Parameters	Parameters	Type	Expected as
Network conditions	network load	Dynamic	Minimized
	network coverage	Static	Fixed
	network connection time	Dynamic	Minimized
	available bandwidth	Dynamic	Minimized
Application requirements	throughput	Dynamic	Minimized
	delay	Dynamic	Minimized
	jitter	Dynamic	Minimized
	PLR	Dynamic	Minimized
	energy consumption	Dynamic	Minimized
User preferences	budget	Static	Fixed
	cost	Static	Fixed
	design		
Mobile equipment	energy	Dynamic	Fixed
	mobility	Dynamic	Fixed

Table 1: Network selection inputs and classification of parameters [1]

Application layer	Network layer	Sensing layer
Service time	Bandwidth	**Energy consumption**
Service availability	Packet loss	Sleep management
Service cost	Jitter	Life time management
Service reliability	Delay	**Coverage**
	Availability	Sensing area
		Information accuracy
		Data accuracy
		Sensing time accuracy
		Spatial accuracy
		Reduce data redundancy
		Data packaging
		Sampling rate
		Bit rate error

Table 2: QoS parameters [2] [3]

Network selection
Service selection
Gateway selection
Input:
Method: Ranking machine learning
Output: Ranked list of gateway

New York (NY)

Maximize	Minimize
(**T**) Throughput	(**RT**) Response Time
(**F**) Fairness	(**LT**) Latency
(**R**) Reliability	(**J**) Jitter
(**IA**) Information Accuracy	(**TF**) Traffic
(**Cov**) Coverage of IoT	(**AWT**) Average Waiting Time
(**NL**) Network Life	(**D**) Delay
(**RU**))Resource Utilization	(**L**) Load
	(**EC**) Energy Consumption
	(**BP**) Blocking Probability
	(**CCI**) Co-channel Interference
	(**SC**) Service Cost
	(**ST**) Service Time

Table 3: Objectives of IoT resource scheduling

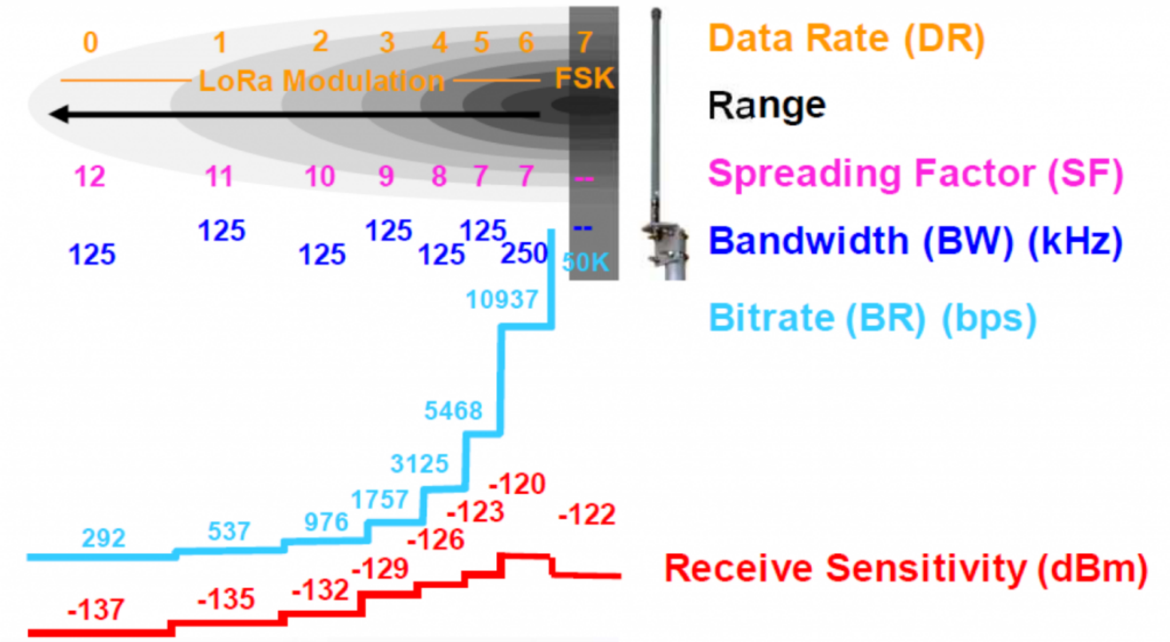

Figure 1: LoraWan Parameters.

Plan de controle	**Plan de gestion**	**Plan de doonées**
Controle d'admission	Controle et supervision de QoS	Controle du trafic
Réservation de ressources	Gestion de contrats	Façonnage du trafic
Routage	QoS mapping	Controle de congestion
Signalisation	Politique de QoS	Classification de paquets
		Marquage de paquets
		Ordonnancements des paquets
		Gestion de files d'attente

Table 4: An example table.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Length	FCF	DSN	DST PAN
Destination address			
Source address			

802.15.5 Header

I	O	S	D	Hop Limit
Source address				
Destination address				

Mesh addressing Header

(a) Sensors diversity. (b) Data diversity.

Figure 2: Sensors & data diversity.

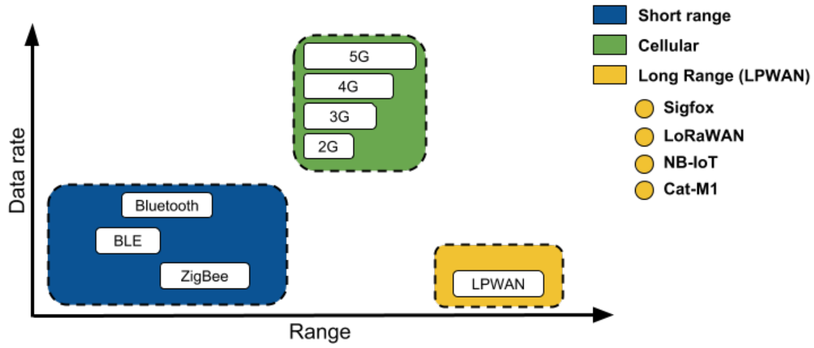

Figure 3: Communication diversity.

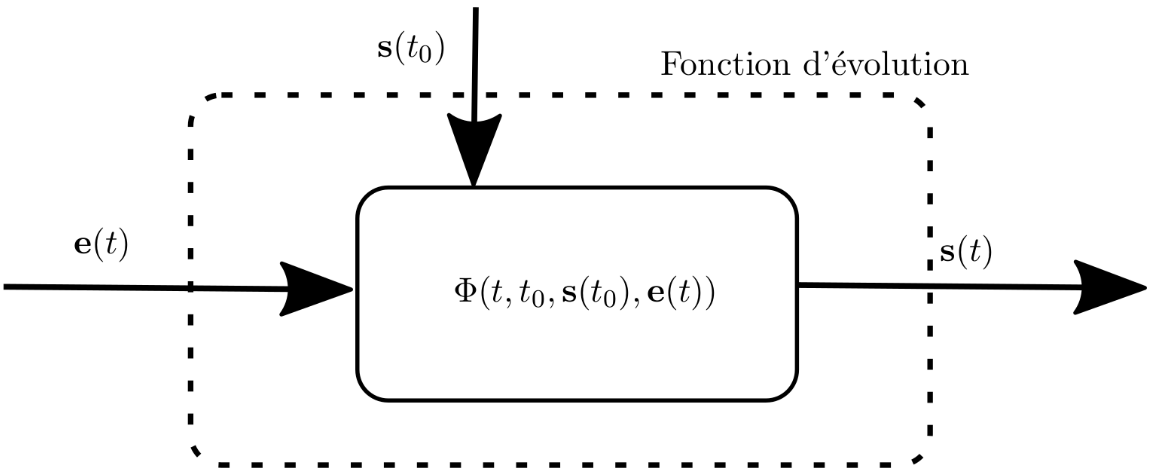

Figure 4: Filtres [4].

