Development of Robust Wireless Sensor Network Communications for an Embedded Real Time System on Construction Sites

Alexandru Cohal

~ Master's Thesis ~

Examiners: Prof. Dr. Paul Lukowicz
Prof. Dr.-Ing. Wolfgang Kunz
Supervisor: Marco Hirsch

November 2018

Overview

- Motivation
- Problem Statement
- State of the Art
- Contributions
- TI CC1350 Microcontroller
- Spectrum Access Methods
- IoT Standards and Protocols
- Application Layer Protocols
- Security
- Solution
- Implementation and Evaluation
- Conclusions

Motivation

- The construction industry → one of the most important sectors
  - ↳ number of workers → 6,4% of Europe's employment (2017)
  - ↳ invested money → 1,36 *billion Euro* in the EU (2017)

- The construction industry → 20% of fatal accidents at work in the EU (2015)
- The construction equipment theft → between 300 and 1000 *million Dollars* yearly in the USA → less than 25% is recovered

- The equipment and tools → not always brought back to the storage space

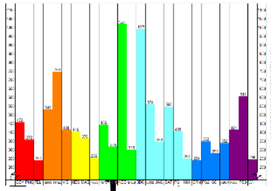
↓

- Take care of employees
- Increase the profit
- Optimize the workflow

→

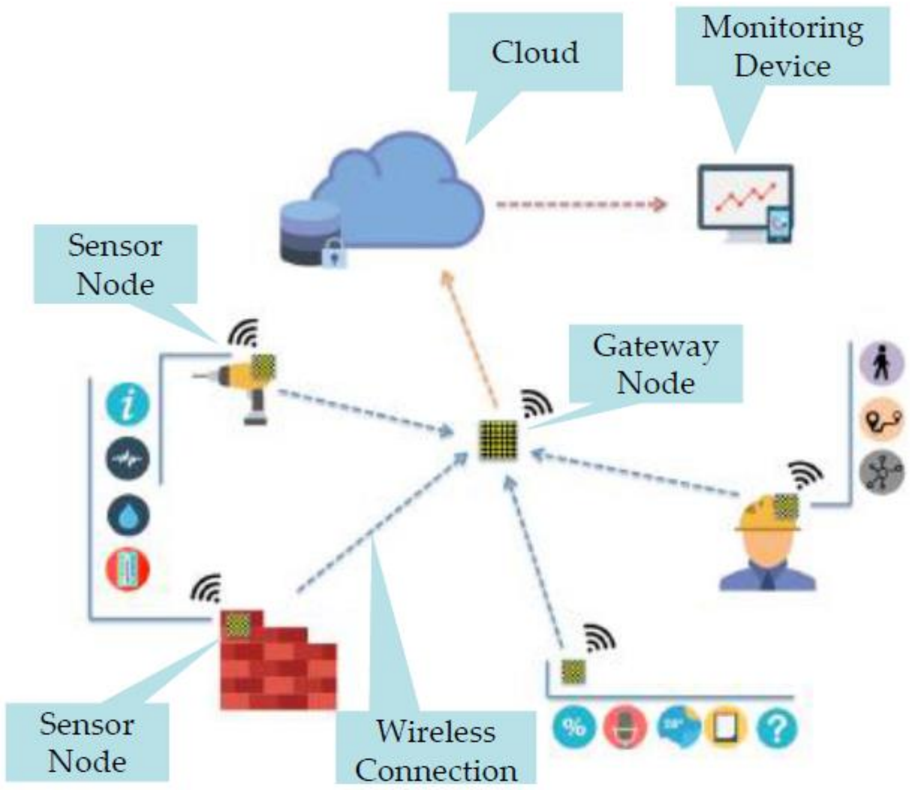
Monitor → statuses of the machines and tools
→ health of the workers
→ environment

Track → equipment, materials, workers

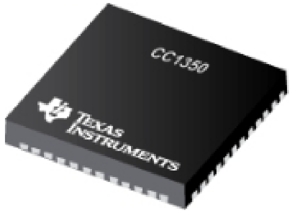


Problem Statement

- A solution ➡ combine *IoT* and a *Wireless Sensor Network* (*WSN*)



