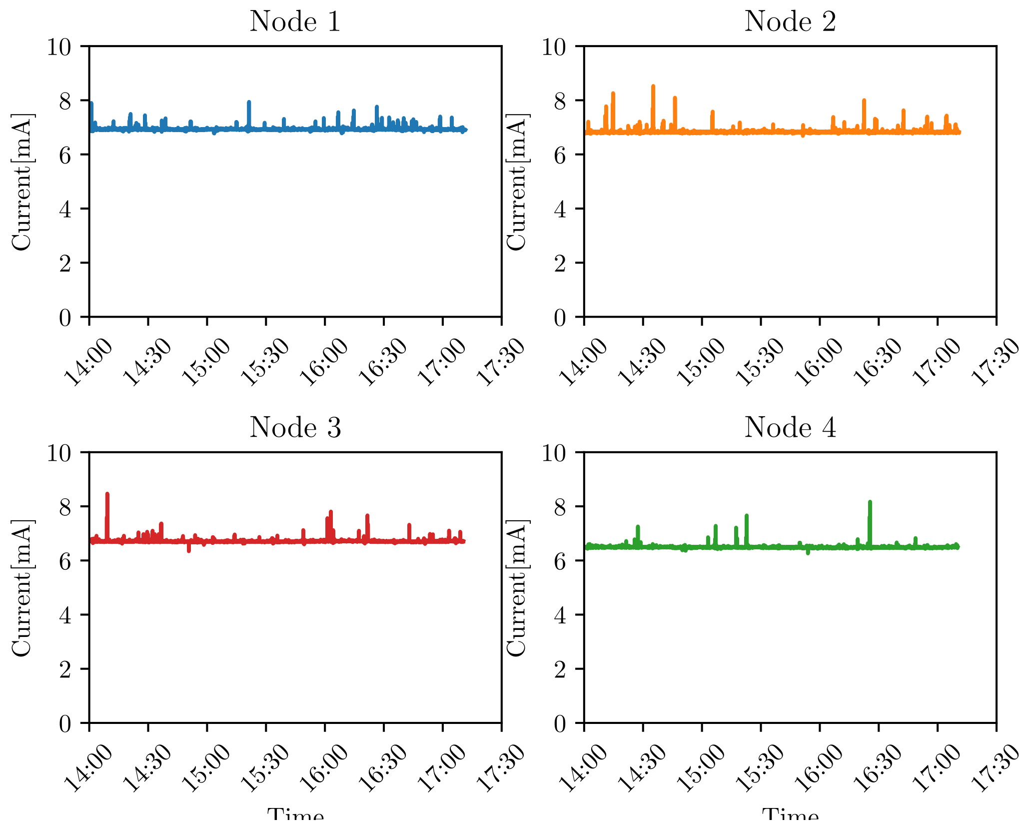


Figure 21: Current consumption for nodes during Thread send test under revised condition, using 4 nodes.

Send test Thread, 4 nodes

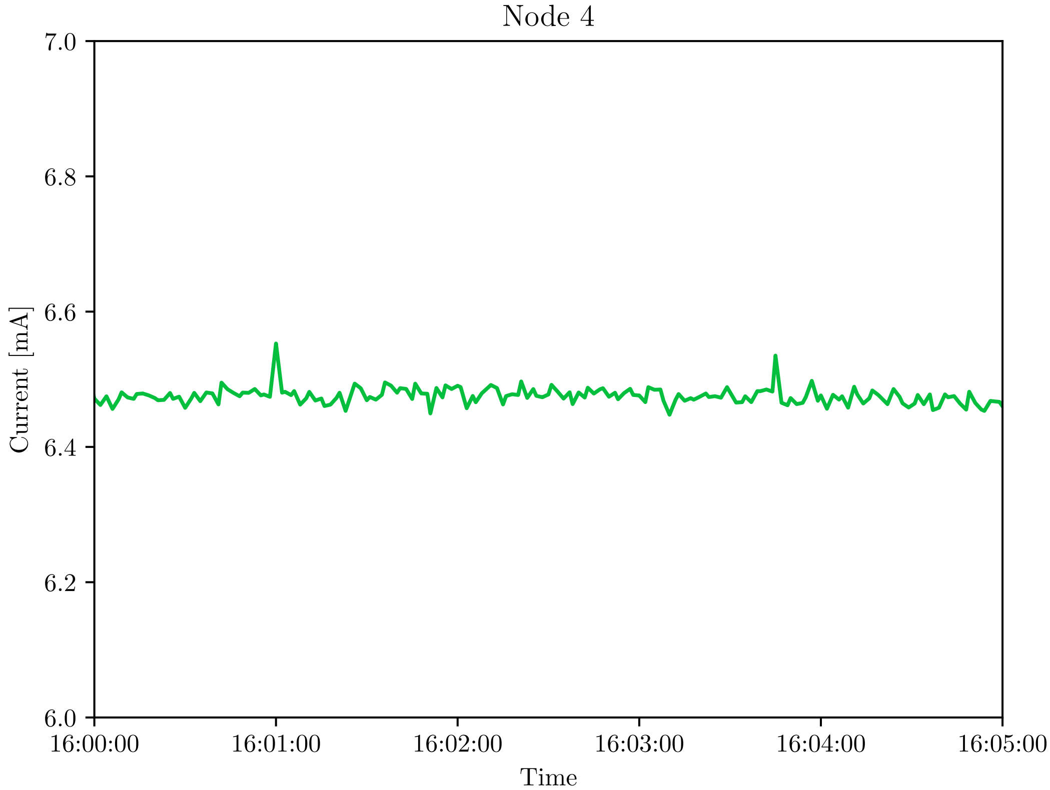


Figure 22: Current consumption for sensor 4 during Thread send test under revised condition, using 4 nodes. Showing a short interval of measurements.

Benchmarking Thread, 4 nodes

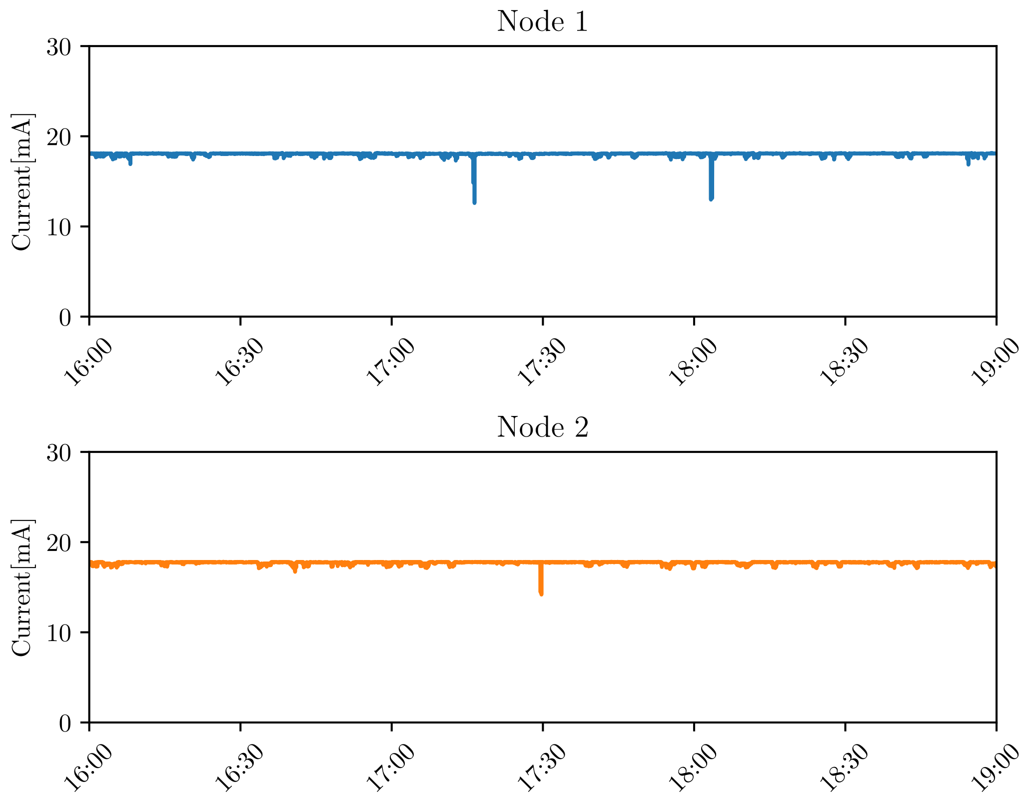


Figure 23: Current consumption for nodes during Thread benchmark test under revised condition, using 4 nodes.

Benchmarking Thread, 4 nodes

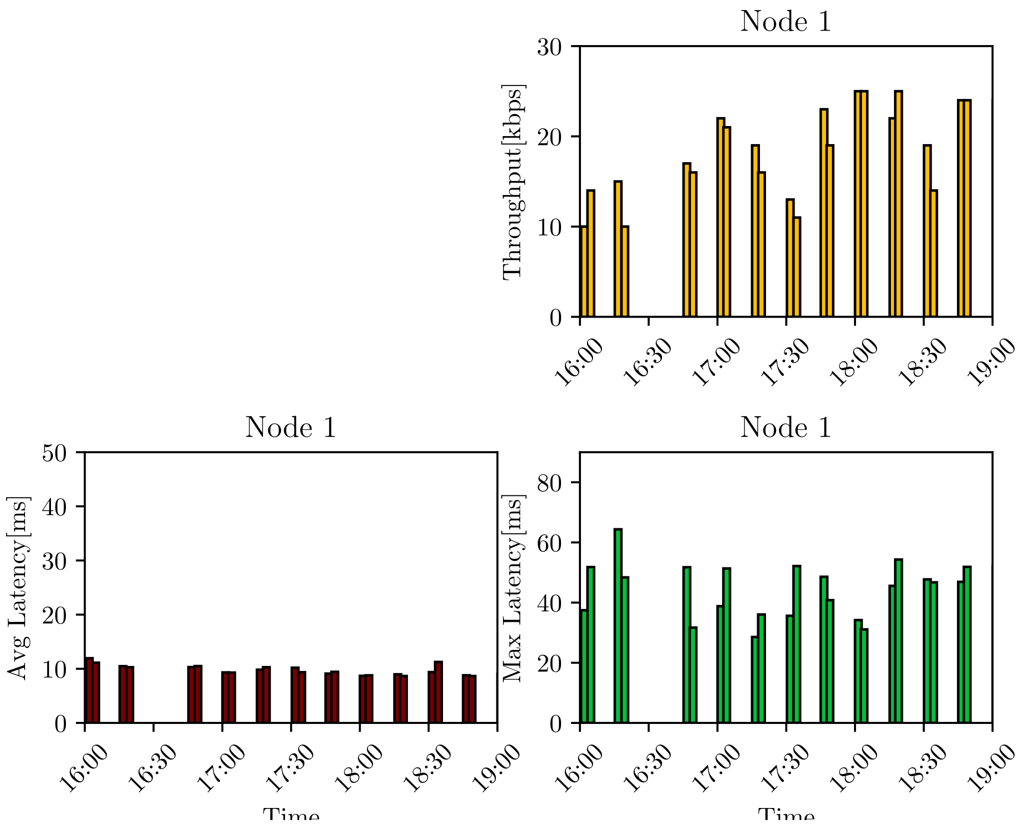


Figure 24: Throughput and latency for node 1 during Thread benchmark test under revised condition, using 4 nodes.

Benchmarking Thread, 4 nodes

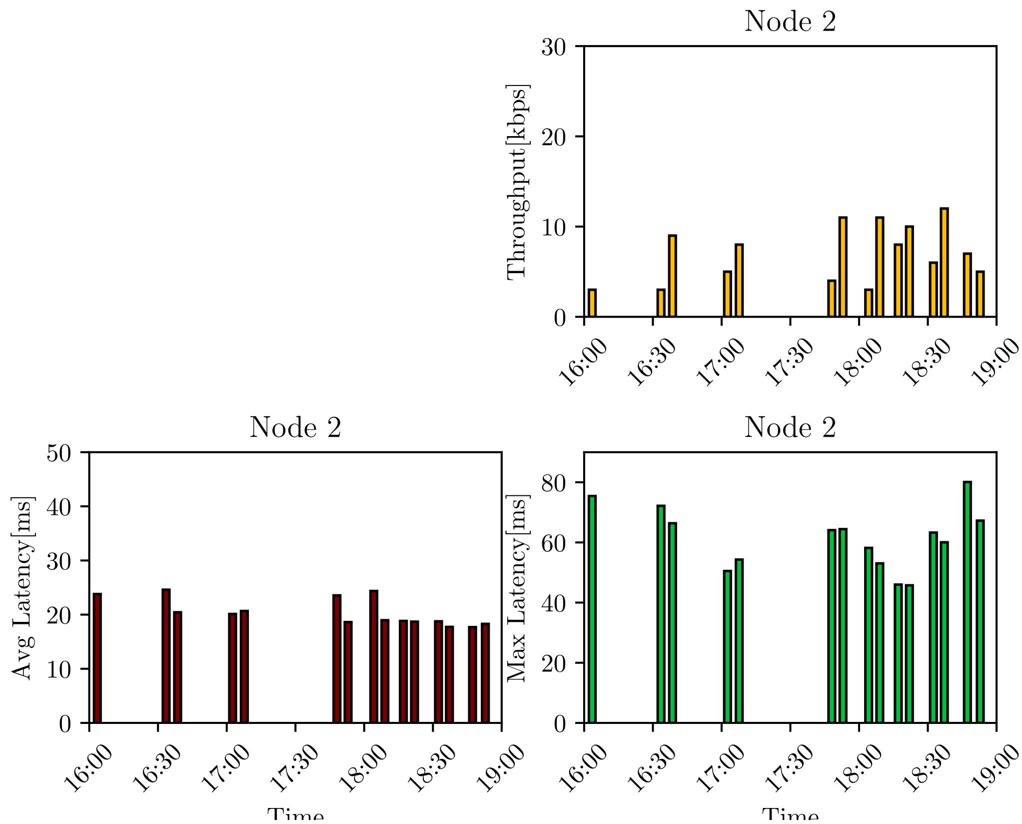


Figure 25: Throughput and latency for node 2 during Thread benchmark test under revised condition, using 4 nodes.