[7] Many studies have identified SDN as a potential solution to the WSN challenges, as well as a model for heterogeneous integration.

[7] This shortfall can be resolved by using the SDN approach.

[8] SDN also enhances better control of heterogeneous network infrastructures.

[8] Anadiotis et al. define a SDN operating system for IoT that integrates SDN based WSN **(SDN-WISE)**. This experiment shows how heterogeneity between different kinds of SDN networks can be achieved.

[8] In cellular networks, OpenRoads presents an approach of introducing SDN based heterogeneity in wireless networks for operators.

[9] There has been a plethora of (industrial) studies synergising SDN in IoT. The major

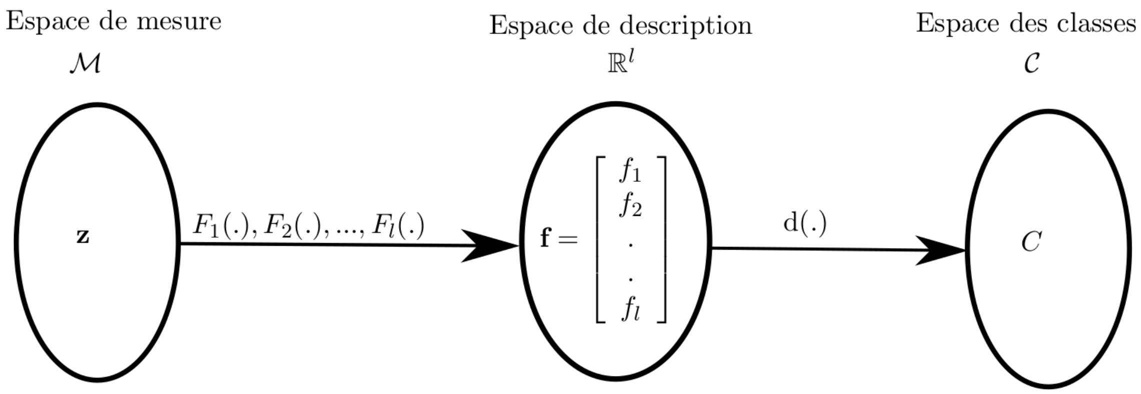

Figure 5: classification [4].

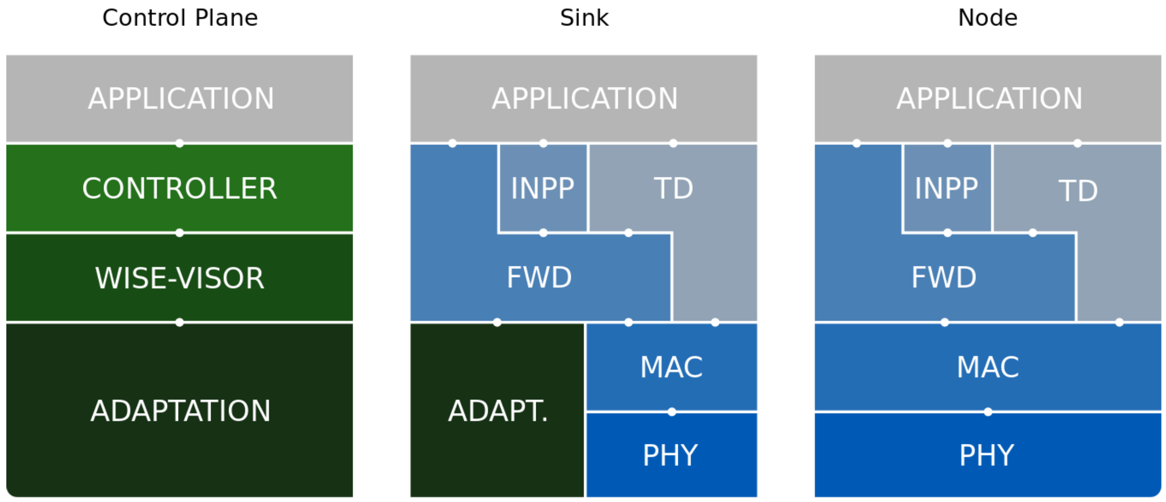

Figure 6: LPWAN connectivity.

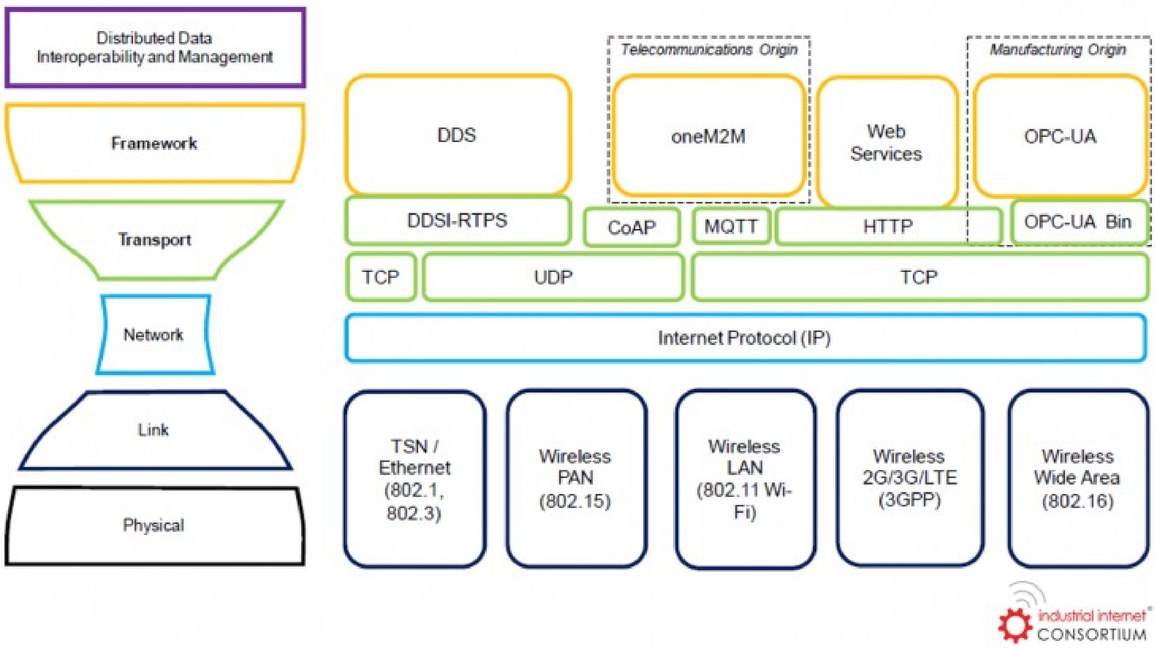

Figure 7: Interoperability.

