UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione

Curriculum: Telecommunications and Telematics

Resilient IoT Systems – Issues and Solutions

Candidate
Adnan Rashid

Supervisors
Prof. Tommaso Pecorella

Prof. Francesco Chiti

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXIV, 2018-2021

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information Engineering. Copyright © 2022 by Adnan Rashid.

“Indeed, mankind is in loss, Except for those who have believed and done righteous deeds and advised each other to truth and advised each other to patience.”

Al-Quran

Acknowledgments

First of all, I am thankful to Almighty Allah for giving me the opportunity, strength, and ability to understand, learn and complete this thesis. Without His numerous blessings, it would not have been possible.

I would like to acknowledge the efforts and input of my supervisor, Prof. Tommaso Pecorella who cared for me a lot during the COVID-19 pandemic, I have no words to explain his friendliness, generosity, and kindness. Besides this, he is a real researcher and I have learned a lot from him. I would like to thanks my all colleagues of the Department of Information Lab (DINFO LAB) who were of great help during my research. In particular, my thanks go to Romano Fantacci, Francesco Chiti, Francesco Grasso, Benedetta Picano, Pascal Thubert (Sophia Antipolis, France), Nalini Elkins (Inside Products, USA), Michael S. Ackermann (Michigan, USA), Ameya Deshpande (Pune, India) who collaborated and support on the main parts of my research work. I would like to thank also to Roberto Picchi, Francesco Ermini, Michele Bonanni, and Csanyi Aliz for their kind support and hospitality during my stay in Florence, Italy.

Last but not least, I would like to acknowledge with gratitude, the support and love of my parents and family members, especially my son Hasin Adnan and my wife Madeeha Farooq. May Allah bless my parents, family, friends, colleagues, and professors.

Abstract

The Internet of Thing (IoT) has been one of the main focus areas of the research community in recent years, their peculiar requirements help network administrators to design and ensure the functionalities and resources of each device. Generally, two types of devices—constrained and unconstrained devices—are typical in the IoT environment. Devices with limited resources—for example, sensors and actuators—are known as constrained devices. The unconstrained devices include gateways or border routers. Such devices are challenging in terms of their deployment because of their connectivity, channel selection, multiple interfaces, local and global address assignment, address resolution, remote access, mobility, routing, border router scope, and security. To deal with these peculiar services, the availability of the IoT system ensures that the desired network services are available even in the presence of denial-of-service attacks, and the use of the system has become a difficult but mandatory task for network designers. To this end, I present a novel design for Wireless Sensor Networks (WSNs) which is the subsystem of IoTs, to address these challenges by shifting mandatory functionalities from unreliable to reliable and stable domains.

Moreover, energy conservation is another aspect that is one of the main constraints and the traditional IPv6 Neighbor Discovery (IPv6-ND) is not designed nor suitable to cope with it. In spite of that, non-transitive wireless links and the use of heavy multicast transmission make it inefficient and sometimes impractical in a Low-Power and Lossy Network (LLN). Due to these peculiarities a significant work has been done by the Internet Engineering Task Force (IETF) to optimize IPv6-ND, known as IPv6 over Low power Wireless Personal Area Network - Neighbor Discovery (6LoWPAN-ND). The implementation of the 6LoWPAN-ND protocol in mesh-under works totally opposite to its main purpose, because it reduces the multicast transmission but increases the unicast transmission in a drastic way. On the other hand IPv6-ND works in a reactive way but the network resilience in terms of reliability and robustness becomes questionable. Obtained results prove to answer a few questions. For example, is there a need for 6LoWPAN-ND protocol for a given LLN or not? What would be the benefits or drawbacks if we utilize it? What will happen if we are not interested to adopt this protocol for LLNs and keep using IPv6-ND protocol? All these questions addresses in terms of IoT resiliency.