Benchmarking Thread, 4 nodes

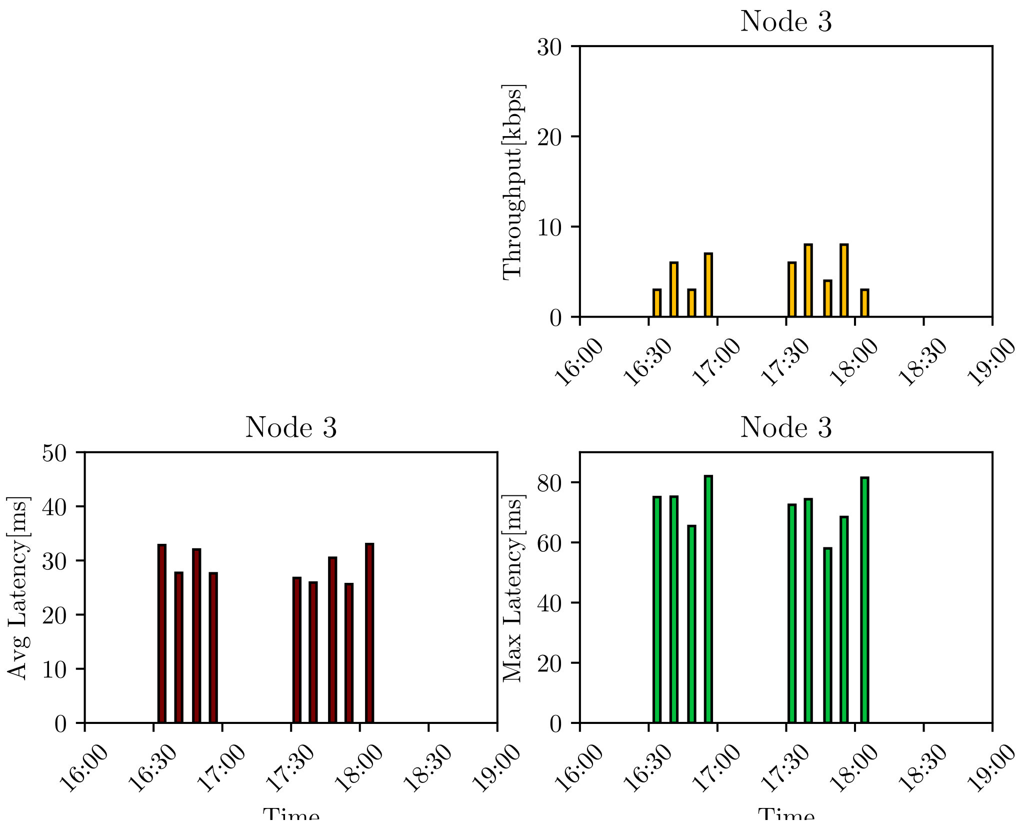


Figure 26: Throughput and latency for node 3 during Thread benchmark test under revised condition, using 4 nodes.

Benchmarking Thread, 4 nodes

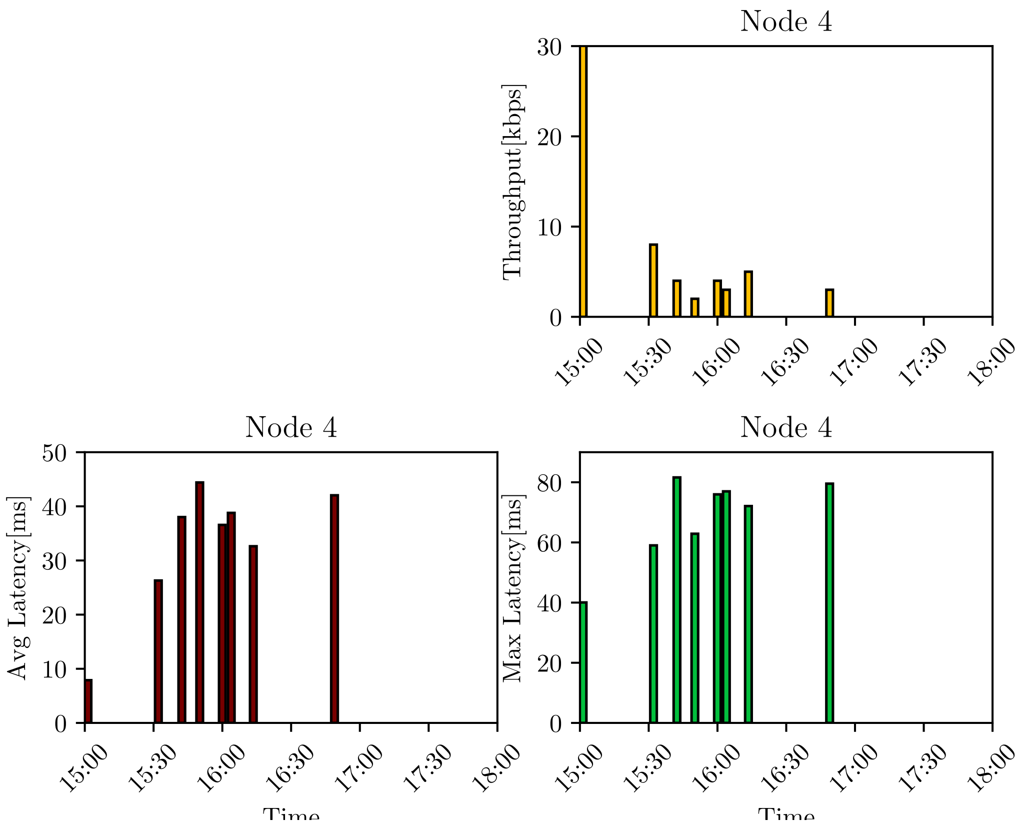


Figure 27: Throughput and latency for node 4 during Thread benchmark test under revised condition, using 4 nodes.

Benchmarking Thread, 7 nodes

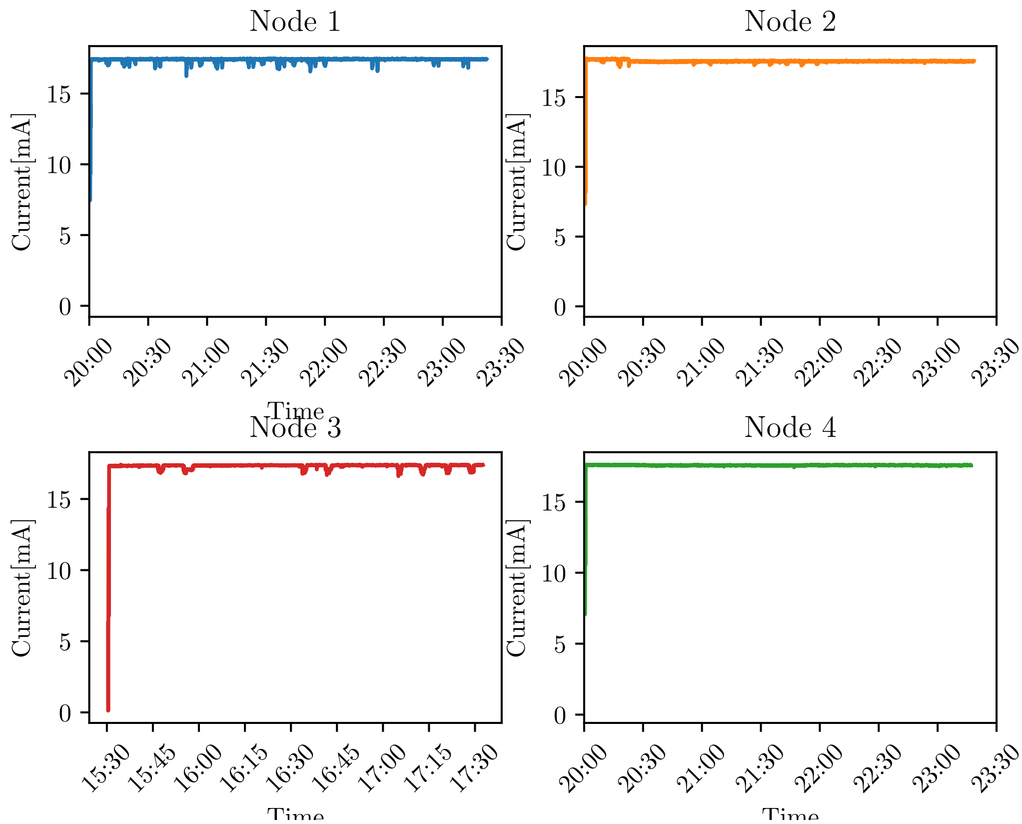


Figure 28: Current consumption for sensor nodes during Thread benchmark test under revised condition, using 7 nodes.

Benchmarking Thread, 7 nodes

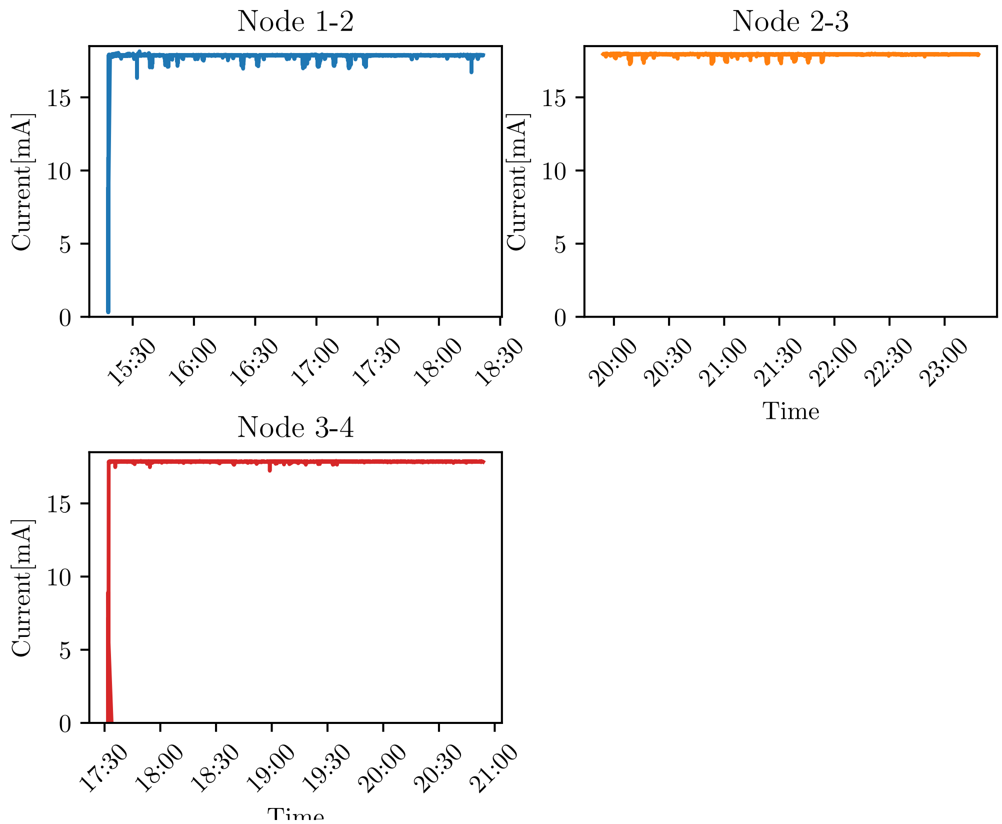


Figure 29: Current consumption for router nodes during Thread benchmark test under revised condition, using 7 nodes.