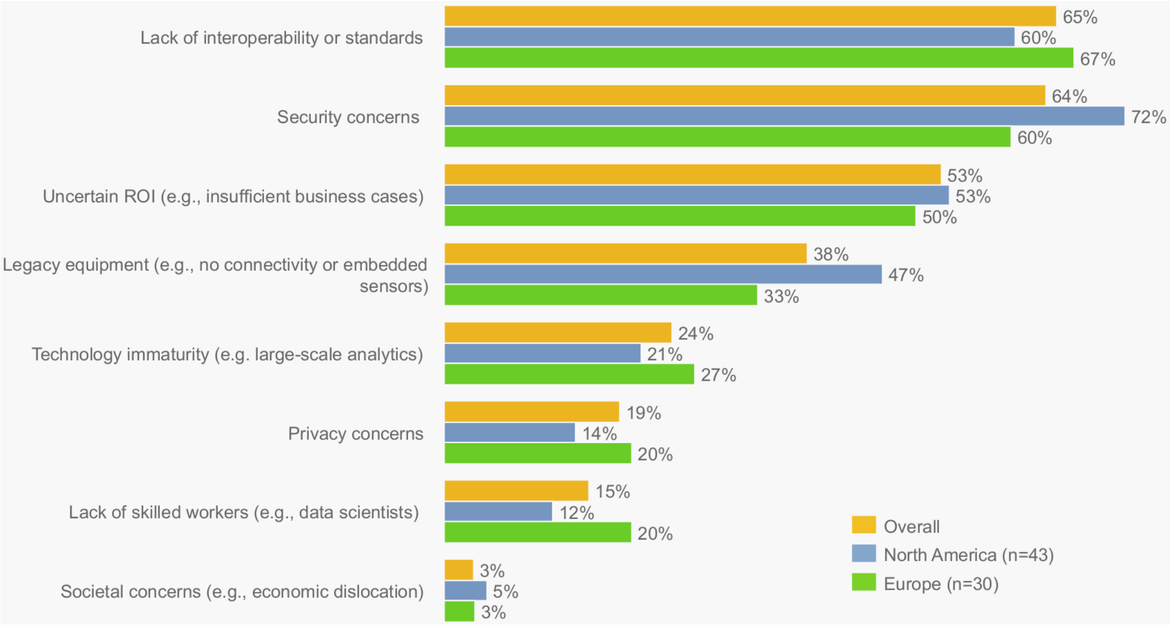

Figure 8: Key barriers in adopting the Industrial Internet[1].

Figure 9: wsn-IoT.

Naïve modes	Instantaneous Hist. average Clustering
Parametric models	Rarely used Traffic Models Time Series Linear regression ARIMA Kalman filtering ATHENA SETAR Gaussian Maximum Likelihood
Non-Parametric models	k-Nearest Neighbor Locally Weighted Regression Fuzzy Logic Bayes Network Neural Network Include temporal/spatial patterns

Table 5: Taxonomy of prediction models [6]

characteristics of IoT are low latency,wireless access, mobility and heterogeneity. [9] Thus a bottom-up approach application of SDN to the realisation of heterogeneous

IoT is suggested.

[9] Perhaps a more complete IoT architecture is proposed, where the authors apply SDN principles in IoT heterogeneous networks.

[10] it provides the SDWSN with a proper model of network management, especially considering the potential of heterogeneity in SDWSN.

[10] We conjecture that the SDN paradigm is a good candidate to solve the heterogeneity in IoT.

Management architecture	Management feature	Controller configuration	Traffic Control	Configuration and monitoring	Scapability and localization	Communication management
[11] **Sensor Open Flow**	SDN support protocol	Distributed	in/out-band	✓	✓	✓
[12] **SDWN**	Duty sycling, aggregation, routing	Centralized	in-band	✓		
[13] **SDN-WISE**	Programming simplicity and aggregation	Distributed	in-band		✓	
[degante_smart_2014] Smart	Efficiency in resource allocation	Distributed	in-band		✓	
SDCSN	Network reliability and QoS	Distributed	in-band		✓	
TinySDN	In-band-traffic control	Distributed	in-band		✓	
Virtual Overlay	Network flexibility	Distributed	in-band		✓	
Context based	Network scalability and performance	Distributed	in-band		✓	
CRLB	Node localization	Centralized	in-band			
Multi-hope	Traffic and energy control	Centralized	in-band			✓
Tiny-SDN	Network task measurement	-	in-band			

Table 6: SDN-based network and topology management architectures. [9]

Application	CoAP, MQTT			
Transport	UDP/TCP			
Network	IPv6 RPL	IPv4/IPv6		
	6LowPan	RFC 2464		RFC 5072
MAC	IEEE 802.15.4	IEEE 802.11 (Wi-Fi)	IEEE 802.3 (Ethernet)	2G, 3G, LTE
	2.4GHz, 915, 868MHz	2.4, 5GHz		
	DSS, FSK, OFDM	CSMA/CA	UTP, FO	

Table 7: An example table.

Application protocol	DDS	CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP
Service discovery		mDNS			DNS-SD		
Network layer				RPL			
Link layer				IEEE 802.15.4			
Physical layer		EPCglobal		IEEE 802.15.4		Z-Wave	

Table 8: Standardization efforts that support the IoT

	LiteOS	Nano-RK	MANTIS	Contiki
Architecture	Monolithic	Layered	Modular	Modular
Scheduling Memory	Round Robin	Monotonic harmonized	Priority classes	Interrupts execute w.r.t.
Network	File	Socket abstraction	At Kernel COMM layer	uIP, Rime
Virtualization and Completion	Synchronization primitives	Serialized access semaphores	Semaphores	Serialized, Access
Multi threading	✓	✓	✗	✓
Dynamic protection	✓	✗	✓	✓
Memory Stack	✓	✗	✗	✗

Table 9: Common operating systems used in IoT environment [14]

1 | Application

Smart systems in smart cities [15]

- Smart Mobility
- Smart semaphores controle
- Smart Red Swarm
- Smart panels
- Smart bus scheduling
- Smart EV management
- Smart surface parking
- Smart signs
- Smart energy systems
- Smart lighting
- Smart water jet systems
- Smart residuals gathering
- Smart building construction
- Smart tourism
- Smart QRinfo
- Smart monitoring
- Smart Hawkeye
- Network selection
  - MADM
    - Ranking methods
    - Ranking & weighted methods
  - Game theory
    - Users vs users
    - Users vs networks
    - Networks vs network
  - Fuzzy logic
    - as a score method
    - another theory
  - Utility function
    - 1

Application protocol	Rest-Full	Trans-port	Publish/Sub-scribe	Request/Re-sponse	Security	QoS	Header size (Byte)
COAP	✓	UDP	✓	✓	DTLS	✓	4
MQTT	✗	TCP	✓	✗	SSL	✓	2
MQTT-SN	✗	TCP	✓	✗	SSL	✓	2
XMPP	✗	TCP	✓	✓	SSL	✗	-
AMQP	✗	TCP	✓	✗	SSL	✓	8
DDS	✗	UDP TCP	✓	✗	SSL DTLS	✓	-
HTTP	✓	TCP	✗	✓	SSL	✗	-

Table 1.1: Application protocols comparison

Callenges-Applications	Gids	EHealth	Transportations	Cities	**Building**
Ressources cinstraints	+	+++	-	++	+
Mobility	+	++	+++	+++	-
Heterogeneity	++	++	++	+++	+
Scalability	+++	++	+++	+++	++
QoS cinstraints	++	++	+++	+++	+++
Data management	++	+	+++	+++	++
Lack of standardization	++	++	++	++	+++
Amount of attacks	+	+	+++	+++	+++
Safety	++	++	+++	++	+++

Table 1.2: Main IoT challenges[16]

Paper	$*$ 2 Architec-ture	Avail-ability	Relia-bility	Mo-bility	Perfor-mance	Manage-ment	Scala-bility	Interoper-ability	Secu-rity
IoT-A									
IoT@Work									
EBBITS									
BETaas									
CALIPSO									
VITAL									
SENSAI									
RERUM									
RELEYonIT									
IoT6									
OpenIoT									
Apec IoV									
Smart Santander									
OMA Device									
OMA-DM									
LWM2M									
NETCONF Light									
Kura									
MASH									
IoT-iCore									
PROBE-IT									
OpenIoT									
LinkSmart									
IETF SOLACE									
BUTLER									
Codo									
SVELETE									

Table 1.3: An example table.