Another aspect is the availability of the application services and user

privacy in IoT systems. Due to the drastic increase of IoT devices, increasing demand for application services with a strict Quality of Service (QoS) requirements. Therefore, service providers are dealing with the functional integration of the classical cloud computing architecture with edge computing networks. However, considering the limited capacity of the edge nodes requires a proper virtual functions allotment to advance the user satisfaction and service perfection. However, demand prediction is crucial but essential in services management. High variability of application requests that result in inaccurate forecasts become a big challenge. The Federated learning methods provide a solution to train mathematical learning models at the end-user sites. Network functions virtualization leverages the IT virtualization technologies to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create and deliver communication services. To preserve the data security and maximize service provider revenue, I use the federated learning approach for the prediction of virtual functions demand in Internet of Everything (IoE) based edge-cloud computing systems. Additionally, my work proposes a matching-based tasks allocation with some numerical results that validate the proposed approach by comparing it with a chaos theory prediction scheme.

The services offered through IoT systems, much like any system on the Internet, must not only be studied and improved, they must be continuously monitored to ensure security and resilience. It is important to know what kind of services they provide, how they evolve, and what is the network performance? One of the most promising ways to enable continuous QoE monitor is to use a novel IPv6 extension header called Performance and Diagnostic Metrics (PDM) Destination Option header, defined in RFC8250. This IETF standard defines an optional header that is included in each packet to offer sequence numbers and timing information for measurement purposes. These measurements can be analyzed in real-time or later. Currently, PDM data is provided in clear-text so malicious actors may be able to gather information for future assaults. The standard proposal, which is still being worked on, uses a lightweight handshake (registration procedure) and encryption to safeguard data. It also includes a list of additional performance measures that might be useful for further performance evaluation of IoT systems. My proposal uses the Internet Research Task Force (IRTF) Hybrid Public Key Encryption (HPKE) framework [23] to provide confidentiality and integrity to PDM data and is currently the candidate system to secure both PDMv2 [46] and Messaging Layer Security (MLS) [22].

Acronyms

IoT Internet of Thing

IIoT Industrial Internet of Things

IoE Internet of Everything

IEEE Institute of Electrical and Electronic Engineers

IETF Internet Engineering Task Force

ITU International Telecommunication Union

TLS Transport Layer Security

IPsec Internet Protocol Security

HTTP Hypertext Transfer Protocol

V2X Vehicle-to-everything

VDR Virtual DoDAG Root

TCP Transmission Control Protocol

SPoF Single Point of Failure

SDN Software-Defined Networking

NFV Network Function Virtualization

6LBR 6LoWPAN Border Router

6LN 6LoWPAN Node

6LoWPAN-ND IPv6 over Low power Wireless Personal Area Network - Neighbor Discovery