Benchmarking Thread, 7 nodes

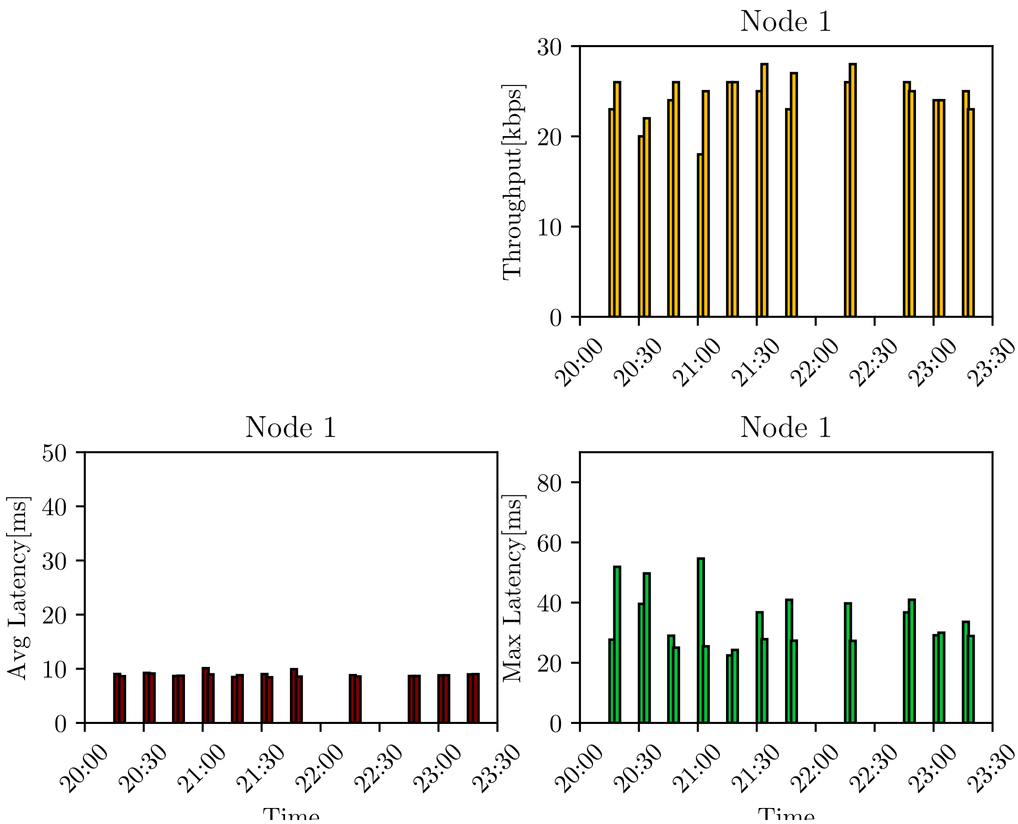


Figure 30: Throughput and latency for node 1 during Thread benchmark test under revised condition, using 7 nodes.

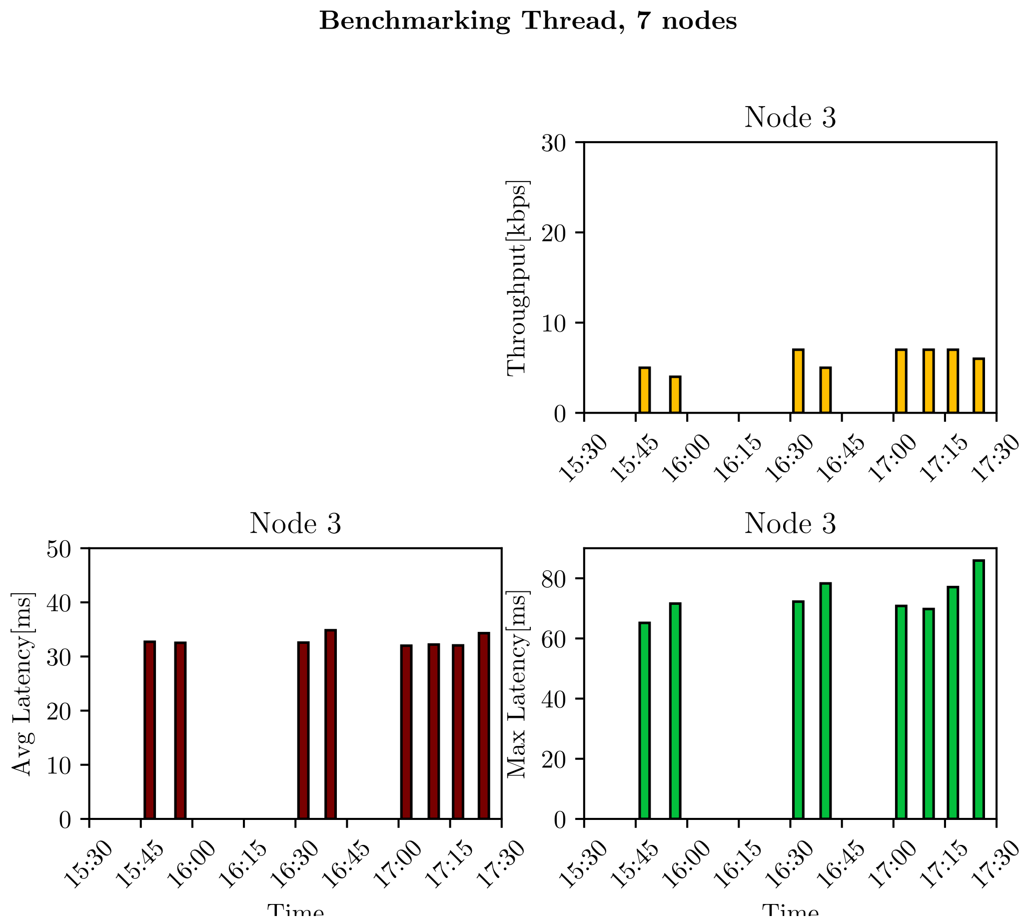


Figure 31: Throughput and latency for node 3 during Thread benchmark test under revised condition, using 7 nodes.

6.2.1 ZigBee test results

Current consumption logged in the ZigBee send test with 4 nodes for original conditions can be seen in Fig. 32. Measurements for node 3 were discarded due to the PPK2 bug, as mentioned in Section 6.1.2.

Current consumption logged in the ZigBee send test with 4 nodes for revised conditions can be seen in Fig. 32. Two different RTC clocks were used for initializing time. One for nodes 1 and 2, and the other for nodes 3 and 4. The RTC module used for nodes 3 and 4 were configured with the wrong time. Therefore, measurements from nodes 3 and 4 are not synchronized. Node 4 is one hour out of sync due to being configured for another timezone.

Fig. 34 shows current consumption from sensor 4, during the send test with four nodes. The figure shows a sample size of ten minutes, from 12:55 to 13:05.

Throughput, average latency and max latency during ZigBee benchmarking with 4 nodes, for node 2 can be seen in Fig. 35. Remark that the axis of the graphs are not scaled equally. Node1 was used as a benchmarking target, and therefore had no measurements. Node 3 and 4 were not able to connect to the ZigBee network after reprogramming as explained in Section 6.1.4, for unknown reasons.

Send test ZigBee, 4 nodes, original conditions

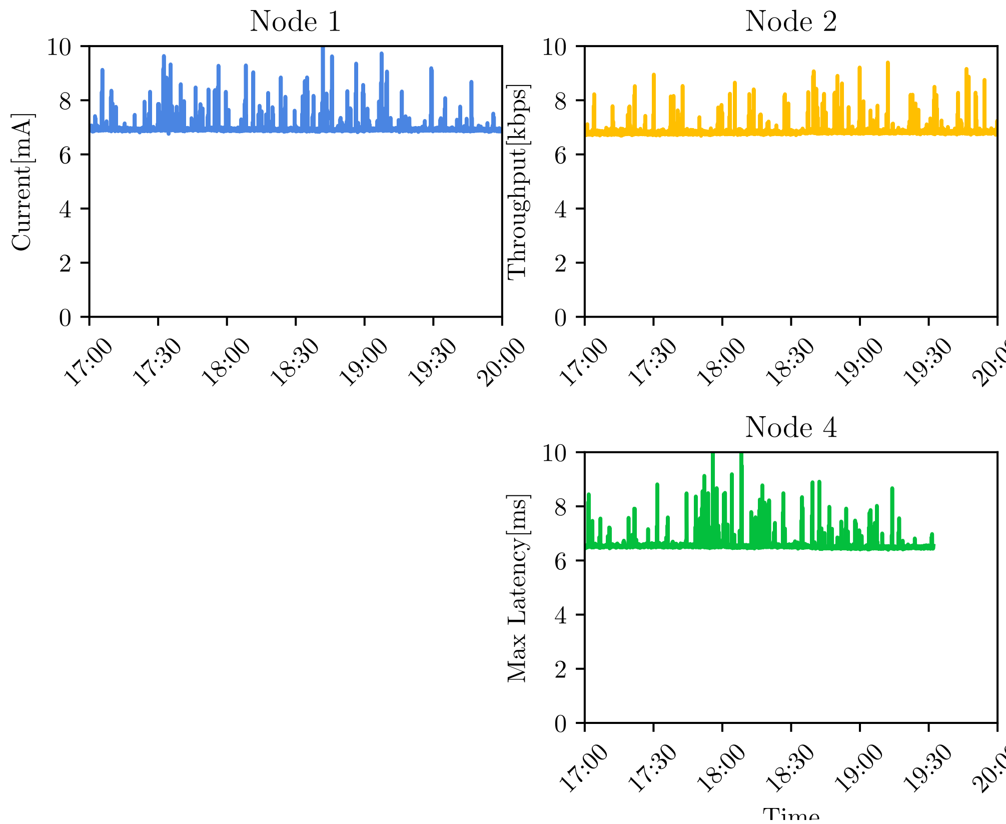


Figure 32: Current consumption for nodes during ZigBee send test under original condition, using 4 nodes.

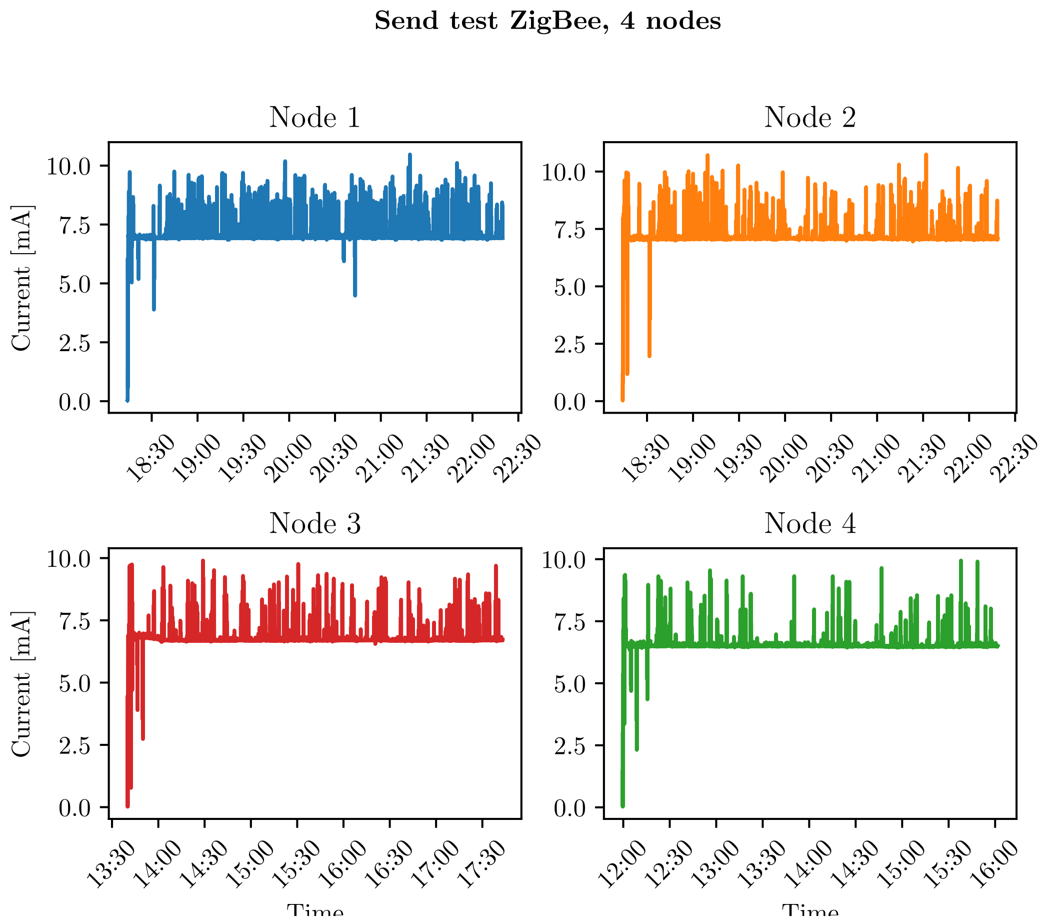


Figure 33: Current consumption for nodes during ZigBee send test under revised condition, using 4 nodes.