Platform	**COAP**	**XMPP**	**MQTT**
Arkessa			✓
Axeda			
Etherios			
LittleBits			
NanoService	✓		
Nimbits		✓	
Ninja blocks			
OnePlateformv	✓	✓	
RealTime.io			
SensorCloud			
SmartThings			
TempoDB			
ThingWorx			✓
Xively			✓
Ubidots			✓

Table 1.4: IoT cloud platforms and their characteristics

Use cases			
Health Monitoring			
Water Distribution			
Electricity Distribution			
Smart Buildings			
Intelligent Transportation			
Surveillance			
Environmental Monitoring			

Table 1.5: Use cases [17]

Application protocol	Rest-Full	Trans-port	Publish/Sub-scribe	Request/Re-sponse	Security	QoS	Header size (Byte)
COAP	✓	UDP	✓	✓	DTLS	✓	4
MQTT	✗	TCP	✓	✗	SSL	✓	2
MQTT-SN	✗	TCP	✓	✗	SSL	✓	2
XMPP	✗	TCP	✓	✓	SSL	✗	-
AMQP	✗	TCP	✓	✗	SSL	✓	8
DDS	✗	UDP TCP	✓	✗	SSL DTLS	✓	-
HTTP	✓	TCP	✗	✓	SSL	✗	-

Table 1.6: Application protocols comparison

2 | Network

Routing protocol	Control Cost	Link Cost	Node Cost
OSPF/IS-IS	✗	✓	✗
OLSRv2	?	✓	✓
RIP	✓	?	✗
DSR	✓	✗	✗
RPL	✓	✓	✓

Table 2.1: Routing protocols comparison [**_rpl2_**]

- Routing over low-power and lossy links (ROLL)
- Support minimal routing requirements.
  - like multipoint-to-point, point-to-multipoint and point-to-point.
- A Destination Oriented Directed Acyclic Graph (DODAG)
  - Directed acyclic graph with a single root.
  - Each node is aware of ts parents
  - but not about related children
- RPL uses four types of control messages
  - DODAG Information Object (DIO)
  - Destination Advertisement Object (DAO)
  - DODAG Information Solicitation (DIS)
  - DAO Acknowledgment (DAO-ACk)
- Standard topologies to form IEEE 802.15.4e networks are

  Star contains at least one FFD and some RFDs

  Mesh contains a PAN coordinator and other nodes communicate with each other

  Cluster consists of a PAN coordinator, a cluster head and normal nodes.
- The IEEE 802.15.4e standard supports 2 types of network nodes

  FFD Full function device: serve as a coordinator
    * It is responsible for creation, control and maintenance of the net
    * It store a routing table in their memory and implement a full MAC

  RFD Reduced function devices: simple nodes with restricted resources
    * They can only communicate with a coordinator
    * They are limited to a star topology

Preamble	PHDR	PHDR CRC	MHDR	FHDR	FPort	Payload	MIC	CRC

Routing protocol	Control Cost	Link Cost	Node Cost
OSPF/IS-IS	✗	✓	✗
OLSRv2	?	✓	✓
RIP	✓	?	✗
DSR	✓	✗	✗
RPL	✓	✓	✓

Table 2.2: Routing protocols comparison [**_rpl2_**]

3 | MAC

Minimize	Maximize

Table 3.1

SF		07	08	09	10	11	12	07	08	09	10	11	12	07	08	09	10	11	12
	BW	125						250						500					
07	125	x								x								x	
08			x								x								x
09				x								x							
10					x								x						
11						x													
12							x												
07	250							x								x			
08									x								x		
09		x								x								x	
10			x								x								x
11				x								x							
12					x								x						
07	500													x					
08															x				
09								x								x			
10									x								x		
11		x								x								x	
12			x								x								x

Table 3.2: uyuyuy

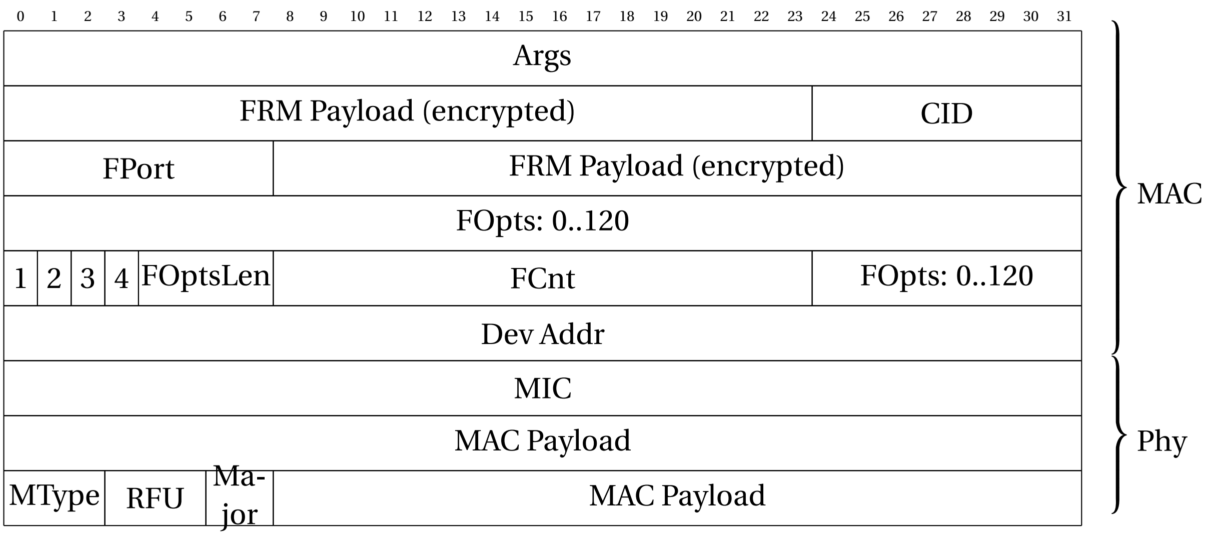

Figure 1: LoRaWAN frame format.[18]

- DevAddr: the short address of the device
- FPort: a multiplexing port field
  - 0: the payload contains only MAC commands
- FOptsLen:
- FCnt: frame counter
- MIC is a cryptographic message integrity code
  - computed over the fields MHDR, FHDR, FPort and the encrypted FRMPayload.
- MType is the message type (uplink or a downlink)
  - whether or not it is a confirmed message (reqst ack)
- Major is the LoRaWAN version; currently, only a value of zero is valid
- ADR and ADRAckReq control the data rate adaptation mechanism by the network server
- ACK acknowledges the last received frame
- FPending indicates that the network server has additional data to send
- FOptsLen is the length of the FOpts field in bytes
- FOpts is used to piggyback MAC commands on a data message
- CID is the MAC command identifier
- Args are the optional arguments of the commands
- FRMPayload is the payload, which is encrypted using AES with a key length of 128 bits