LoWPAN Low power Wireless Personal Area Network

6LoWPAN IPv6 over Low power Wireless Personal Area Network

PDM Performance and Diagnostic Metrics

PSDU PHY Service Data Unit

MLSN Multi-Link Subnets

IPv6-ND IPv6-Neighbor Discovery

EP Egress Point

6LR 6LoWPAN Router

ACL Access Control List

DAC Duplicate Address Confirmation

DAR Duplicate Address Request

P2P point-to-point

MP2P multipoint-to-point

P2MP point-to-multipoint

IID Interface Identifier

IRTF Internet Research Task Force

CFRG Crypto Forum Research Group

DAD Duplicate Address Detection

EDAR Extended Duplicate Address Registration

EDAC Extended Duplicate Address Confirmation

DAG Directed Acyclic Graph

DHCP Dynamic Host Configuration Protocol

DHCPv6 Dynamic Host Configuration Protocol version 6

DIO DODAG Information Object

DIS DODAG Information Solicitation

DAO DODAG Advertisement Object

DODAG Destination-Oriented Directed Acyclic Graph

DoS Denial-of-Service

DDoS Distributed Denial-of-Service

SSL Secure Sockets Layer

CPU Central Processing Unit

DTLS Datagram Transport Layer Security

UDP User Datagram Protocol

SSH Secure Socket Layer

P2MP point-to-multipoint

LoRaWAN Long Range Wide Area Network

WiFi Wireless Fidelity

EUI-64 Extended Unique Identifier - 64 bits

FFD Full-Function Device

CAP Contention Access Period

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

FCS Frame Check Sequence

CRC Cyclic Redundancy Check

ICMPv6 Internet Control Message Protocol version 6

ICMP Internet Control Message Protocol

6lo IPv6 over Networks of Resource-constrained Nodes

ROLL Routing Over Low power and Lossy networks

OSI Open Systems Interconnection

IP Internet Protocol

IPv6 Internet Protocol version 6

IPv4 Internet Protocol version 4

IPv6-ND IPv6 Neighbor Discovery

LLN Low-Power and Lossy Network

LR-WPAN Low-Rate Wireless Personal Area Network

MAC Medium Access Control

MIC Message Integrity Code

FFD Full Function Device

RFD Reduced Function Device

NA Neighbor Advertisement

NCE Neighbor Cache Entry

ND Neighbor Discovery

NDP Neighbor Discovery Protocol

NS Neighbor Solicitation

ARP Address Resolution Protocol

PAN Personal Area Network

QoS Quality of Service

RA Router Advertisement

ROHC Robust Header Compression

RPL Routing Protocol for Low-Power and Lossy Networks

RS Router Solicitation

SLAAC Stateless Address Autoconfiguration

SLLAO Source Link-Layer Address Option

TLLAO Target Link-layer Address Option

PIO Prefix Information Option

RH Redirected Header

MTU Maximum Transmission Unit

ARO Address Registration Option

EARO Extended Address Registration Option

6CO 6LoWPAN Context Option

ABRO Authoritative Border Router Option

6CIO 6LoWPAN Capability Indication Option

6LoWPAN-HC1 6LoWPAN_Header Compression1

LoWPAN-NHC LoWPAN Next Header Compression

6LoWPAN-IPHC 6LoWPAN Internet Protocol Header Compression

6LoWPAN-GHC 6LoWPAN Generic Header Compression

WPAN Wireless Personal Area Network

WSN Wireless Sensor Network

6LBR 6LoWPAN Border Router

6LR 6LoWPAN Router

6LN 6LoWPAN Node

6BBR 6LoWPAN Backbone Router

ROVR Registration Ownership Verifier

ROLL Routing Over Low power and Lossy networks

NUD Neighbor Unreachability Detection

WG Working Group

IPSec Internet Protocol Security

AES Advanced Encryption Standard

AH Authentication Header

ESP Encapsulating Security Payload

HPKE Hybrid Public Key Encryption

KDF Key Derivation Function

KEM Key Encapsulation Mechanism

AEAD Authenticated Encryption with Additional Data

PDMv1 Performance and Diagnostic Metrics Version1

IDS Intrusion Detection Systems

IPS Intrusion Prevention System

IANA Internet Assigned Numbers Authority

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the IoT systems in general, their applications in our daily life, current security threats to IoTs, the key requirements to build IoT networks. Moreover, it explains that how standardized organizations are designing IoT architectures. The problem formulation of this thesis is also explained in this chapter, which deals with the resilience of IoT networks services and data.

1.1 Foundation

The Internet of Things (IoT) comprises of network "smart" devices (things or objects). These smart devices are able to self configure, connect, exchange, and elaborate data to each other. It works like an ecosystem. The IoT Systems are assumed to be developed in a way where all devices can directly send (immediately or periodically) their information to the Internet. A designated server (or servers) on the Internet takes action accordingly after receive and process the data.

Whereas, the WSN is a network that is based on multiple dedicated sensors to monitor, record, and measure the physical conditions of the environment. Later, these sensors cooperatively move the given data to the main location or central point, called the *Gateway* (GW), which is installed at the edge of the WSN. Only the GW is responsible to forward the received data to the Internet for the required processing. There is no direct connection to the Internet for each sensor node installed inside WSNs.

The WSN is the subset of the IoT Systems. They are based on very small, low power, low data rate, low complexity, short-range Radio Frequency (RF), sleep mode, undefined location, topologies, and inexpensive devices, usually only equipped with just a transceiver and a micro-controller. The devices are often battery-operated, and they make intensive use of duty-cycled operations. Usually, they are based on Institute of Electrical and Electronic Engineers (IEEE) 802.15.4 standard [67], which imposes some more restrictions, e.g., the payload size. Due to this kind of peculiarities, the Internet Protocol version 6 (IPv6) can not be used as it is and must be adapted because of its several beneficial features [42]. The IETF has developed several standards to attempt to adapt IPv6 to the LLNs, where a LLN is a generalization of a WSN.

There are some reference models presented by the different standard organisations, for example, in [136], International Telecommunication Union (ITU) introduced new dimensions on IoTs in terms of any TIME, PLACE, and THING connectivity (see Sections 1.3). This demands high *availability* and *scalability* of the IoT devices. The IEEE 802.15.4 standard defines compatible the *Physical* and *Data-link* layers for constrained devices those form the Wireless Personal Area Network (WPAN) [67].