Send test ZigBee, 4 nodes

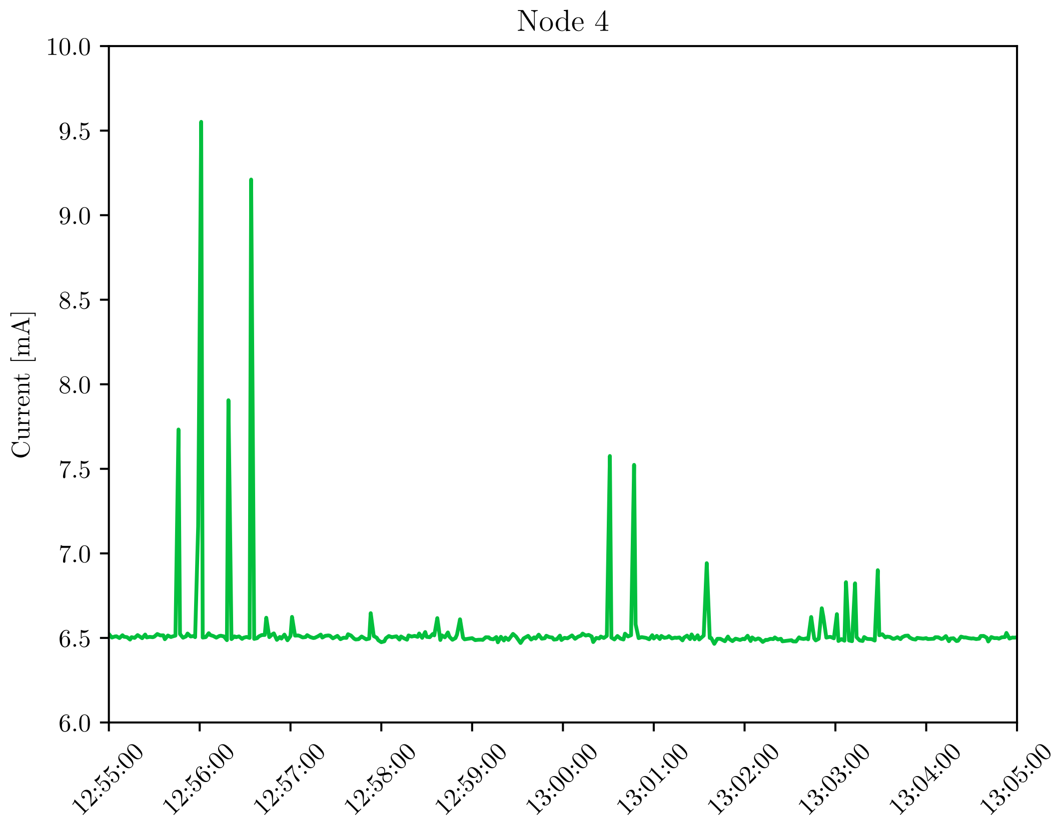


Figure 34: Current consumption for sensor 4 during ZigBee send test under revised condition, using 4 nodes. Showing a short interval of measurements.

Benchmarking ZigBee, 4 nodes

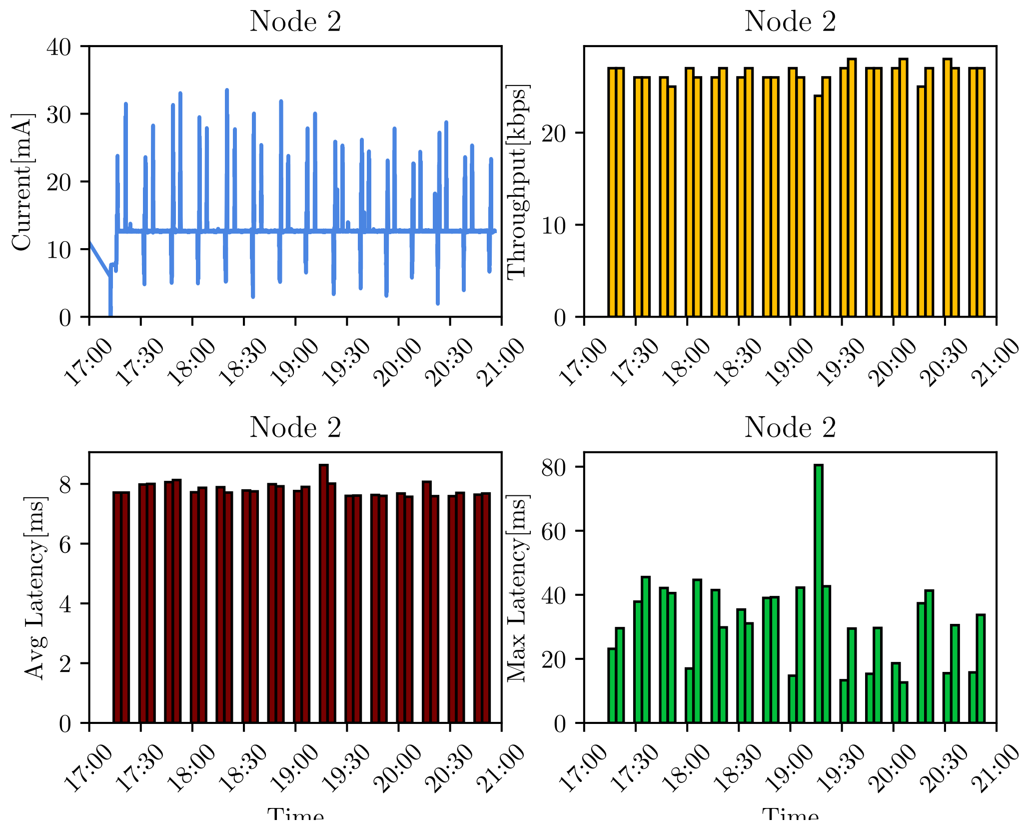


Figure 35: Current consumption, throughput and latency measurements for node 2 during ZigBee benchmarking test under revised condition, using 4 nodes.

6.2.2 LTE-M test results

All LTE-M tests were performed using revised conditions. Current consumption logged in the LTE-M send test with 4 nodes for conditions can be seen in Fig. 36. Measurements for node 3 were dropped due to date synchronization.

Fig. 37 shows current consumption from sensor 4, during send test with four nodes. The figure shows a sample size of five minutes, from 15:00 to 15:05.

Throughput, average latency and max latency during LTE-M benchmarking with 4 nodes, for node 2 can be seen in Fig. 38. Remark that the axis of the graphs are not scaled equally. The overnight benchmarking test for all 4 nodes failed and spent all data on SIM cards. Only the SIM card for node 2 was recharged.

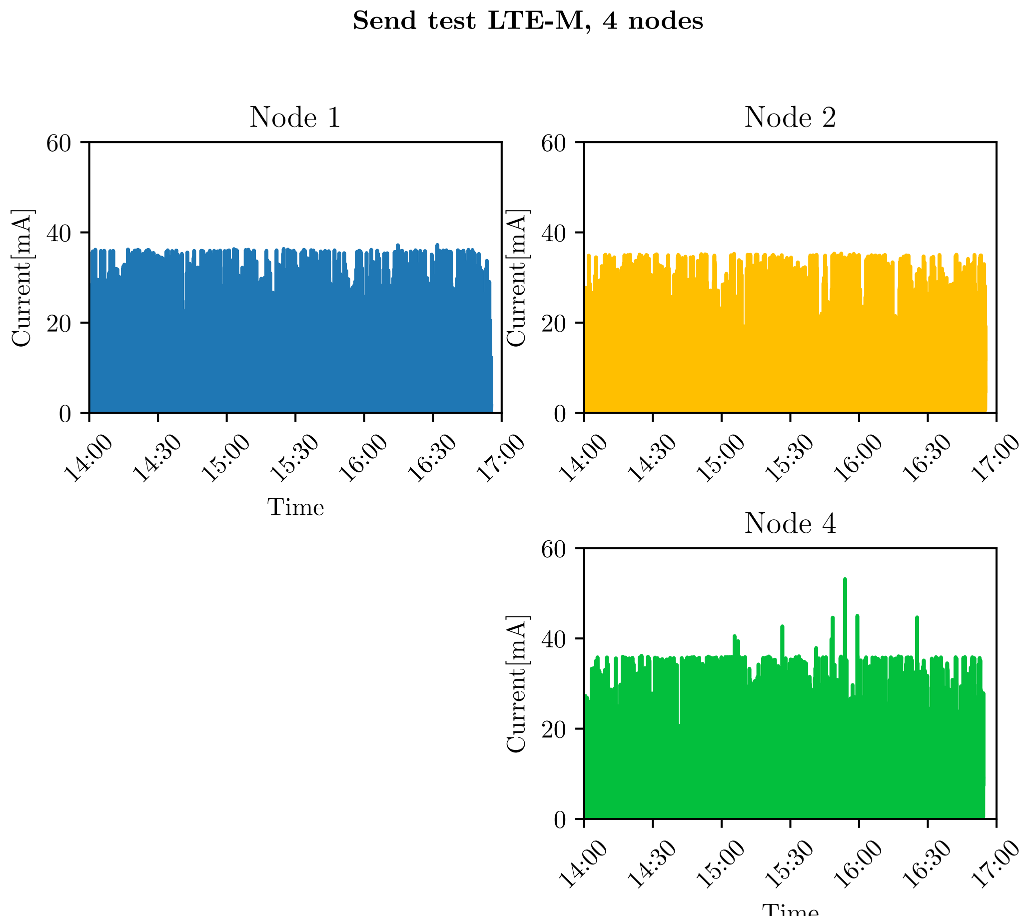


Figure 36: Current consumption for nodes during LTE-M send test under revised condition, using 4 nodes.

Send test LTE-M, 4 nodes

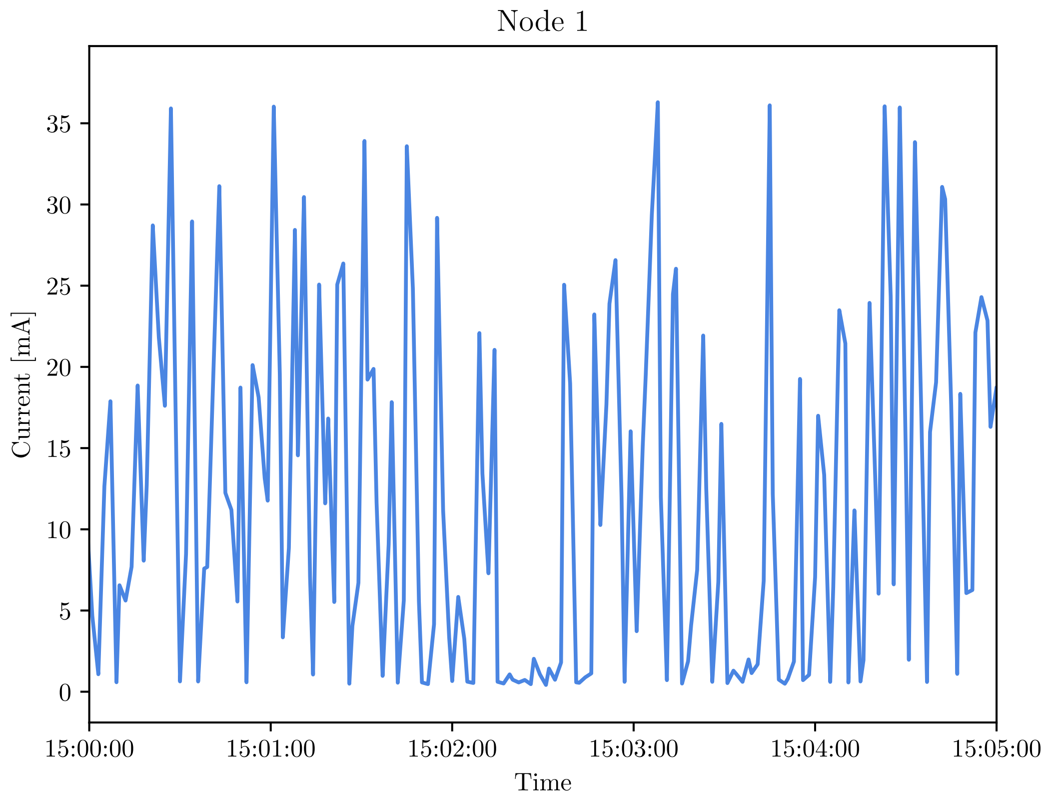


Figure 37: Current consumption for sensor 1 during LTE-M send test under revised condition, using 4 nodes. Showing a short interval of measurements.