Already chosen for this project

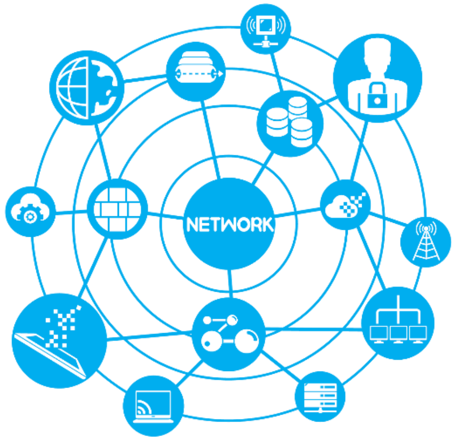


- The *Texas Instruments*' ultra-low power dual-band wireless ($Sub - 1\ GHz$ and $2.4\ GHz$) $CC1350$ microcontroller ➡ used due to its advantages

State of the Art

- Multiple solutions ➡ based on WSN
  - ↳ intelligent transportation
  - ↳ smart water network
  - ↳ resource tracking
  - ↳ work environment monitoring

- The differences ➡ WSN solution ➡ this project
  - ↳ handle both $Sub - 1\ GHz$ and 2.4 GHz bands
  - ↳ not depend on any infrastructure (e.g. base stations, power)
  - ↳ easy to extend / reduce
  - ↳ easy to deploy
  - ↳ easy to use
  - ↳ compatible with third-party $Sensor\ Nodes$



Contributions

- The main goal
  - → Develop a $Communication\ Protocol\ Stack$ for a $CC1350$-based WSN
    - → to run on the $Real - Time\ Operating\ System\ TI - RTOS$
    - → to be used → construction industry

- The WSN has to be
  - → Predictable
  - → Scalable
  - → Robust
  - → Reliable
  - → Secure
  - → Mobile

  → requirements and specifications for the $CC1350$-based WSN → defined

- $CC1350$'s features → compared with the newer versions $CC1350P$ and $CC1350R$

Contributions

- Overview and comparison
  - → the most popular wireless Standards and Protocols used in IoT (**29**)
  - → $Application\ Layer$ protocols (**6**)
  - → $Spectrum\ Access\ Methods$ (**7**)
  - → security vulnerabilities and solutions
- The best choices are selected → based on → requirements and specifications
  - → $CC1350$'s features
  - → for both $Sub-1\ GHz$ and $2.4\ GHz$ bands

→ { A $Communication\ Protocol\ Stack$
An Architecture of the whole system

TI CC1350 Microcontroller

- Ultra-Low Power Dual Band Wireless Microcontroller
- Frequencies supported

  → $\boldsymbol{Sub-1\ GHz}$ (higher ranges, ability to pass through walls, lower power consumption)

  → sending the messages within the WSN

  → $\boldsymbol{2.4\ GHz}$ (higher popularity protocols – e.g. Bluetooth)

  → configuring / checking the Nodes using a Smartphone / Tablet

- The European Commission → defined a series of regulations → *Short Range Devices* using the *Sub-1 GHz* ISM bands

  → Spectrum Access regulations

Frequency Band	Spectrum Access
863 – 870 MHz	≤ 0.1% Duty Cycle *OR* LBT + AFA

Spectrum Access Methods

- **Duty Cycle** → defines → maximum total transmission time per hour
  → maximum time of a single transmission
  → minimum time between 2 consecutive transmissions
  ↳ Duty Cycle of 0.1% → 3.6 *seconds* max. total transmission time per hour
  ↳ Analysis → Maximum number of transmitted packets per hour
    ↳ Different standard communication protocols
    ↳ Different types of packets (e.g. GPS, IMU, ACK)
    ↳ Different data rates