It must be noted that WPANs, and more in general LLNs can differ a lot from 'traditional' networks. As an example, in IEEE 802.15.4 it is possible to use short and extended Medium Access Control (MAC) addresses (16 & 64 bits respectively), the bandwidth is very low (250 kbps, 40 kbps, and 20 kbps for each physical layers 2.4 GHz, 915 MHz, 868 MHz, respectively), datagrams are short (less than 128 bytes), devices have typically low power and low cost (constrained devices), the links have a large loss probability, etc.

One of the first IETF standards for WPANs was released in Aug. 2007 [84]. It is commonly referred as *6LoWPAN*, even though the correct title is *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals.* Here, one can find summarized all the points that make WPANs (and LLNs) peculiar with respect to traditional wireless networks.

The transmission of IPv6 packets over IEEE 802.15.4 networks is described in [105]. It provides some solutions such as the mechanism to accommodate the large Maximum Transmission Unit (MTU) requirement of IPv6. To solve the said issues, a shim adaptation layer was developed, which is also known as IPv6 over Low power Wireless Personal Area Network (6LoWPAN)

Layer. It works between the Network and MAC layers and it handles the IPv6 header compression and size adaptation (trough fragmentation and reassemply). In 2011 [64] improved the compression mechanism for multicast addresses, IPv6 Next Header and User Datagram Protocol (UDP) header, furtherly improved in [27]

Unfortunately, efficient compression and transmission is only one part of the problem. Other open issues are represented by routing, neighbor discovering, and IPv6 address management.

In 2012, IETF developed Routing Protocol for Low-Power and Lossy Networks (RPL) (or *Ripple*), to fulfill the 6LoWPANs routing requirements. This Directed Acyclic Graph (DAG) based protocol supports point-to-point (P2P), and (optionally) point-to-multipoint (P2MP), and multipoint-to-point (MP2P) traffic flows [158]. In the same year, IETF proposed a solution for the problem of the neighbor discovery in 6LoWPANs [130]. Later on [143] refined the node registration mechanism, and mobility detection for different network topologies.

For what concerns network robustness and security, the 6LoWPANs, in the present release, is not capable to fulfill all the functional and security requirements. For example, all the above-cited proposals rely on the Link level security which is an open threat to Layer-3 protocols (RPL can have its own security mechanisms). In addition to this, more than one standard is based on *tree* topology such as [67], [158], and [143], where a single head node, also known as *root* node, may become a Single Point of Failure (SPoF). So while designing the IoT networks the availability of the root node is very critical and important.

The work that I present in this thesis is to make secure and resilient IoT networks. My research investigation put light on how the standard protocols make IoT networks *robust* and *agile* such as 6LoWPAN-ND and IPv6-ND, and where they are not *reliable* or fit in terms *availability*, which is the core function of any kind of network data and security services. I also explore the application of federated learning to virtual functions demand prediction in IoE (i.e., the extended concept of IoT) based edge-cloud computing systems, to preserve the data security and maximize service provider revenue. Moreover, to make secure IoT networks, it is important to get knowledge about the data and services a network is utilizing. If data is communicating in a clear-text then it will become an open door for the malicious actors to get information for subsequent attacks. In this aspect, I present the application of the HPKE framework to secure the Performance and Diagnostic Metrics (PDM) Destination Option header. This framework includes three important security operations known as Key Encapsulation Mechanism (KEM), Key Derivation Function (KDF), and Authenticated Encryption with Additional Data (AEAD). Moreover, the security solution with a lightweight handshake (registration procedure) and encryption to secure data will be explained.

1.2 IoT Background: History and Applications

The term IoT was introduced by Kevin Ashton in 1999. At that time, the Internet was the hottest trend, and technology was getting a new foundation. It is a system of interrelated computing devices, mechanical and digital machines, animals or people, all everyday objects directly or indirectly connected to human's daily lives. They provide unique identification and the ability to transfer data over the Internet without requiring human-to-human or human-to-computer interaction. Abstractly, I can say the Internet of Things – the "things" that are connected, and the "Internet" that interconnects them [136].