Benchmarking LTE-M, 2 nodes

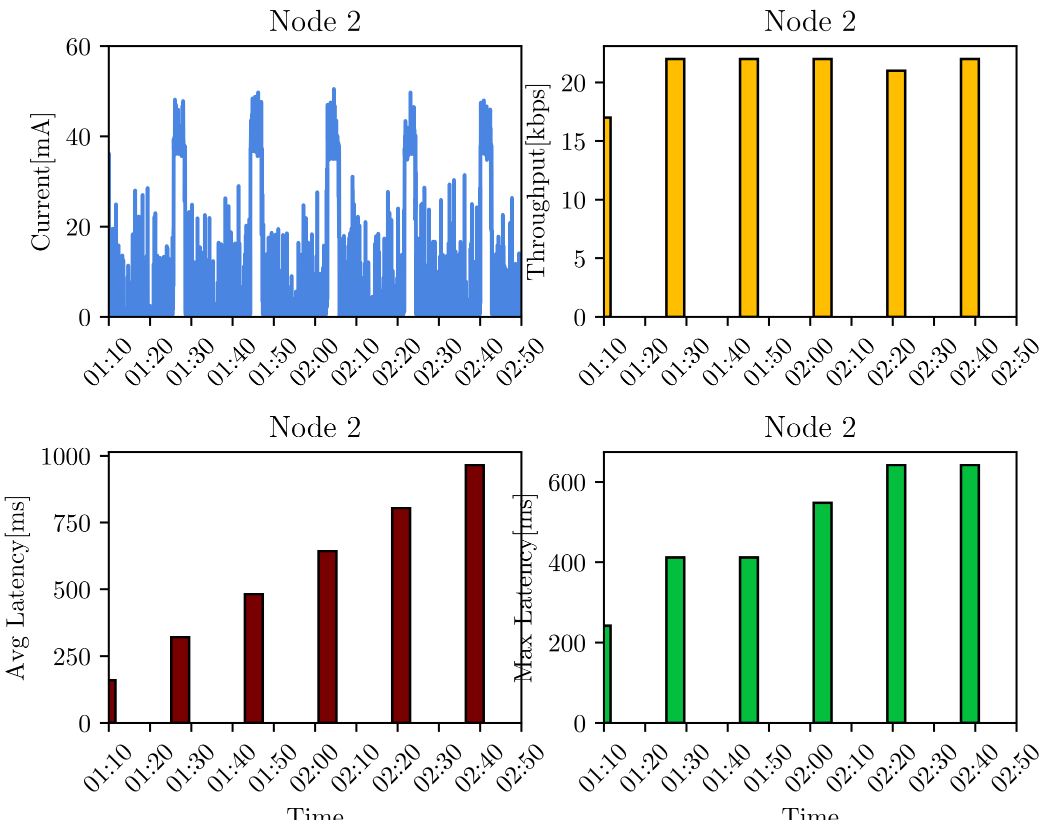


Figure 38: Throughput and latency for node 2 during LTE-M benchmark test under revised condition, using 2 nodes.

6.2.3 NB-IoT test results

Current consumption logged in NB-IoT send test with 4 nodes for conditions can be seen in Fig. 39. Measurements for node 3 were dropped due to date synchronization.

Benchmarking tests for NB-IoT was dropped due to time constraints.

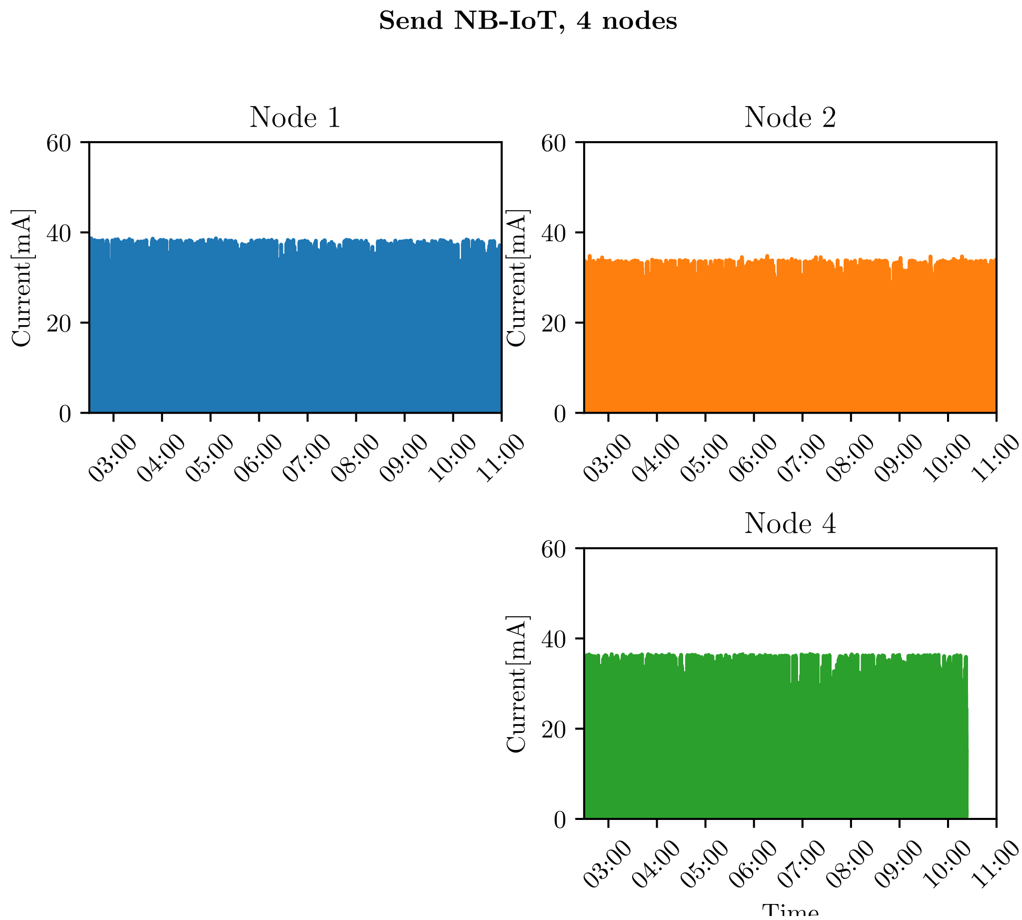


Figure 39: Current consumption for nodes during NB-IoT send test under revised condition, using 4 nodes.

6.3 Pictures and description of tests

The following figures show photographs of the testing equipment used for the real tests. Fig. 40 shows the full equipment for a single node. Fig. 41 and Fig. 42 show the different parts of the setup. Fig. 43 Show how the equipment fits inside the box, while Fig. 44 and Fig. 45 show how the box would look when deployed. These pictures are from mesh network tests. For cellular tests, setup was the same, but with nRF9160DK instead of nRF52840DK.

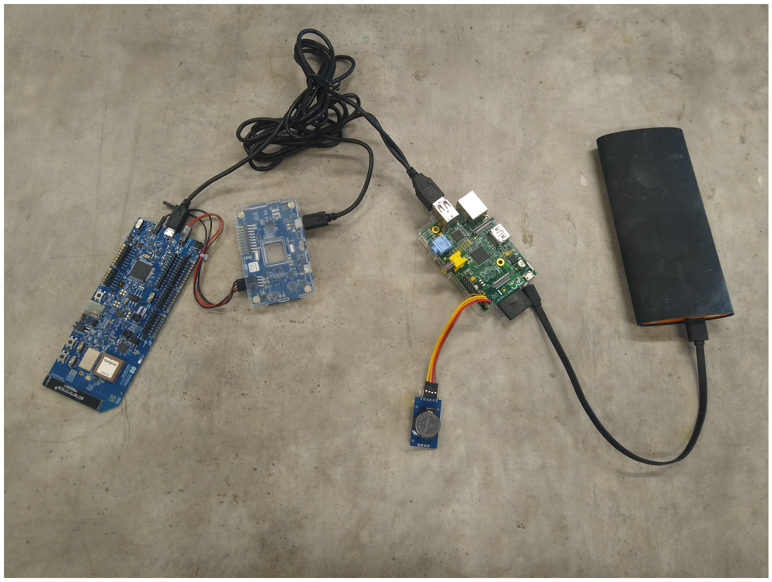


Figure 40: All parts from the test nodes.

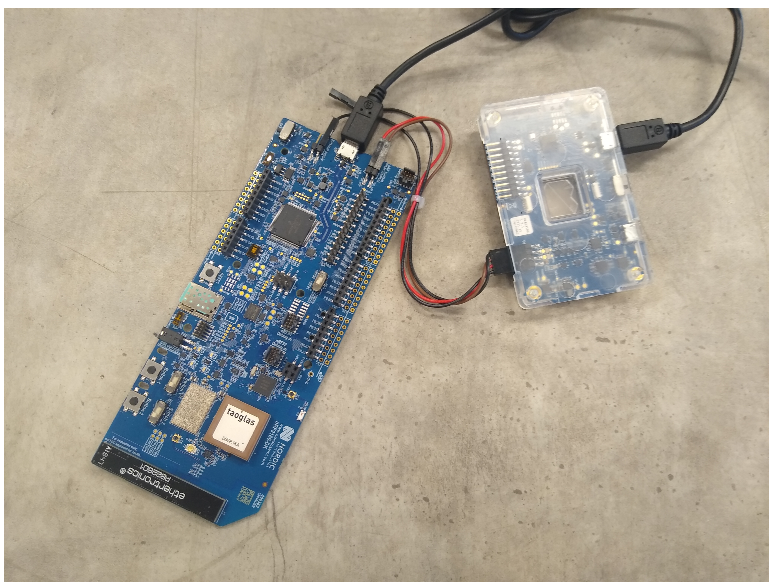


Figure 41: Closeup of the nRF52840DK and Power Profiler Kit 2.

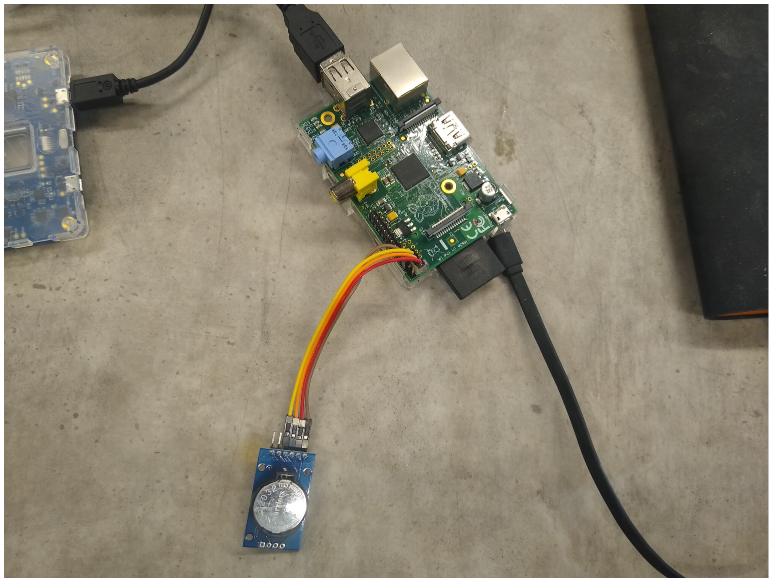


Figure 42: Closeup of Raspberry Pi 1, with a Real Time Clock (RTC) connected.

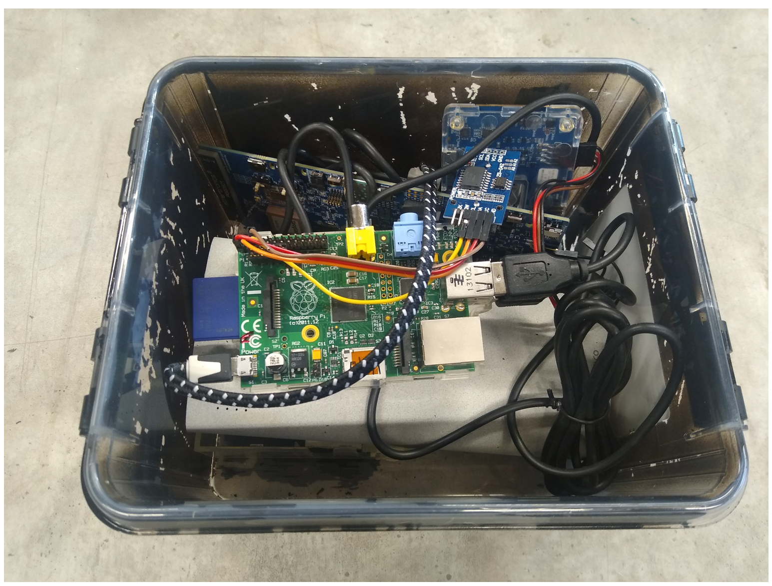


Figure 43: The test equipment was placed inside black box.