SigFox
LoRa

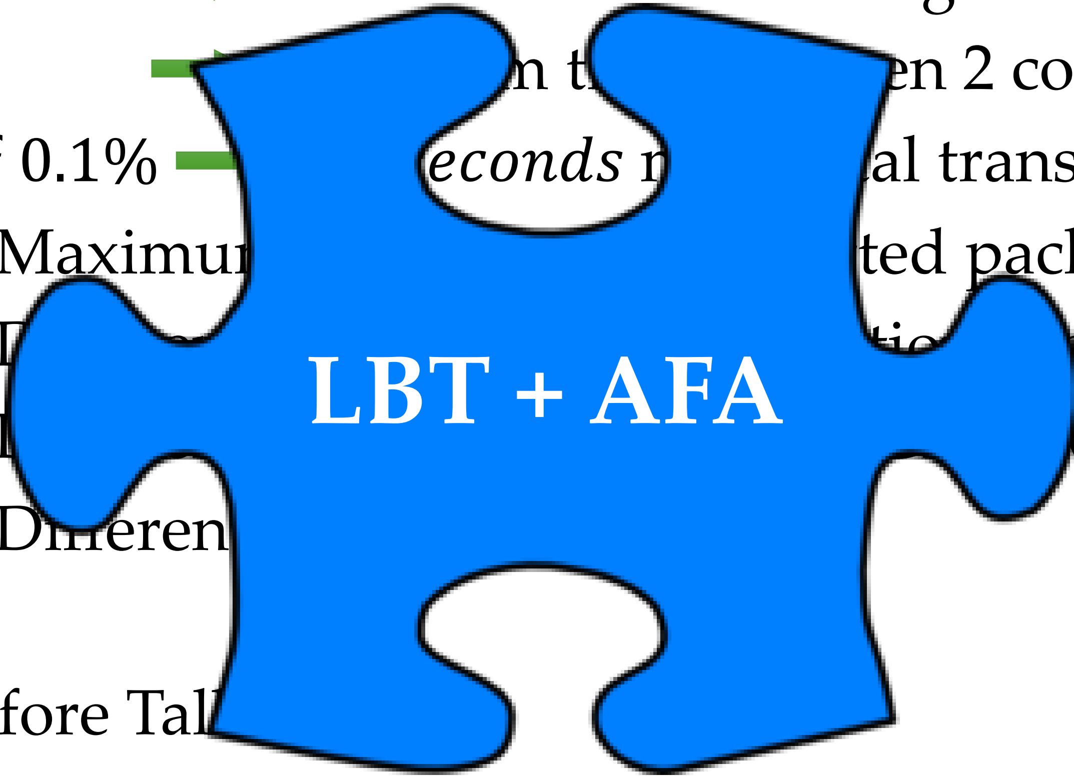
- **LBT** → Listen Before Talk
- **AFA** → Adaptive Frequency Agility → Frequency Hopping
  ↳ Adaptability → <u>*NOT IN*</u> CC1350's SDK

<u>*IN*</u> CC1350's SDK

Spectrum Access Methods

- **Duty Cycle** → defines → maximum total transmission time per hour
  → maximum time of a single transmission
  → …n t… …en 2 consecutive transmissions
  - ↳ Duty Cycle of 0.1% → … *econds* … …al transmission time per hour
  - ↳ Analysis → Maximu… …ted packets per hour
    - ↳ D… …tio… …otocols
    - ↳ I… …MU, ACK)
    - ↳ Differen…

SigFox
LoRa



- **LBT** → Listen Before Tal…
- **AFA** → Adaptive Frequency Agility → Frequency Hopping

IN CC1350's SDK

↳ Adaptability → *NOT IN* CC1350's SDK

IoT Standards and Protocols – 2.4 GHz

- For configuring / checking the Nodes using a Smartphone / Tablet
- 10 Standards and Protocols compared

BLE → covers all the stack's layers (*PHY, MAC, Network, Application*)
→ big range (up to 400 *meters*)
→ beacons can be used (indoor localization)

Zigbee → not supported by *CC*1350
→ too complex → mesh topology → point-to-point needed