Characteristics	**6LoWPAN**	**LoRaWAN**	**SigFox**	**NB-IoT**
Standar body		LoRa Alliance		3GPP
TX power				
Modulation		CSS	BPSK	QSPSK
Frequency (MHz)	902-929 868-868.6	902-928 863-870 and 434	902 868	
Channels	0016 for 2400 0010 for 915 0001 for 868.3	80 for 915 10 for 868 and 780	25	
Bandwidth [MHz]	0005 for 2400 0002 for 915 0600 for 868.3	0.125 and 0.50 for 915 0.125 and 0.25 for 868 and 780	0.0001-0.0012	
Data rate (kbps)	0250 for 2400 0040 for 915 0020 for 868.3	0.00098-0.0219 for 915 0.250-0.05 for 868 and 780	0.1-0.6	
Modulation	QPSK for 2400 BPSK for 915 BPSK for 868.3	LoRa for 915 LoRa and GFSK for 868 and 780	BPSK and GFSK	
Coding (dBm)	-085 for 2400 -092 for 915 -092 for 868.3	-137	-137	
Coding	Direct	CSS	Ultra	
Coverage	10-100 m	5-15 km	10-50 km	
Battery lifetime	1-2 years	<10 years	<10 years	
Standard Body	IETF			
Security	ACL			
Uplink			100bps, 12 bytes/msg	
Downlink			8 bytes/msg	
Scalability				
Proprietary			✓	
Cost		High		

Table 3.3: LPWan Characteristics [19]

LoRa has three configurable parameters:

- Bandwidth (BW)
- Carrier Frequency (CF)
- Coding Rate (CR)
- Spreading Factor (SF)

	SIGFOX	LORAWAN	INGENU	TELENSA
Modulation	UNB DBPSK(UL), GFSK(DL)	CSS	RPMA-DSSS(UL), CDMA(DL)	UNB 2-FSK
Band	S UB -GH Z ISM:EU (868MHz), US(902MHz)	S UB -GH Z ISM:EU (433MHz 868MHz), US (915MHz), Asia (430MHz)	ISM 2.4GHz	S UB -GH Z bands including ISM:EU (868MHz), US (915MHz), Asia (430MHz)
Data rate	100 bps(UL), 600 bps(DL)	0.3-37.5 kbps (L O Ra), 50 kbps (FSK)	78kbps (UL), 19.5 kbps(DL)	62.5 bps(UL), 500 bps(DL)
Range	10 km (URBAN), 50 km (RURAL)	5 km(URBAN), 15 km (RURAL)	15 km (URBAN)	1 km (URBAN)
Num. of channels, orthogonal signals	360 channels	10 in EU, 64+8(UL) and 8(DL) in US plus multiple SFs	40 1MHz channels, up to 1200signals per channel	multiple channels
Link symmetry	✗	✓	✗	✗
Forward error correction	✗	✓	✓	✓
MAC	unslotted A LOHA	unslotted A LOHA	CDMA-like	
Topology	star	star, stars	star,tree	star
Adaptive Data Rate	✗	✓	✓	✗
Payload length	12B(UL), 8B(DL)	up to 250B (depends on SF and region)	10KB	
Handover	end devices do not join a single base station	end devices do not join a single base station		
Encryption	not supported	AES 128b	16B hash, AES 256b	

Table 3.4: [20]

Standard	**802.15.4k**	**802.15.4g**	**Weightless-W**	**Weightless-N**	**Weightless-P**	**DASH 7 Alliance**
Modulation	DSSS, FSK	MR-[FSK, OFDMA, OQPSK]	16-QAM, BPSK, QPSK, DBPSK	UNB DBPSK	GMSK, offset-QPSK	GFSK
Band	ISM S UB -GH Z, 2.4GHz	ISM S UB -GH Z, 2.4GHz	TV white spaces 470-790MHz	ISM S UB -GH Z EU (868MHz), US (915MHz)	S UB -GH Z ISM or licensed	UB -GH Z 433MHz, 868MHz, 915MHz
Data rate	1.5 bps-128 kbps	4.8 kbps-800 kbps	1 kbps-10 Mbps	30 kbps-100 kbps	200 bps-100kbps	9.6,55.6,166.7 kbps
Range	5 km (URBAN)	up to several kms	5 km (URBAN)	3 km (URBAN)	2 km (URBAN)	0-5 km (URBAN)
MAC	CSMA/CA, CSMA/CA or A LOHA with PCA	CSMA/CA	TDMA/FDMA	slotted A LOHA	TDMA/FDMA	CSMA/CA
Topology	star	tar, mesh, peer-to-peer	star	star	star	tree, star
Payload length	2047B	2047B	>10B	20B	>10B	256B
Encryption	AES 128b	AES 128b	AES 128b	AES 128b	AES 128/256b	AES 128b
Forward error correction	✓	✓	✓	✗	✓	✓

Table 3.5: [20]

[21] Nous avons vu en effet plus haut quil a été démontré que la méthode CSMA est plus efficace pour le traitement des faibles trafics, tandis que TDMA est nettement plus

Feature	Wi-Fi	802.11p	UMTS	LTE	LTE-A
Channel width MHz	20	10	5	1.4, 3, 5, 10, 15, 20	<100
Frequency band(s) GHz	2.4 , 5.2	5.86-5.92	0.7-2.6	0.7-2.69	0.45-4.99
Bit rate Mb/s	6-54	327	2	<300	<1000
Range km	<0.1	<1	<10	<30	<30
Capacity	Medium	Medium	✗	✓	✓
Coverage	Intermittent	Intermittent	Ubiquitous	Ubiquitous	Ubiquitous
Mobility support km/h	✗	Medium	✓	<350	<350
QoS support	EDCA Enhanced Distributed Channel Access	EDCA Enhanced Distributed Channel Access	QoS classes and bearer selection	QCI and bearer selection	QCI and bearer selection
Broadcast/multicast support	Native broadcast	Native broadcast	Through MBMS	Through eMBMS	Through eMBMS
V2I support	✓	✓	✓	✓	✓
V2V support	Native (ad hoc)	Native (ad hoc)	✗	✗	Through D2D
Market penetration	✓	✗	✓	✓	✓
Data rate	<640 kbps	250 kbps	106424 kbps	✓	✓

Table 3.6: An example table.