Figure 44: Closed black box with test equipment inside. Also has a warning label on the outside. In tests, my name and contact information was also written on the box, but this was removed for the picture.

Figure 45: The box was hidden out of sight, to get as little attention as possible.

6.4 Node placement

As mentioned in Section 5.10, the nodes were to be placed such that the mesh network connects linearly. This was done by using the nRF Thread Topology Monitor(TTM) [40] to visualize the thread network, and moving the nodes physically until connected linearly. For the original test, nodes were placed outdoors, 70 meters apart, as seen in Fig. 46. For the revised test, nodes were placed inside a building, ca 30 meters apart, as seen in Fig. 47.

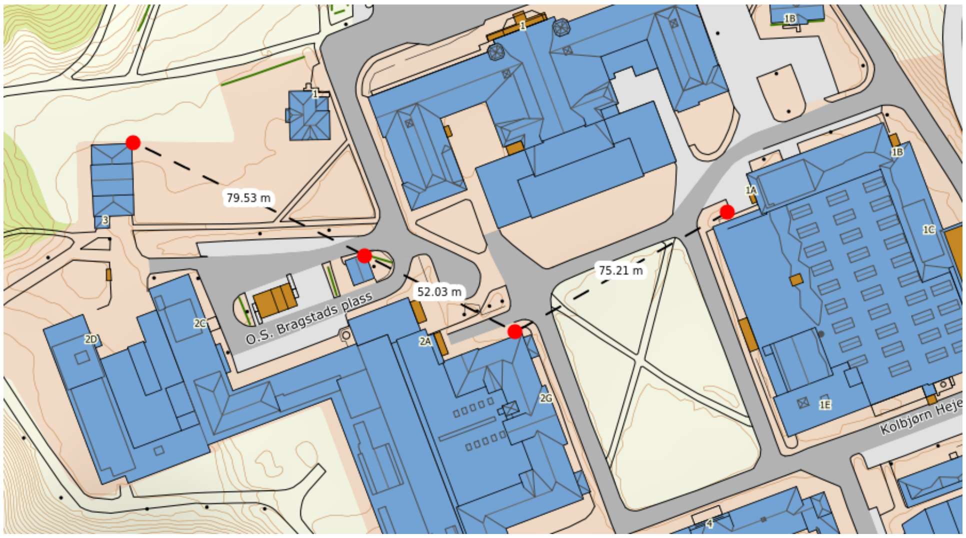


Figure 46: Map from original node placement.

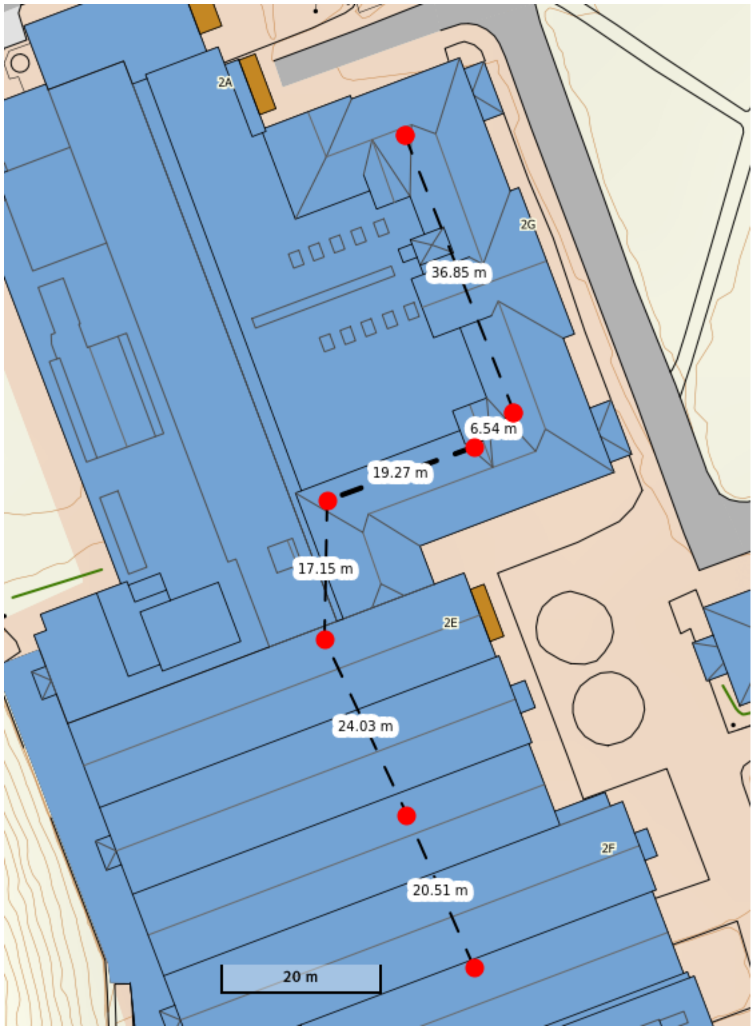


Figure 47: Map from revised node placement. The distance marked 6.54m is larger in reality, due to altitude.

6.5 Interference

The interference module was used in all mesh tests except for the original ZigBee test and the ZigBee benchmarking test. For the original ZigBee test, the interference module was not yet ready. Due to the problems with the ZigBee benchmarking tests, the interference module was skipped here, to not introduce more possible error sources.

7 Discussion

7.1 Discuss reliability

The reliability tests described in Section 5.5.2 were partly implemented. However, due to time restrictions, the reliability tests were not finished, and therefore not a part of the final tests. In the results folder of the git repository [19], "server_logs" can be found. Here, messages that arrived at the cloud server can be seen. Some reliability can be found from these. However, this shows when messages arrived at the server. The error may be in any step on the route from the mesh to the server. Either in the mesh node not sending, another node not forwarding the message or in the cloud router. To properly look at reliability, the test described in Section 5.5.2 should be implemented. Then those results could be compared to the cloud server logs to find how reliability affects the wireless networks.

7.2 Discuss interference

Interferences were a part of these experiments in two ways: passive interference and actively introduced interference.

The actively introduced interference is described in Section 4.5, Section 5.5.5 and Section 6.5.

In the Thread benchmarking test, the effects from the interference are clear. After two hours of operation, sensor 3 and sensor 4 lose connection to the mesh network. This can be seen in Fig. 26 for sensor and Fig. 27 for sensor 4. For comparison, it can be seen in Fig. 24 that sensor 1 is still connected

This indicates that when the interference were activated on sensor 2, it could no longer receive signals from sensor 3. Since the nodes were connected in a line, traffic from sensor 4 will neither arrive, since it travels through sensor 3. The interference is physically at the same location as sensor 2. Therefore, the interference noise will be a lot stronger than the signals that have traveled all the way from sensor 3, and will not be received. Remark that sensor 2 could still send to sensor 1. This is because the interference and sensor 2 both send signals from the physical location of sensor 2. When these signals both arrive at sensor 1, they have both lost the same amount of strength, and sensor 1 can make out the signal of sensor 2.

The effects of the interference can not, however, be seen in the current measurements from either the send test, Fig. 21, or the benchmark test, Fig. 23. Unfortunately, no current measurement is available for the nodes 3 and 4 during the benchmarking test.

The passive interference is the interference present due to testing being performed at a busy campus. This affected the possible range between the test nodes, as seen in Section 5.10. To understand better the effect of interference on the wireless signals in this project, the passive interference should be measured. This was not done in this project, but could be interesting as further work.

Generally, the interference seems to affect the range of the mesh networks.

Active interference was not used for the cellular tests. Due to the proximity of cellular base stations, it was assumed that cellular interference would not cause the cellular sensors to lose connection. In addition, cellular base stations are designed to support many hundred devices. Cellular devices will behave differently based on distance from base stations, as seen in Section 2.4. It would be interesting to add active interference to cellular nodes, to find if the interference would affect the cellular communication in the same way as range would. This could be done as further work.

7.3 Causes for Latency

Under ideal conditions, all wireless messages would spend the same amount of time traveling from one device to the other. The messages are transmitted in the same way, over the same medium(air), and should use the same amount of time. In real uses, conditions are never ideal, however. This will introduce different latencies due to multiple different causes; Most often retransmissions to ensure that packages are transmitted successfully. Different protocols will have different causes for latency.

7.3.1 ZigBee latency

The MAC layer of the networking standard 802.15.4 uses CSMA/CA. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used to avoid collisions between messages transmitted. In short, CSMA/CA will listen to the network, waiting until no other nodes are transmitting. Then it will wait a random amount of time before transmitting. This will introduce delays in the ZigBee communication.

Secondly, the ZigBee Application Support Sublayer (APS) adds retransmissions. APS defines that ACK is returned after sent messages, and that the sender will retransmit messages if ACK is not returned after a delay. This may also introduce latency in sending. APS retransmission is defined in the ZigBee specification [10], chapter 2.2.8.4.4.

7.3.2 Thread latency

Thread uses 902.15.4, and the CSMA/CA delay described in Section 7.3.1 will also be relevant for Thread.

Thread does not specify an application layer. Therefore, application layer retransmission depends on the application. In this thesis, two different applications are used: CoAP for the send test, and a Custom benchmarking application from Nordic. CoAP messages can be sent either as Conformable (CON) or Non-Conformable (NON). [41] Conformable messages will be retransmitted if the receiver does not return ACK. Messages sent by the CoAP clients in the sensor nodes are configured as NON.

7.3.3 LTE-M and NB-IoT latency

LTE-M and NB-IoT both use CoAP in the send test, and are configured to use Non-Conformable (NON) messages, as in Section 7.3.2.

7.4 Discuss Current measurement

Current measurements return a measurement every two seconds, as noted in Section 5.5.1. This measurement frequency is too low to get information on the exact behavior of the node. The frequency could not be higher, due to the extended interval of time used for testing. This is intended, as average current over time is of more interest for comparing different protocols. If measurements with higher frequencies are needed, it is possible to do it quite fast if a PPK2 [23] is available. Alternatively, an online power profiler exists at [42].

7.4.1 Discuss current measurements for mesh networks

Thread and ZigBee mesh networks can have different nodes; The main being end devices and routers, as seen in Section 3.4 and Section 3.3. As nodes are configured as routers in the tests, they will not be able to sleep. This results in a constant high power consumption. This can be seen in Fig. 21, where the current is around 6.5mA for the whole test. The best strategy for low power devices to save power is to enter sleep mode when no work is required. Only the outer node could possibly have been configured as an End Device in the mesh test. And only End Devices can sleep in the Thread or ZigBee mesh.

In the nRF52840 Product Specification [43], section 5.2.1.5, expected radio current consumption can be found.

Measurements of the Thread sensors for the send test, show that they generally use about 6.5 mA, seen in Fig. 21 and Fig. 27. The overview over the whole test period looks like there are few spikes to the current consumption. In the measurements of five minutes from sensor 4, this is confirmed. It seems that the thread network consumes the same amount of energy when the router is on, regardless of whether it is transmitting the messages for the CoAP test or not. The Thread router will be in Receive mode all the time, and seen from the nRF52840 Product Specification [43], power consumption I_{RADIO_RX2} is expected to be 6.53mA. This matches the measurement from the Thread send tests.

Measurement from the ZigBee send test can be seen in Fig. 33, with a closer look in Fig. 34. The current consumption of the ZigBee mesh nodes have a lot more current spikes than the Thread network, despite both using the same physical and data link layers. The current spikes seen in Fig. 34 are at times ca 10 seconds apart, which would coincide with the 10 seconds send interval. However, at other times, such as from 12:59 to 13:00, the current consumption is stable. The cloud server did receive messages during this time. Because of this, it is possible that the spikes in the current measurement is due to ZigBee protocol functions, rather than sending over the mesh network.

Observe that for Thread and ZigBee send tests for 4 nodes, the current consumption for node 4 i a bit lower than for nodes 1 to 3, seen in Fig. 23 and Fig. 33. This is likely due to the fact that nodes 1 to 3 all relay data, while node 4 is only sending, as it is at the edge of the mesh networks.

7.4.2 Discuss cellular current measurements

The LTE-M current consumption seen in Fig. 36 has a lot of variance. The current graph swings from almost 0mA to 35mA. The expected modem current consumption can be found in the nRF9160 Product Specification, 5.2.1.15. [44] LTE-M is also known as Cat-M1. In the table, under "Average current consumption, radio resource control (RRC) mode, Cat-M1", it can be seen that I_{RMC_0DBM} ("Pout 0 dBm, QPSK, 1SC, 15 kHz, TX 33% RX 33% ("balanced TX and RX")") = 40mA. From this, 35mA would be a reasonable max current for tests done in this project.