Thread → based on *6LoWPAN* → direct Internet compatibility not needed

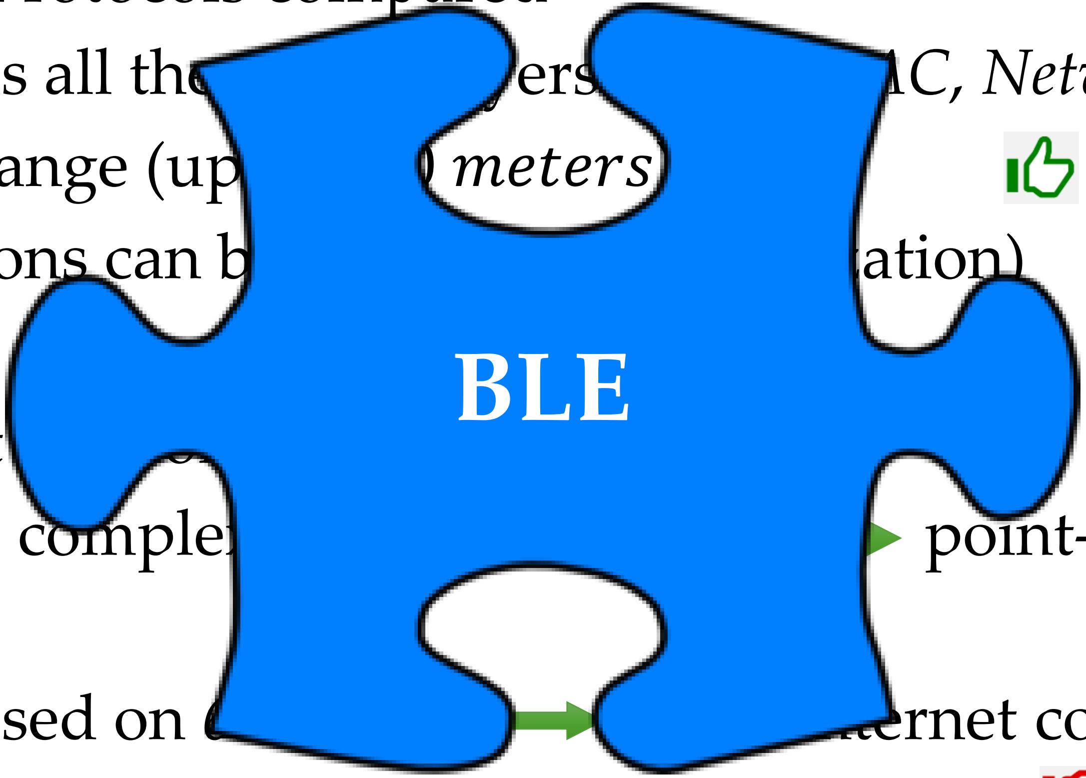
IoT Standards and Protocols – 2.4 GHz

- For configuring / checking the Nodes using a Smartphone / Tablet
- 10 Standards and Protocols compared

BLE ➡ covers all the ... ers ... C, *Network*, *Application*)

➡ big range (up ... *meters*

➡ beacons can b ... ation)



Bluetooth SMART

Zigbee ➡ not

➡ too comple ... ➡ point-to-point needed

zigbee

Thread ➡ based on ... ➡ ... ernet compatibility not needed

THREAD

IoT Standards and Protocols – Sub-1 GHz

- For sending the messages within the *WSN*
- 13 Standards and Protocols compared

TI 15.4 Stack → PHY and MAC layers based on *IEEE 802.15.4*

→ uses Frequency Hopping → not Adaptive

→ very low message overhead (28 – 40 *Bytes*) compared with the payload size (500 *Bytes*)

→ supported by *CC*1350 (included in the *SDK*)

→ open source → can be modified

LoRa → specific and not public hardware for the *PHY* layer

Zigbee → not supported by *CC*1350

→ does not use Frequency Hopping

→ very high message overhead (39 – 66 *Bytes*) compared with the payload size (73 – 100 *Bytes*)

IoT Standards and Protocols – Sub-1 GHz

- For sending the messages within the *WSN*
- 13 Standards and Protocols compared

TI 15.4 Stack → PHY a… C l… sed on *IEEE 802.15.4*

→ uses … ency Ho… → not Adaptive

→ very … ad (28 – 40 *Bytes*) compared

w… 500 … *tes*)

… the *SDK*)

… ope… odi… d

LoRa → specific and … lic ha… or the *PHY* layer

Zigbee → not supp… *CC135*…

→ does not use Frequency Hopping

→ very high message overhead (39 – 66 *Bytes*) compared with the payload size (73 – 100 *Bytes*)

Application Layer Protocols

- 6 Protocols compared

MQTT-SN → on top of any wireless protocol → not only *TCP* or *UDP*

→ based on *Publish – Subscriber* model

→ small overhead (2 or 4 *Bytes*)