Phy protocol	IEEE 802.15.4	BLE	EPCglobal	Z-Wave	LTE-M	ZigBee
Standard Body		IEEE 802.15.1				IEEE 802.15.4, ZigBee Alliance
Radio band (MHz)	868/915/2400	2400	860-960	868/908/2400	700-900	
MAC address	TDMA, CSMA/CA	TDMA	ALOHA	CSMA/CA	OFDMA	
Data rate (bps)	20/40/250 K	1024K	varies 5-640K	40K	1G (up), 500M (down)	
Throughput				9.6, 40, 200kbps		
Scalability ???	65K nodes	5917 slaves	-	232 nodes	-	
Range	10-20m	10-100m				
Addressing	8|16bit	16bit				

Table 3.7: IoT cloud platforms and their characteristics [14]

	802.15.4	802.15.4e	802.15.4g	802.15.4f
Frequency	2.4Ghz (DSSS + oQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK, CSS+DQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK, CSS+DQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK,CSS+DQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK) 3~10Ghz (BPM+BPSK)
Data rate	Upto 250kbps	Upto 800kbps	Up to 800kbps	
Differences	-	Time sync and channel hopping	Phy Enhancements	Mac and Phy Enhancements
Frame Size	127 bytes	N/A	Up to 2047 bytes	N/A
Range	1 75+ m	1 75+ m	Upto 1km	N/A
Goals	General Low-power Sensing/Actuating	Industrial segments	Smart utilities	Active RFID
Products	Many	Few	Connode (6LoWPAN)	LeanTegra PowerMote

Table 3.8: IEEE 802.15.4 standards [**sarwar_iot_**]

appropriée pour supporter les trafics intensesj.

$$T_s = \frac{2^{SF}}{BW_{[Hz]}} \tag{3.1}$$

$$DR_{[bps]} = SF * \frac{BW_{[Hz]}}{2^{SF}} * CR \tag{3.2}$$

SF	Sensitivity[dBm]	Data Rate[kb/s]		
6	-118	9.38		
7	-123			
8	-126			
9	-129			
10	-132			
11	-134.5			
12	-137			

Table 3.9: hghg

$$n_s = 8 + max([\frac{8PL - 4SF + 8 + CRC + H}{4 * (SF - DE)}] * \frac{4}{CR}) \tag{3.3}$$

3.1 Lora modules

Ref	Module	Frequency MHz	Tx power	Rx power	Sensitivity	Channels	Distance
[22]	**Semtech SX1272**	863-870 (EU) 902-928 (US)	14 dBm	dBm	-134 dBm	8 13	22+ km
[22]	**rn2483**						
[22]	**rn2903**						
[22]	**rak811**						
[22]	**Semtech sx1276**						
[22]	**rfm95**						
[22]	**CMWX1ZZABZ-078**						
[22]	**LoPy4**						
[22]	**mDot**						
[22]	**xDot**						
[22]	**Laird RM192**						
[22]	**Laird RM186**						
[22]	**CMWX1ZZABZ-078**						
[22]	**Also Laird RM1xx**						
[22]	**iMST iM88x/iM98x**						
[22]	**Mic SAM RN34/35**						
[22]	**Semtech SX1278**						

Table 3.10

4 | Introduction

4.1 Introduction

4.1.1 Context & motivation

4.1.2 Methodology and contributions

4.1.3 Organization of the thesis

5 | State of the art [23]

5.1 Introduction

5.2 IoT Hardware and software platforms

5.2.1 Software platform: Operating systems

OS	Architecture	Multi threading	Scheduling	Dynamic Memory	Memory protection	Network Stack	Virtualization and Completion
Contiki/Contiki-ng	Modular	✓	Interrupts execute w.r.t.	✓	✗	uIP Rime	Serialized Access
MANTIS	Modular	✗	Priority classes	✓	✗	At Kernel COMM layer	Semaphores.
Nano-RK	Layered	✓	Monotonic harmonized	✗	✗	Socket abstraction	Serialized access semaphores
LiteOS	Monolithic	✓	Round Robin	✓	✓	File	Synchronization primitives

Table 5.1: Common operating systems used in IoT environment [14]

Contiki

RIOT

TinyOS

freeRTOS

5.2.2 Hardware platform

OpenMote

MSB430-H

Zolertia

5.2.3 Communication protocol

IEEE 802.15.4

6LoWPAN

ZigBee

Bluetooth LE

LoaraWAN

SEMTECH

ALIANCE

Class-A

Uplink

Downlink

Confirmed data

Note:

Class-B

Downlink

Confirmed data

Requirements

Device

Gateway

Class-C

Downlink

Confirmed data

5.2.4 Application protocol

CoAP

- Constrained Application Protocol
- The IETF Constrained RESTful Environments
- CoAP is bound to UDP
- CoAP can be divided into two sub-layers
  - messaging sub-layer
  - request/response sub-layer
    a) Confirmable.
    b) Non-confirmable.
    c) Piggybacked responses.
    d) Separate response
- CoAP, as in HTTP, uses methods such as:

- ✳ GET, PUT, POST and DELETE to
- ✳ Achieve, Create, Retrieve, Update and Delete
- ✳ Ex: the GET method can be used by a server to inquire the clients temperature

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Ver	T	TKL	Code	Message ID
Token				
Options				
11111111			Payload	

CoAP Header

Ver: is the version of CoAP
T: is the type of Transaction
TKL: Token length
Code: represents the request method (1-10) or response code (40-255).
- ✳ Ex: the code for GET, POST, PUT, and DELETE is 1, 2, 3, and 4, respectively.

Message ID: is a unique identifier for matching the response.
Token: Optional response matching token.

MQTT

- ➠ Message Queue Telemetry Transport
- ➠ Andy Stanford-Clark of IBM and Arlen Nipper of Arcom
  - ✳ Standardized in 2013 at OASIS
- ➠ MQTT uses the publish/subscribe pattern to provide transition flexibility and simplicity of implementation
- ➠ MQTT is built on top of the TCP protocol
- ➠ MQTT delivers messages through three levels of QoS
- ➠ Specifications
  - ✳ MQTT v3.1 and MQTT-SN (MQTT-S or V1.2)
  - ✳ MQTT v3.1 adds broker support for indexing topic names
- ➠ The publisher acts as a generator of interesting data.

0 1 2 3 4 5 6 7

Message Type	UDP	QoS Level	Retain
Remaining length			
Variable length header			
Variable length message payload			

CoAP Header

Message type: CONNECT (1), CONNACK (2), PUBLISH (3), SUBSCRIBE (8) and so on
DUP flag: indicates that the massage is duplicated
QoS Level: identify the three levels of QoS for delivery assurance of Publish messages
Retain field: retain the last received Publish message and submit it to new subscribers as a first message

XMPP

- ➠ Extensible Messaging and Presence Protocol
- ➠ Developed by the Jabber open source community

- An IETF instant messaging standard used for:
  - multi-party chatting, voice and telepresence
- Connects a client to a server using a XML stanzas
- An XML stanza is divided into 3 components:
  - message: fills the subject and body fields
  - presence: notifies customers of status updates
  - iq (info/query): pairs message senders and receivers
- Message stanzas identify:
  - the source (from) and destination (to) addresses
  - types, and IDs of XMPP entities

AMQP

- Advanced Message Queuing Protocol
- Communications are handled by two main components
  - exchanges: route the messages to appropriate queues.
  - message queues: Messages can be stored in message queues and then be sent to receivers
- It also supports the publish/subscribe communications.
- It defines a layer of messaging on top of its transport layer.
- AMQP defines two types of messages
  - bare massages: supplied by the sender
  - annotated messages: seen at the receiver
- The header in this format conveys the delivery parameters:
  - durability, priority, time to live, first acquirer & delivery count.
- AMQP frame format

  Size the frame size.

  DOFF the position of the body inside the frame.

  Type the format and purpose of the frame.
  - Ex: 0x00 show that the frame is an AMQP frame
  - Ex: 0x01 represents a SASL frame.