Section 5.5.3 states that sensors shall send data every ten seconds during the send test. An error in the programming resulted in cellular nodes sending every five seconds instead during the send test. In Fig. 37, where a five minutes span of the LTE-M send test current consumption, current spikes can be seen individually. There are approximately five seconds between each spike, which coincides well with the sending frequency. Current measurements are made over two second intervals, as seen in Section 7.4. The current measurement interval does not line up perfectly with the send interval of the LTE-M test. This is most likely the reason for the different sizes of the current consumption spikes seen. Spikes will become higher when the intervals coincides, and lower when both low current and send current is within the same interval.

Current consumption for NB-IoT for send test using 4 nodes, as seen in Fig. 39, is very similar to the LTE-M current consumption graph for the same test. The reasoning would be the same for the NB-IoT and the LTE-M tests.

PSM is activated during the cellular tests. From the configuration from the code, sleep time is set to 30 min, and active timeout is set to one minute. As the sensor transmits every five seconds, the active timeout will never expire, end the sensor will never enter PSM sleep mode. Two different approaches could be used to make the device use less power. Firstly, the send interval should be longer, as this would be more realistic for sensor networks. Usually, ten seconds is too frequent. Multiple different sensor data send intervals could be compared as well. Secondly, RAI could be activated before the minute of active timeout ran out. This would allow the device to enter sleep mode more after, and thereby save more power. These improvements could be performed as further work.

7.4.3 Discuss benchmarking current

Current consumption for benchmarking test can be seen in Fig. 23 for Thread, Fig. 35 for ZigBee and Fig. 38 for LTE-M.

Current measurements for Thread sensors during the benchmarking test can be seen in Fig. 23. The current measures as almost 20 mA. From section 5.2.1.5 in the nRF52840 Product Specification [43], this can not be any single protocol. From the Benchmarking example documentation [39], it can be found that the Benchmarking example from the nRF5 SDK also uses BLE. Therefore, the current drain seen in the measurements here would include BLE current consumption as well as Thread current consumption. In further experiments such as these, the BLE functionality in the benchmarking sample should be disabled, to get an isolated look at Thread current consumption during benchmarking.

BLE will also be active during the ZigBee benchmarking test, so the current consumption measurements from this test do not give isolated ZigBee measurements. ZigBee nodes were reprogrammed before each benchmarking test. Benchmarking tests were performed twice every fifteen minutes. The two test were performed 2 minutes apart, as the tests were performed on Node 2, as explained in Section 5.5.4. This behavior can be seen in the current consumption, which will fall (reprogramming) before spiking twice each 15 minutes.

For the LTE-M, current consumption spikes every fifteen minutes, at the same time as the benchmarking tests are run.

The current consumption graphs for the Thread benchmarking tests show no distinguishing alternation.

7.5 Discuss benchmarking tests

From the Thread benchmarking tests, seen in Fig. 24, Fig. 25, Fig. 26 and Fig. 26, the throughput is approximately 25kbps for node 1, 10kbps for node 2, 7kbps for node 3 and 5kbps for node 4. It can be seen that throughput will decrease noticeably when the signal has to be relayed through multiple nodes of a mesh network.

Average latency increase linearly for each jump in the thread mesh, seen in Fig. 38. Node 1 has a latency of approximately 10ms, Node 2 is 20ms, Node 3 is 30 ms and Node 4 is 40ms. Each jump seems to add 10ms.

The throughput from the benchmarking test for Thread, Fig. 26, and benchmarking test for ZigBee, Fig. 35, are very similar, showing that the throughput of ZigBee and Thread are capped similarly. Node 2 from the ZigBee benchmarking sample did not use any relays to get to the benchmarking hub.

The throughput of the LTE-M benchmarking test, seen in Fig. 38, is approximately 25 kbps, which is the same as for the mesh networks. This is coincidentally, as the mesh and cellular protocols use different physical layers for communication.

The average latency of the LTE-M benchmarking test increases linearly. This is most likely due to an error in the code, where the counter is not emptied between tests, and the counter increases. Because of the linear increase, we know that the average latency will be the difference between each increase: 160ms.

The LTE-M latency is 160ms, while the mesh latencies are much lower, at 10 to 40 ms. This is most likely because the LTE-M benchmarking test sends larger packages than the mesh benchmarking test. The packages will thereby spend more time transmitting, but transmit more data. It should be remarked that the LTE-M benchmarking test was developed in this thesis, while the mesh benchmarking example is from the official Nordic nRF5 SDK. [18] In the development of the cellular benchmarking test, the packet size was planned to be the same size as for the mesh example. However, an error must have happened.

Due to time constraint, NB-IoT functionality was not added to the cellular benchmarking test. In further work, the cellular benchmarking test should be revised, to fix bugs and add functionality for NB.IoT, to fix bugs and add functionality for NB-IoT.

7.6 Discuss difference between 4 and 7 nodes

The only test performed with seven nodes was the Thread benchmarking test for 7 nodes. The test results for the 7 nodes were similar as for the 4 nodes benchmarking test. As mentioned in Section 7.4.3, the current consumption for the benchmarking test includes BLE current consumption, and therefore does not reflect the Thread mesh. However, the benchmarking test for 7 nodes shows the same current consumption as for the test with 4 nodes. Relay nodes had the same current consumption as sensor nodes. Both types of nodes were configured as Thread routers, so this was expected. Measurements for node 1 for the Thread benchmarking tests for 4 and 7 nodes showed the same results, seen in Fig. 24 and Fig. 30. In the Thread benchmarking test for 7 nodes, node 3 was number five in the row of nodes. Measurements for node 3 for the Thread benchmarking tests for 7 nodes and node 4 for the Thread benchmarking test for 7 nodes showed also showed very similar results, seen in Fig. 31 and Fig. 27. It was expected that nodes in similar positions in the mesh network would test similarly.

7.7 Discuss test system

As seen in Table 9, tests were not completed fully. This was partly due to bugs and limitations explained in Section 6.1, and partly due to time restrictions.

Timestamp bugs, described in Section 6.1.1, were due to an implementation which required steps of the tests to be activated in a specific order. This introduced human error, where tests could be started in the wrong way. Tests should preferably be fully automated, where the operator can activate tests using a single simple step. This could be done as further work.

The PPK2 bug and the Raspberry Pi programming bug, described in Section 6.1.2 and Section 6.1.3, resulted in corrupt or no test results. Often, these bugs could be fixed by simply restarting the equipment. Due to no way to monitor the current measurement or benchmarking tests, these errors were not discovered until the test finished. This resulted in wasted time. Real time monitoring of all test results could be implemented as further work, to increase test efficiency.

While this test was not for testing mesh communication distance, distance measurement was a challenge. It is easy to check if two mesh nodes connect or disconnect, to find a maximum range, as done in Section 2.4. However, in the test for this thesis, mesh node placement had to fulfill two different requirements: linear mesh network connection and discretion. Especially for the revised tests, where nodes had to be placed inside a building, mesh node distribution became challenging. Node distribution could possibly have been impossible without the use of the Thread Topology Monitor (TTM) tool. TTM and node distribution are described in Section 7. If one of the requirements were removed, the process would be simplified drastically. If tests are to be repeated, it is recommended to perform tests in a large open area with no people passing through.

For mesh network internet connection, an internet connection was required, as described in Section 5.3.3. This also imposed restrictions on the node placement for tests in this thesis. There is no workaround for this, and internet connectivity must be taken into account when distributing nodes.

Except for these bugs, the test method and implementation provided the expected results for tests in an orderly and efficient way. The test setup can be regarded as two different parts: The testing equipment, and the subject equipment. In this thesis, subject equipment was either the nRF52840DK or the nRF9160DK. The testing equipment was the measurement module and the cloud server. The testing equipment is generic, and can be equipped to any subject equipment. Because of this, the tests described in this thesis are applicable for not only Nordic Semiconductor microcontrollers or even mesh/cellular communication protocols. The subject equipment could be exchanged for most low power hardware or protocols.

7.8 Discuss logistics

Even though the focus of this thesis is on the technical capabilities of the protocols, logistical capabilities should also be considered when choosing between mesh and cellular protocols. Logistical considerations could be protocol lifetime, maintenance costs or implementation and development costs, as seen in Section 2.3. In this project, the development costs were highest for ZigBee, and lowest for cellular protocols, as explained in Section 6.1.5.

It should be mentioned that cellular networks require cellular coverage in the area. On the other hand, mesh networks require that the installer has access to internet connection.

7.9 Discuss mesh

ZigBee and Thread wireless communication protocols are both mesh protocols built on IEEE 802.15.4. Because of this, measurements of the two protocols are very similar. In particular, throughput and latency measurements were very similar for ZigBee and Thread.

The largest difference between Thread and ZigBee, was the current consumption spikes. While both protocols had similar average current consumption of approximately 6.5 mA, the ZigBee measurements had a lot of current spikes, up to 10 mA. As current measurements are measured as an average for two seconds, spikes in current may look flattened. This is probably the reason that spikes are not seen at sending intervals for the Thread current measurements. The spikes for the ZigBee measurement is therefore most likely that the ZigBee current consumption spikes are wide or high enough to show up on the averaged current consumption, in difference from Thread.

7.10 Discuss cellular

As seen in Section 3.5, NB-IoT is expected to have longer range, but less throughput than LTE-M. Unfortunately, benchmarking tests were not performed on NB-IoT in this project.

For tests in this thesis, LTE-M and NB-IoT performed very similarly for current measurements. This is different from what was found in Section 2.2. However, in this project, the same power saving configuration was used for both protocols. It would seem that LTE-M and NB-IoT would consume similar amounts of power, since the protocols use the same hardware. The main thing that could result in different power usage for NB-IoT and LTE-M would be transmission power. If NB-IoT has a longer range than LTE-M, NB-IoT could possibly use lower transmission power at the same ranges. In this project, both protocols were in an urban area, and therefore close enough to cellular base stations to transmit with low transmission power. As further work, it could be interesting to test current consumption for the cellular protocols for different distances to base stations, similar to the tests from both this thesis, and from Section 2.4.

7.11 Discuss mesh vs cellular

Cellular nodes measured a throughput of ca 30kbps. Mesh nodes measured a throughput of ca 25kbps for a single hop, and ca 5kbps for multiple hops. For cellular networks, all nodes would therefore have the same high throughput, while mesh networks get influenced by distance.

Latency measurements for mesh vs cellular can not be considered, as the package sizes most likely were different when sent, as discussed in Section 7.4.3.

Current consumption tests show that mesh network nodes consume ca 6.5 mA, while cellular network nodes consume 35 mA. Both these are measured while the radio is active. The devices will spend approximately $3\mu A$ when in sleep mode, found in section 5.2.1 of both the nRF52840 Product Specification [43] and the nRF9160 Product Specification [44]. As sleep current is a thousand times lower than the active radio current, devices will mainly save power based on the percentage of time they are able to sleep.

The biggest difference between power consumption for the two devices is therefore that the mesh nodes have to be configured as routers to relay traffic. And ZigBee and Thread routers can not enter sleep mode. For medium distances, where mesh networks would need multiple hops, cellular communication is a lot more power efficient.

When looking at results from the measurements, cellular networks are superior for medium distance communication. The main reason for this is that the ZigBee and Thread mesh networks require always active routers.

If looking at more practical matters, mesh networks have some advantages. Firstly, cellular networks are hosted by a third party, and subscriptions have to be paid for use of the cellular network. Mesh on the other hand, requires no payments to third parties. In addition, cellular networks are dependent on cell coverage, while mesh networks only need internet connection in general.

8 Further work

In Section 7, multiple parts of the project were found that could require further work. The most important of these will be listed below.

- Measure passive interference in the environment during tests.
- Add interference node for cellular networks.
- Run benchmarking test for mesh with BLE disabled to get isolated look at current consumption for mesh protocols.
- Implement reliability test, and run with send tests.
- Perform send tests with different sensor data transmit intervals.
- Add RAI to the cellular tests.
- Fix the cellular benchmarking test as explained in Section 7.5.
- Perform cellular tests at different distances from base stations.
- Automate test setup, in order to minimize human error.
- Real time monitoring of all test results, to increase test efficiency.

9 Conclusion

A literary study were performed, to learn about cellular and mesh networks for medium range use. In addition to insight into the logistics of wireless sensor networks, the literary study provided considerations for planning a test system.

A system for testing power consumption, reliability, throughput and latency for low power wireless communication systems were planned. This test system was implemented for different cases: nRF52840 DK for ZigBee and Mesh networks and nRF9160 DK for LTE-M and NB-IoT networks. Due to time restrictions, implementation of reliability tests were not finished.

The implemented test system was used to collect data for power consumption, throughput and latency. Despite some bugs during tests, data were collected for most test scenarios.

While ZigBee and Thread had similar results, Thread routers had a bit lower power consumption than ZigBee nodes. In addition, ZigBee development costs were high.