→ different levels of *QoS* → at most / at least / exactly once

→ does not support frequency hopping specific messages

CoAP, XMPP, AMQP, MQTT, DDS → based on *TCP* or *UDP*

→ larger overheads

→ more suited for Gateway - Cloud

Application Layer Protocols

- 6 Protocols compared

MQTT-SN → on top of any wireless protocol → not only *TCP* or *UDP*

→ based – *S* odel

→ small o d (2 or 4

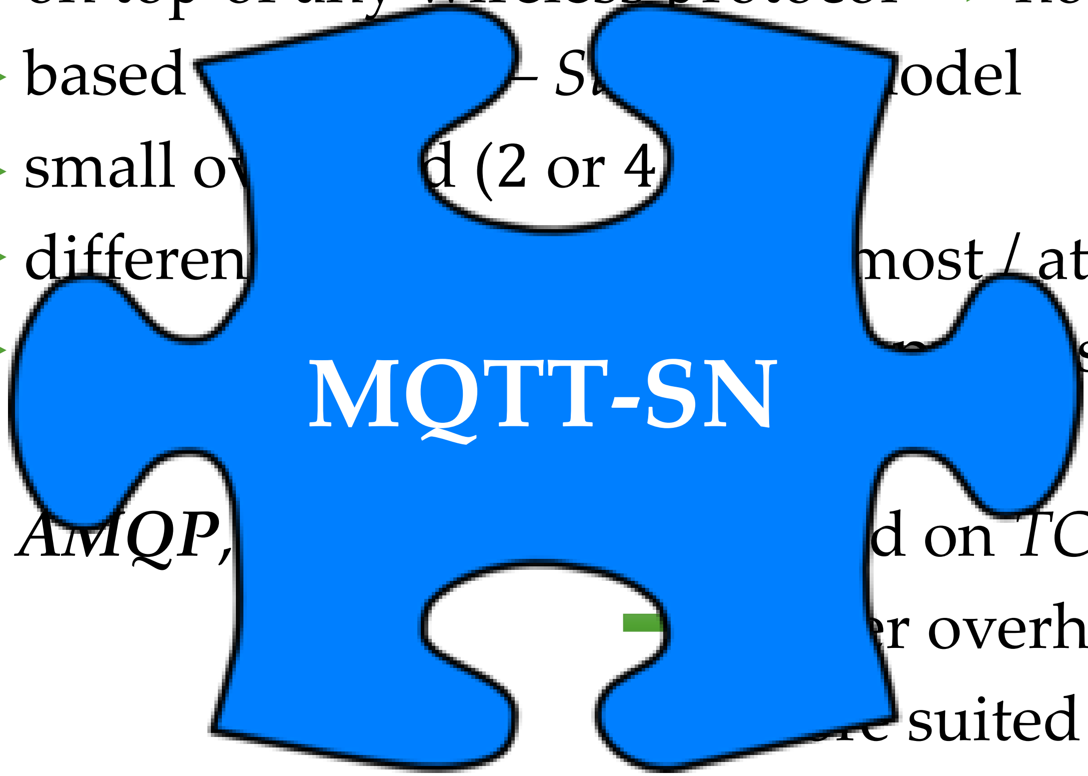
→ differen most / at least / exactly once

→ specific messages

CoAP, XMPP, AMQP, d on *TCP* or *UDP*

→ r overheads

suited for Gateway - Cloud



Security

- *CC*1350 → *128 bit AES (Advanced Encryption Standard)* hardware accelerator
  ↳ encryption/decryption and authentication → shared key
  → implementation of the *ECC (Elliptic Curve Cryptography)* core in *ROM*
  ↳ more *Flash memory* → application
  → *True Random Number Generator*

- *TI 15.4 Stack* → *128 bit AES* based encryption and authentication → shared keys
  ↳ part of the *IEEE 802.15.4e* security for the *MAC Layer*

- *Zigbee* → the same → plus more for the Application Support Sublayer
  ↳ larger overhead

Security

- *CC*1350 → *128 bit AES (Advanced Encryption Standard)* hardware accelerator
  - ↳ encryption/decryption and authentication → shared key
  - → implementati … ECC (… *urve Cryptography)* core in *ROM*
    - ↳ more *Fla* … *ory* → … tion
  - → *True Random N* …