DDS

- Data Distribution Service
- Developed by Object Management Group (OMG)
- Supports 23 QoS policies:
  - like security, urgency, priority, durability, reliability, etc
- Relies on a broker-less architecture
  - uses multicasting to bring excellent Quality of Service
  - real-time constraints
- DDS architecture defines two layers:

  DLRL Data-Local Reconstruction Layer
  - serves as the interface to the DCPS functionalities

  DCPS Data-Centric Publish/Subscribe
  - delivering the information to the subscribers
- 5 entities are involved with the data flow in the DCPS layer:
  - Publisher:disseminates data
  - DataWriter: used by app to interact with the publisher
  - Subscriber: receives published data and delivers them to app

    - DataReader: employed by Subscriber to access received data
    - Topic: relate DataWriters to DataReaders
- No need for manual reconfiguration or extra administration
- It is able to run without infrastructure
- It is able to continue working if failure happens.
- It inquires names by sending an IP multicast message to all the nodes in the local domain
    - Clients asks devices that have the given name to reply back
    - the target machine receives its name and multicasts its IP @
    - Devices update their cache with the given name and IP @

mDNS

- Requires zero configuration aids to connect machine
- It uses mDNS to send DNS packets to specific multicast addresses through UDP
- There are two main steps to process Service Discovery:
    - finding host names of required services such as printers
    - pairing IP addresses with their host names using mDNS
- Advantages
    - IoT needs an architecture without dependency on a configuration mechanism
    - smart devices can join the platform or leave it without affecting the behavior of the whole system
- Drawbacks
    - Need for caching DNS entries

5.2.5 Summary and discussion

5.3 IoT applications

5.3.1 Transportation and logistics

5.3.2 Healthcare

5.3.3 Smart environnement

5.3.4 personal and social

5.3.5 Futuristic

5.3.6 Summary and discussion

5.3.7 Summary and discussion

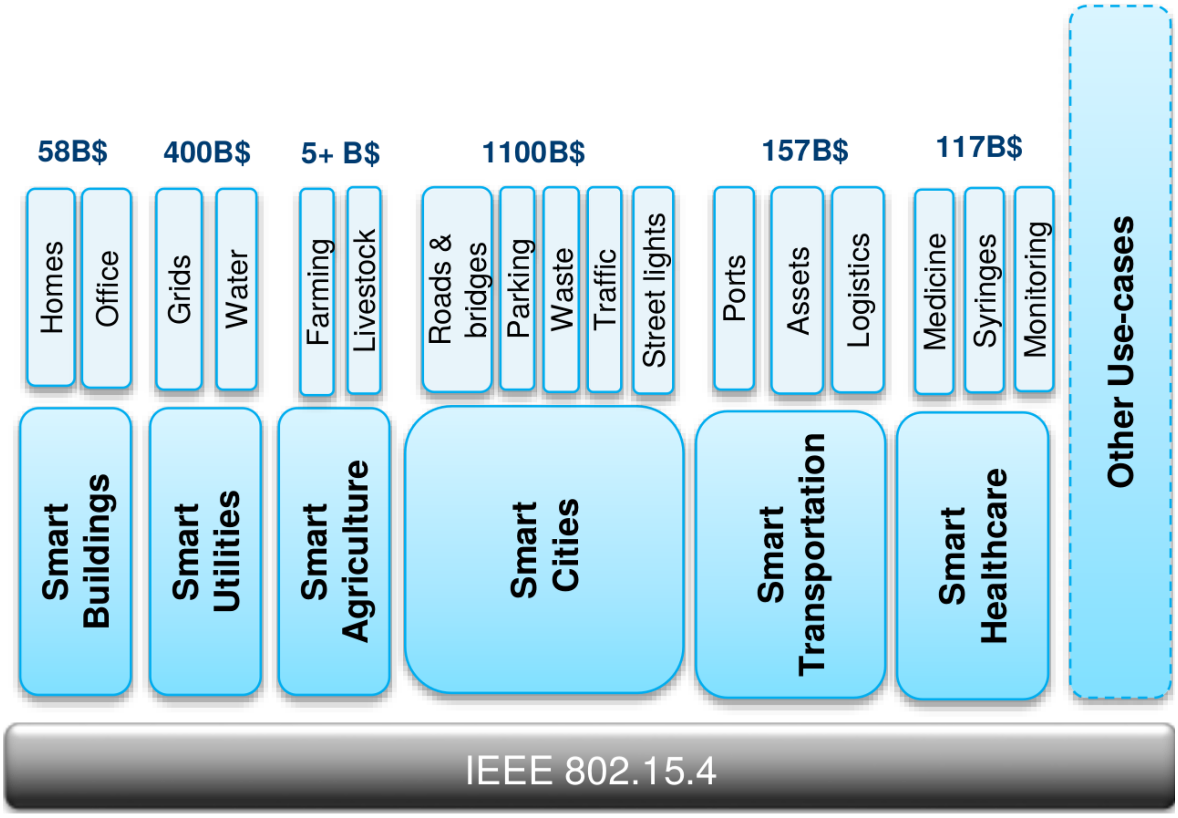

Figure 1: 802.15.4 use cases [**sarwar_iot_**].

5.4 IoT security

5.4.1 Summary and discussion

5.5 Conclusion

6 | Aghiles [23]

6.1 Introduction & problem statement

6.1.1 Background

6.1.2 Purpose (Goal)

6.1.3 Limitations

6.1.4 Method

6.2 Background

6.2.1 Requirements

Hardware

Operating system

Communication protocol

6.2.2 Hardware

6.2.3 Operating system

6.2.4 Communication protocol

6.2.5 Workspace and tools

6.3 Prototype

6.3.1 Drivers and firmware

6.3.2 CoAP server

Testing

Final prototype

6.4 Evaluation

6.4.1 Range

6.4.2 Response time

6.4.3 Connection speed

6.4.4 Power consumption

6.5 Discussion

6.5.1 The prototype

Results

Range

Response time

Connection speed

Power consumption

6.5.2 Project execution

6.6 Conclusion

7 | SDN: Sentilo [24]

7.1 Introduction & problem statement

7.1.1 Background

7.1.2 Purpose (Goal)

7.1.3 Limitations

7.1.4 Method

7.2 Background

7.2.1 Requirements

7.2.2 Hardware: Zolertia Z1 Motes

Peripherals ports

North Port

East Port

South Port

West Port

Internal sensors

Temperature Sensor

Accelerometer

External Sensors

Analog sensors

Precision Light Sensor

Force Sensor

Relay actuator

Distance sensor

7.2.3 Operating systems

Main aspects

Contiki size

Contiki Hardware

Kernel structure

7.2.4 Communication protocol

Composition

Physical and MAC Layer (IEEE 802.15.4)

Physical Layer

Definitions

Topologies

RIME

6LowPAN

Characteristics

Encapsulation Header format

Fragment Header

Mesh addressing header

Header compression (RFC4944)

Header compression Improved (draft-hui-6lowpan-hc-01)

RPL

7.2.5 Application protocol

COAP (COnstrained Application Protocol)

Overview

Coap Methods

Coap Transactions

Coap Messages

7.2.6 Workspace ant tools

7.3 Sentilo

7.3.1 Definitions

7.3.2 Sentilo Architecture

PubSub Server

Web Catalog Application

Extensions (Agents)