Mesh and cellular tests showed similar throughput.

Power consumption tests clearly favored cellular networks over multihop mesh networks. The main reason for this was that the ZigBee and Thread mesh networks require relays to always be in active listening mode, which has high power consumption. Cellular nodes on the other hand allow devices to sleep, which reduce power consumption.

In the end, the important product from this thesis was the design and implementation of the tests, which can be used not only to test LTE-M, NB-IoT, Thread or ZigBee, but can test power consumption, reliability, throughput and latency of any other low power wireless protocol.

A Tests naming and sorting

Here the folder structure for results can be found. This can also be found on git. [19]

Listing 1: Test result folder structure shown as generated from the *tree* commandline tool.

```
──2021-05-11
│   └──4send
│       ├──all_nodes_zigbee_70m_current_2021-05-11.pgf
│       ├──node1_zigbee_70m_current_2021-05-11.csv
│       ├──node2_zigbee_70m_current_2021-05-11.csv
│       ├──node3_zigbee_70m_current_2021-05-11.csv
│       └──node4_zigbee_70m_current_2021-05-11.csv
──2021-05-15
│   ├──4send
│   │   ├──all_nodes_ltem_100m_current_2021-05.15.pgf
│   │   ├──node1_ltem_100m_current_2021-05-15.csv
│   │   ├──node2_ltem_100m_current_2021-05-15.csv
│   │   ├──node3_ltem_100m_current_2021-05-15.csv
│   │   └──node4_lte_100m_current_2021-05-15.csv
│   └──README.md
├──2021-05-18
│   └──4send
│       ├──all_nodes_nbiot_100m_current_2021-05-18.pgf
│       ├──node1_nbiot_100m_current_2000-01-01.csv
│       ├──node1_thread_70m_current_2021-05-18.csv
│       ├──node2_nbiot_100m_current_2000-01-01.csv
│       ├──node3_nbiot_100m_current_2021-05-18.csv
│       └──node4_nbiot_100m_current_2021-05-18.csv
├──2021-05-19
│   └──4send
│       ├──all_nodes_thread_70m_current_2021-05-19.pgf
│       ├──all_nodes_zigbee_70m_current_2021-05-19.pgf
│       ├──node1_thread_70m_current_2021-05-19.csv
│       ├──node1_zigbee_70m_current_2021-05-19.csv
│       ├──node2_thread_70m_current_2021-05-19.csv
│       ├──node2_zigbee_70m_current_2021-05-19.csv
│       ├──node3_thread_70m_current_2021-05-19.csv
│       ├──node3_zigbee_70m_current_2021-05-19.csv
│       ├──node4_thread_70m_current_2021-05-19.csv
│       └──node4_zigbee_70m_current_2021-05-19.csv
──2021-05-21
│   ├──4benchmark
│   │   ├──node1_node2_thread_70m_current_2021-05-21.pgf
│   │   ├──node1_thread_70m_avg_latency_2021-05-21.csv
│   │   ├──node1_thread_70m_benchmark_tests_2021-05-21.pgf
│   │   ├──node1_thread_70m_current_2021-05-21.csv
│   │   ├──node1_thread_70m_max_latency_2021-05-21.csv
│   │   ├──node1_thread_70m_troughput_2021-05-21.csv
```

```
│   │   ├──node1_thread_70m_troughput_retransmitted_2021-05-21.csv
│   │   ├──node2_thread_70m_avg_latency_2021-05-21.csv
│   │   ├──node2_thread_70m_benchmark_tests_2021-05-21.pgf
│   │   ├──node2_thread_70m_current_2021-05-21.csv
│   │   ├──node2_thread_70m_max_latency_2021-05-21.csv
│   │   ├──node2_thread_70m_troughput_2021-05-19.csv
│   │   ├──node2_thread_70m_troughput_2021-05-21.csv
│   │   ├──node2_thread_70m_troughput_retransmitted_2021-05-21.csv
│   │   ├──node3_thread_70m_avg_latency_2021-05-21.csv
│   │   ├──node3_thread_70m_benchmark_tests_2021-05-21.pgf
│   │   ├──node3_thread_70m_max_latency_2021-05-21.csv
│   │   ├──node3_thread_70m_troughput_2021-05-21.csv
│   │   ├──node3_thread_70m_troughput_retransmitted_2021-05-21.csv
│   │   ├──node4_thread_70m_avg_latency_2021-05-21.csv
│   │   ├──node4_thread_70m_benchmark_tests_2021-05-21.pgf
│   │   ├──node4_thread_70m_max_latency_2021-05-21.csv
│   │   ├──node4_thread_70m_troughput_2021-05-21.csv
│   │   └──node4_thread_70m_troughput_retransmitted_2021-05-21.csv
│   └──7benchmark
│       ├──all_nodes_thread_140m_current_2021-05-21.pgf
│       ├──all_routers_thread_140m_current_2021-05-21.pgf
│       ├──node1_2_thread_140m_current_2021-05-10.csv
│       ├──node1_thread_140m_avg_latency_2021-05-21.csv
│       ├──node1_thread_140m_benchmark_tests_2021-05-21.pgf
│       ├──node1_thread_140m_current_2021-05-21.csv
│       ├──node1_thread_140m_max_latency_2021-05-21.csv
│       ├──node1_thread_140m_troughput_2021-05-21.csv
│       ├──node1_thread_140m_troughput_retransmitted_2021-05-21.csv
│       ├──node2-3_thread_140m_current_2021-05-21.csv
│       ├──node2_thread_140m_avg_latency_2021-05-21.csv
│       ├──node2_thread_140m_current_2021-05-21.csv
│       ├──node2_thread_140m_max_latency_2021-05-21.csv
│       ├──node2_thread_140m_troughput_2021-05-21.csv
│       ├──node2_thread_140m_troughput_retransmitted_2021-05-21.csv
│       ├──node3_4_thread_140m_current_2021-05-10.csv
│       ├──node3_thread_140m_avg_latency_2021-05-21.csv
│       ├──node3_thread_140m_benchmark_tests_2021-05-21.pgf
│       ├──node3_thread_140m_current_2021-05-21.csv
│       ├──node3_thread_140m_max_latency_2021-05-21.csv
│       ├──node3_thread_140m_troughput_2021-05-21.csv
│       ├──node3_thread_140m_troughput_retransmitted_2021-05-21.csv
│       ├──node4_thread_140m_current_2021-05-21.csv
│       └──node4_thread_140m_troughput_2021-05-21.csv
├──2021-05-22
│   └──4benchmark
│       ├──node2_zigbee_70m_all_meas_2021-05-22.pgf
│       ├──node2_zigbee_70m_avg_latency_2021-05-22.csv
│       ├──node2_zigbee_70m_current_2021-05-22.csv
│       ├──node2_zigbee_70m_max_latency_2021-05-22.csv
│       │
```

```
├──node2_zigbee_70m_troughput_2021-05-22.csv
├──node2_zigbee_70m_troughput_retransmitted_2021-05-22.csv
├──node4_zigbee_70m_current_2021-05-22.csv
└──node4_zigbee_70m_troughput_2021-05-22.csv
```

References

[1] Nordic semiconductor asa webpage. https://www.nordicsemi.com/. Accessed: 2021-06-11.

[2] Nassr Myasar Abdullah and Rabee Hagem. The experience of using particle mesh for lagre iot projects with power and range management, 2020.

[3] Tim Ekman and Samuel Jönsson. An empirical study of celluar-iot, master thesis, 2019.

[4] Interact lightning: Which communications platform is best for connected street lighting? https://www.interact-lighting.com/global/iot-insights/cellular-vs-rf-mesh-communications. Accessed: 2021-02-01.

[5] Field-testing how distance affects the behavior of lte-m and nb-iot. https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/field-testing-how-distance-affects-the-behavior-of-lte-m-and-nb-iot. Accessed: 2021-02-02.

[6] nrf52840-dk range testing with ble, zigbee and thread protocols at 0, 4 and 8dbm transmit power settings. https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/nrf52840-dk-range-testing-with-ble-zigbee-and-thread-protocols-at-0-4-and-8dbm-transmit-power-settings. Accessed: 2021-02-02.

[7] Behrtech website. https://behrtech.com/about-us/. Accessed: 2021-02-10.

[8] Low-power mesh networks vs. lpwan: What you need to know, by behrtech. https://behrtech.com/resources/low-power-mesh-networks-vs-lpwan-guide/. Accessed: 2021-02-10.

[9] Network topologies in wireless solutions. https://blog.nordicsemi.com/getconnected/wireless-network-topologies. Accessed: 2021-05-25.

[10] Zigbee technical specifications. https://zigbeealliance.org/solution/zigbee/. Accessed: 2021-02-10.

[11] Thread group members. https://www.threadgroup.org/thread-group. Accessed: 2021-06-03.

[12] Openthread website. https://openthread.io/. Accessed: 2021-06-09.

[13] Wikipedia, lte frequency bands. https://en.wikipedia.org/wiki/LTE_frequency_bands. Accessed: 2021-06-04.

[14] Nordic webinar: Introduction to cellular iot. https://webinars.nordicsemi.com/introduction-to-cellular-iot-4. Accessed: 2021-06-04.

[15] Nordic infocenter, release assistance indication https://infocenter.nordicsemi.com/topic/ref_at_commands/REF/at_commands/nw_service/rai.html. Accessed: 2021-06-04.

[16] nrf connect sdk 1.4.2. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/introduction.html. Accessed: 2021-02-17.

[17] nrf connect for desktop. https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-desktop. Accessed: 2021-02-17.

[18] nrf5 sdk v17.0.2. https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk_nrf5_latest.html. Accessed: 2021-02-17.

[19] Public gitlab repository for code used in this thesis. https://gitlab.com/hellesvik/cellular_vs_mesh_test_setup. Accessed: 2021-06-10.

[20] nrf52840dk product page. https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK. Accessed: 2021-02-17.

[21] nrf5 sdk for thread and zigbee v4.1.0. https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk_thread_zigbee_latest.html. Accessed: 2021-02-17.

[22] nrf9160dk product page. https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF9160-DK. Accessed: 2021-02-17.

[23] Nordic power profiler kit ii. https://www.nordicsemi.com/Software-and-tools/Development-Tools/Power-Profiler-Kit-2. Accessed: 2021-02-17.

[24] Ppk2 python api by irnas. https://github.com/IRNAS/ppk2-api-python. Accessed: 2021-06-01.

[25] Raspberry pi 3b+ product page. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. Accessed: 2021-02-17.

[26] Raspbian operating system. https://www.raspbian.org/. Accessed: 2021-02-17.

[27] Ubuntu for raspberry pi. https://ubuntu.com/raspberry-pi. Accessed: 2021-02-17.

[28] nrf9160: nrf coap client. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/openthread/coap_client/README.html. Accessed: 2021-02-17.

[29] Zigbee: Light switch. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/zigbee/light_switch/README.html. Accessed: 2021-02-17.

[30] nrf9160: nrf coap client. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/nrf9160/coap_client/README.html. Accessed: 2021-02-17.

[31] Thread: Cli. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/openthread/cli/README.html. Accessed: 2021-02-17.

[32] Openthread border router. https://openthread.io/guides/border-router. Accessed: 2021-02-17.

[33] Thread: Ncp. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/openthread/ncp/README.html. Accessed: 2021-02-17.

[34] Zigbee: Ncp. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/zigbee/ncp/README.html. Accessed: 2021-02-17.

[35] Zigbee: Network coordinator. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/zigbee/network_coordinator/README.html. Accessed: 2021-02-17.

[36] Debian distribution. https://www.debian.org/. Accessed: 2021-06-01.

[37] libcoap: a c-implementation of coap. https://libcoap.net/. Accessed: 2021-02-17.

[38] Deltaco pb-1050 power bank. https://www.dustinhome.se/product/5010777981/pb-1050?ssel=false. Accessed: 2021-06-08.

[39] nrf5: Benchmark application example. https://infocenter.nordicsemi.com/topic/sdk_tz_v4.1.0/benchmark.html. Accessed: 2021-02-24.

[40] nrf thread topology monitor. https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF-Thread-topology-monitor. Accessed: 2021-06-01.

[41] Coap technical specifications. https://datatracker.ietf.org/doc/html/rfc7252. Accessed: 2021-05-31.

[42] Nordic semiconductor: Online power profiler. https://devzone.nordicsemi.com/nordic/power/w/opp. Accessed: 2021-06-06.

[43] nrf52840 product specification. https://infocenter.nordicsemi.com/topic/ps_nrf52840/keyfeatures_html5.html?cp=4_0_0. Accessed: 2021-06-06.

[44] nrf9160 product specification. https://infocenter.nordicsemi.com/topic/ps_nrf9160/nRF9160_html5_keyfeatures.html?cp=2_0_0#. Accessed: 2021-06-06.

NTNU

Norwegian University of
Science and Technology