128 bit AES encryption and authentication

- *TI 15.4 Stack* → *128 bit AE* … encry … d authentication → shared keys
  - ↳ p … he *IEEE* … *e* security for the *MAC Layer*
- *Zigbee* → the same → plus more for the Application Support Sublayer
  - ↳ larger overhead

Solution

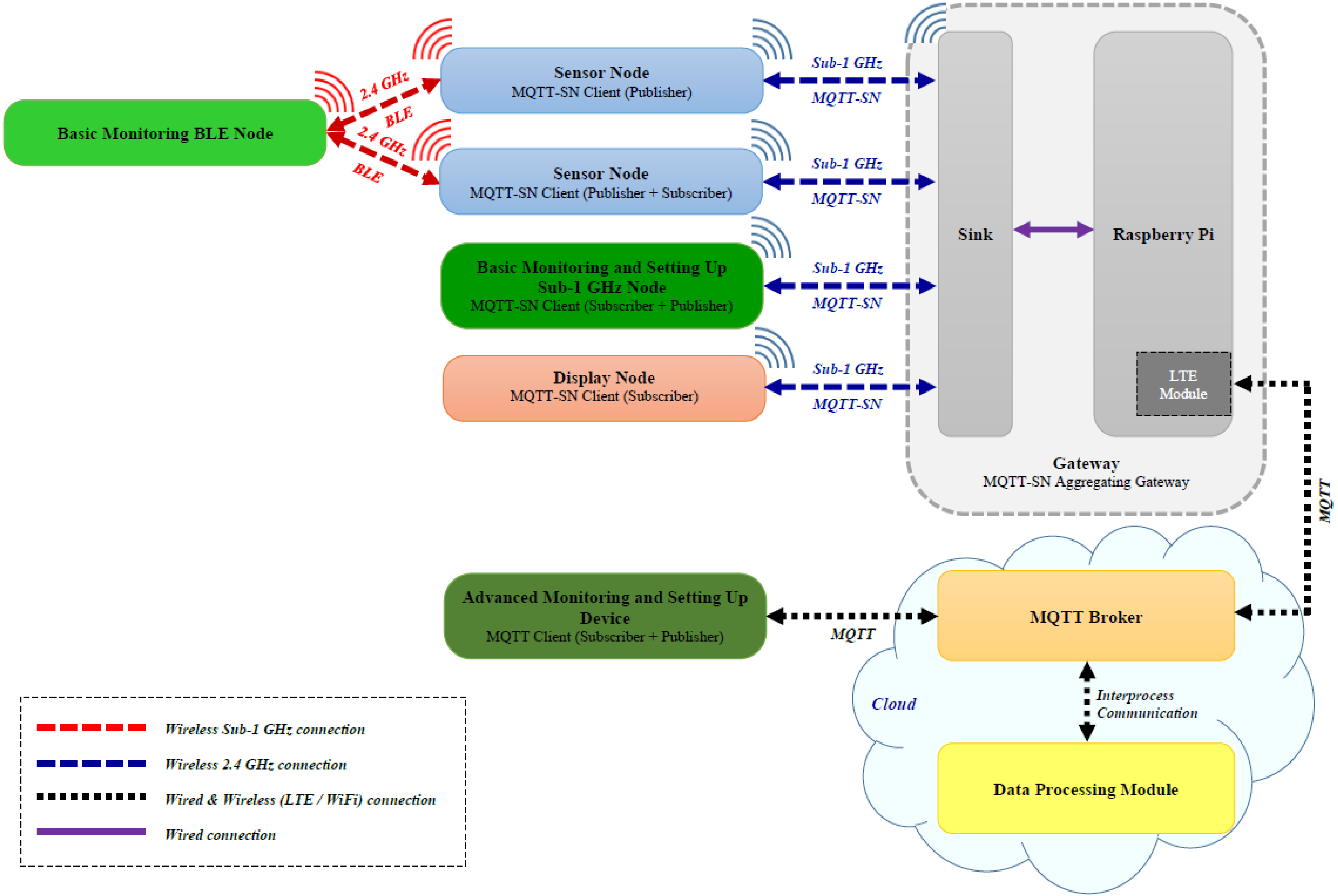
The Communication Protocol Stack

Layer	Sub-layer	Sub-1 GHz		2.4 GHz
Application Layer	*User Application*	• Sensor data acquisition, aggregation and sending • Response to queries • React to commands / settings messages • Caching on *Flash* memory and *SD card* • Switch between the Communication Protocol stacks • Defining, entering and leaving the power-saving modes		
	Framework for the User Application	***TI 15.4 Stack*** • Message frames for advertising, connecting, acknowledgements and plain – data	***MQTT – SN*** • *QoS* levels • Message frames for advertising, connecting, publishing, subscribing, acknowledgements, Pings, Wills • Persistent session	***BLE Core Stack 4.2 Version 2.3.2***
Network Layer		***TI 15.4 Stack*** • Star network topology • No routing • Starting the network • Managing the devices joining / leaving the network • No neighbours discovery • Presence advertisement through Beacons for the *PAN* coordinator (the Gateway) – in *Beacon Enabled Mode*		
Medium Access Control Layer		***IEEE 802.15.4 e/g*** • *AES-128 CCM* security (encryption and authentication) • *CSMA-CA (LBT + Backoff time + Interframe space)* • *Frequency Hopping* • Frame creation and synchronization	***Additional feature (has to be added)*** • Adaptability of *Frequency Hopping*	
Physical Layer		***IEEE 802.15.4 e/g*** • *863 – 870 MHz* frequency band • *34* frequency channels with *200 KHz* spacing • *14 dBm* transmission power • *2-GFSK* modulation • *50 Kbps* data rate		

Sub-1 GHz *2.4 GHz*

Solution

The Architecture of the whole system



Implementation and Evaluation

- Use case → highway construction site
  - → Sensor Nodes → on trucks → acquire information
  - → optimize the entire process

- Proprietary $CC1350$-based Sensor Nodes → designed within $DFKI$