7.3.3 Sentilo structure

7.3.4 Sentilo API

7.4 Evaluation

7.4.1 Environment description

Sensor Network

Border Router

Nodes

Network connector

Application workflow

Sensor registration

Sensor data publish

7.5 Discussion

7.5.1 The prototype

Results

Range

Response time

Connection speed

Power consumption

7.5.2 Project execution

7.6 Conclusion

7.6.1 Future lines of work

8 | MEC: Chapter 4

Contents

8.1 Introduction & problem statement

8.1.1 Background

8.1.2 Purpose (Goal)

8.1.3 Limitations

8.1.4 Method

8.2 Background

8.2.1 Selection of technology

Requirements

Hardware

Operating system

Communication protocol

Hardware

Operating system

Communication protocol

Workspace and tools

8.3 Prototype

8.3.1 Drivers and firmware

8.3.2 CoAP server

Testing

Final prototype

8.4 Evaluation

8.4.1 Environment description

8.4.2 Results exploitation

8.4.3 Range

8.4.4 Response time

8.4.5 Connection speed

8.4.6 Power consumption

8.5 Discussion

8.5.1 The prototype

Results

Range

Response time

Connection speed

Power consumption

8.5.2 Project execution

8.6 Conclusion

9 | Conclusion

Contents

9.1 Conclusion

9.2 Perspectives

10 | Publications

Contents

10.1 List of publications

A | Appendix A

Contents

A.1 Introduction & problem statement

A.2 Background

A.3 Approach

A.4 Performance evaluation

A.4.1 Environment description

A.4.2 Results exploitation

A.5 Conclusion

B | Appendix B

Contents

Year		**Factors**	**Computation Model**	**Results interpretation**
2018	[25]	-Closeness Centralityjhjhjhjhjhjh -Degree Centrality	Estimation	**Closeness** have a high degree of nbnbnbnbnbnbnb correlation with **privacy score**

Table B.1: An example table.

Bibliography

Others

[1] Fayssal Bendaoud, Marwen Abdennebi, and Fedoua Didi. "Network Selection in Wireless Heterogeneous Networks: A Survey". In: *Journal of Telecommunications and Information Technology* 4 (Jan. 2019). 00000, pp. 64–74 (p. 1).

[2] Atefeh Meshinchi. "QOS-Aware and Status-Aware Adaptive Resource Allocation Framework in SDN-Based IOT Middleware". 00000. masters. École Polytechnique de Montréal, May 2018 (p. 1).

[3] Abishi Chowdhury and Shital A. Raut. "A Survey Study on Internet of Things Resource Management". In: *Journal of Network and Computer Applications* 120 (Oct. 15, 2018). 00002, pp. 42–60 (p. 1).

[4] Pierre Merdrignac. "Système Coopératif de Perception et de Communication Pour La Protection Des Usagers Vulnérables". In: (2015). 00003, p. 253 (p. 3, 4).

[5] Industrial Internet of Things. *Executive Summary*. 01455. URL: `http://wef.ch/1Ce8qay` (visited on 04/17/2019).

[6] *Short Term Traffic Prediction Models*. 00000. 2007 (p. 5).

[7] Zhijing Qin et al. "A Software Defined Networking Architecture for the Internet-of-Things". In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. NOMS 2014 - 2014 IEEE/IFIP Network Operations and Management Symposium. 00258. Krakow, Poland: May 2014, pp. 1–9 (p. 3).

[8] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke. "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements". In: *IEEE Access* 5 (2017). 00135, pp. 1872–1899 (p. 3).

[9] Musa Ndiaye, Gerhard Hancke, and Adnan Abu-Mahfouz. "Software Defined Networking for Improved Wireless Sensor Network Management: A Survey". In: 17.5 (May 4, 2017). 00053, p. 1031 (p. 3, 5, 6).

[10] Samaresh Bera, Sudip Misra, and Athanasios V. Vasilakos. "Software-Defined Networking for Internet of Things: A Survey". In: *IEEE Internet of Things Journal* 4.6 (Dec. 2017). 00057, pp. 1994–2008 (p. 6).

[11] Tie Luo, Hwee-Pink Tan, and Tony Q. S. Quek. "Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks". In: *IEEE Communications Letters* 16.11 (Nov. 2012). 00341, pp. 1896–1899 (p. 6).

[12] Salvatore Costanzo et al. "Software Defined Wireless Networks (SDWN): Unbridling SDNs". In: (2012). 00181, p. 25 (p. 6).

[13] Laura Galluccio et al. "SDN-WISE: Design, Prototyping and Experimentation of a Stateful SDN Solution for WIreless SEnsor Networks". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE INFOCOM 2015 - IEEE Conference on Computer Communications. 00173. Kowloon, Hong Kong: Apr. 2015, pp. 513–521 (p. 6).

[14] Ala Al-Fuqaha et al. “ Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications ”. In: *IEEE Communications Surveys & Tutorials* 17.4 (24–2015). 02482, pp. 2347–2376 (p. 7, 18, 23).

[15] Enrique Alba. “ Intelligent Systems for Smart Cities ”. In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*. The 2016. 00004. Denver, Colorado, USA: ACM Press, 2016, pp. 823–839 (p. 9).

[16] Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, and Hicham Lakhlef. “ Internet of Things Security: A Top-down Survey ”. In: *Computer Networks* 141 (Aug. 4, 2018). 00029, pp. 199–221 (p. 10).

[17] Gerhard Hancke, Bruno Silva, and Gerhard Hancke Jr. “ The Role of Advanced Sensing in Smart Cities ”. In: 13.1 (Dec. 27, 2012). 00318, pp. 393–425 (p. 11).

[18] Aloßs Augustin et al. “ A Study of LoRa: Long Range & Low Power Networks for the Internet of Things ”. In: 16.9 (Sept. 9, 2016). 00000, p. 1466 (p. 15).

[19] H. A. A. Al-Kashoash and Andrew H. Kemp. “ Comparison of 6LoWPAN and LPWAN for the Internet of Things ”. In: *Australian Journal of Electrical and Electronics Engineering* 13.4 (Oct. 2016). 00010, pp. 268–274 (p. 16).

[20] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. “ Low Power Wide Area Networks: An Overview ”. In: *IEEE Communications Surveys & Tutorials* 19.2 (22–2017). 00000, pp. 855–873 (p. 17).

[21] *Évaluation et Amélioration Des Plates-Formes Logicielles Pour Réseaux de Capteurs sans-Fil, Pour Optimiser La Qualité de Service et l'énergie*. 00000. URL: `http://docnum.univ-lorraine.fr/public/DDOC_T_2016_0051_ROUSSEL.pdf` (visited on 04/17/2019) (p. 17).

[22] *Waspmote Lora 868mhz 915mhz Sx1272*. 00000 (p. 19).

[23] Johan Bregell. “ Hardware and Software Platform for Internet of Things ”. In: *Master of Science Thesis in Embedded Electronic System Design* (2015). 00002 (p. 23, 29).

[24] *Contiki Applications for Z1 Motes for 6LowPAN*. 00000. 2016. URL: `https://upcommons.upc.edu/bitstream/handle/2117/82767/Master%20Thesis.pdf?sequence=1&isAllowed=y` (visited on 01/14/2019) (p. 31).

[25] Pascal Thubert, Maria Rita Palattella, and Thomas Engel. “ 6TiSCH Centralized Scheduling: When SDN Meet IoT ”. In: *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2015 IEEE Conference on Standards for Communications and Networking (CSCN). 00033. Tokyo, Japan: Oct. 2015, pp. 42–47 (p. 45).