- Gateway's Sink → $CC1350\ LaunchPad$

- A basic version → Communication Protocol Stack
  - → the same Physical layer
  - → reduced MAC, Network and Application layers
  - → $TI - RTOS$
- Modules for sensors' data acquisition → implemented

Implementation and Evaluation

Laboratory test

- 2 Sensor Nodes
- 1 Monitoring Gateway
- Different configurations (sending frequency, packet length, data rate, attempts, backoff time)
- Determine → average successfully received (per attempt), lost, duplicate packets

↳ Analysis of the results

↳ A packet not received in the first attempt → it was lost (the next → attempts → not useful)

↳ Split a big packet into multiple smaller → less complete batches are received than large packets

Implementation and Evaluation

Field test

- Highway construction site ➡ Spain

- 1 Sensor Node ➡ on a truck
- 1 Monitoring Gateway ➡ 20 *meters* away from the excavator
- Send/Receive ➡ IMU data (100 *Hz*), other sensors' data (1 *Hz*)
- Real scenario: HW test + Data acquisition + Range tests (2 *kilometers*)

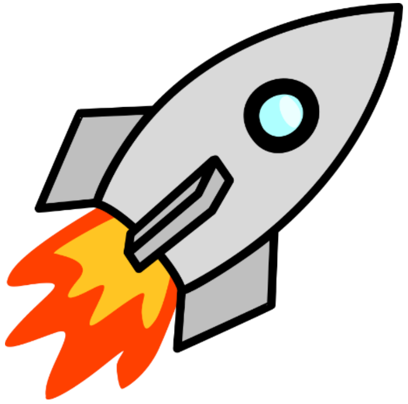
Conclusions

- The scope of this thesis → design a $Communication\ Protocol\ Stack$ → WSN based on the $CC1350$ microcontroller and $TI-RTOS$ → construction industry
- Define → requirements and specifications
- Analyze, compare, decide the most suited option
  - → $CC1350$ family
  - → IoT Standards and Protocols
  - → Application Layer Protocols
  - → Spectrum Acccess Methods
  - → Security solutions

- Design a Communication Protocol Stack
- Design the Architecture of the whole system
- Test a basic implementation → laboratory and on field
- Analyze the results

Conclusions – Future Work

- Implement & Test → all the features → designed *Communication Protocol Stack*
- Analysis → network's performances → different configurations
- Implement & Test → firmware → all the system's entities
  - → Low-Power
- Testing → different environments
- Consider → $CC1350P$ → from 2019
  - → larger Flash memory
  - → Dynamic Multi-Protocol Manager




THANK
YOU