Currently, we are living in an era where we are surrounded by billion IoT devices. They have grown excessively in the last 5 years. IoT devices globally are predicted to almost triple from 9.12 billion in the year 2021 to more than 25.4 billion IoT devices in 2030. IoT applications and benefits in our daily life are innumerable. Figure 1.1 depicts the IoT application in multiple domains. For example, providing the intelligent networking of machines and processes in the industry with the aid of information and communication technology. The basic concept is that factories in which machines are augmented with wireless connectivity and sensors, connected to a system that can visualize the entire production process, control, and make decisions on its own (without human interaction). Currently, a term used *Fourth Industrial Revolution* shortly known as Industry–4.0. This magical shift increases productivity, which is directly connected to the economy of the state [21] [60] [34] [125].

Their use-case in the health care domain is also very important. Since December 2019 and still, the whole world is suffering from the COVID-19, and people are advised by the *World Health Organization*(WHO) to keep

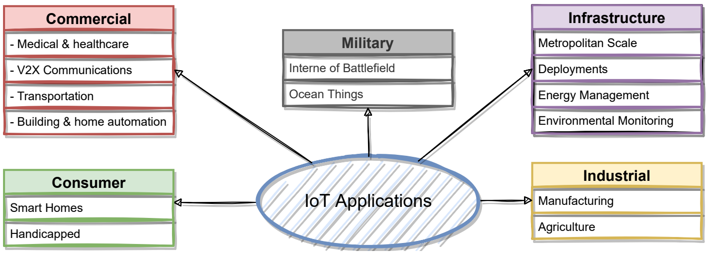


Figure 1.1: IoT Applications.

social distancing (1–meter) to each other and others to reduce the risk of infection. In this scary, insecure, and pandemic situation, where humans can't even shake-hand or hug each other, then how a medical doctor can rescue a patient without touching. So there is a great need above technologies than before, and IoT devices are playing a very important role. IoT devices have enhanced the fight against COVID-19, including the newly commercialized 5G technology [122]. In addition, 5G networks enable functional integration of computing and communication capabilities, enabling smart IoT applications that can profoundly change various aspects of our lives. Despite the potential benefits of 5G, several challenges still need to be overcome before the IoT paradigm actually becomes a widespread reality [57].

1.3 IoT Architecture

Connectivity of heterogeneous IoT devices with Internet yields many problems like *scalability*, *manageability*, *controllability*, and *security*. A new dimension Y.2060 for IoT network's introduces by ITU depicted in the Figure 1.2, is any TIME connection, any PLACE connection, and any THING connection. It is considered to be the next big opportunity, and challenge, for the Internet engineering community, users of technology, companies, and society. Because IoT is primarily driven by deeply embedded devices. These devices are low-bandwidth, low-repetition data capture, and low-bandwidth data-usage appliances that communicate with each other and provide data via user interfaces. Embedded appliances, such as high-resolution video security cameras, video over IP (VoIP) phones, and a handful of others, require high bandwidth streaming capabilities. Yet countless products simply require packets of data to be intermittently delivered. With these dimensions,

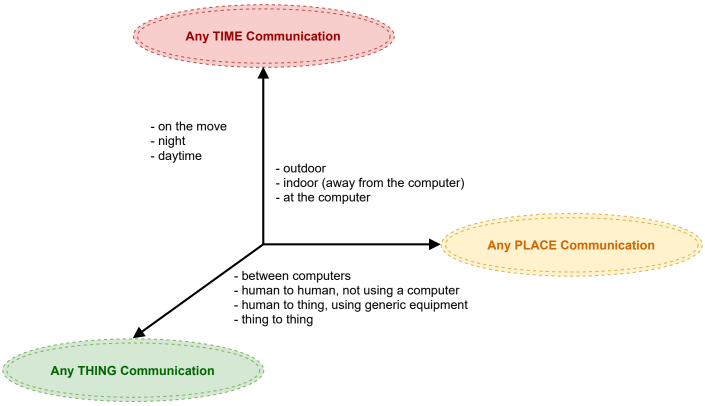


Figure 1.2: ITU-T Y.2060 new dimension introduced in the Internet of Things

security considerations became the focus of consumers.

A standard architecture design for IoT is still an open issue. But many international organizations such as ITU, IETF, IEEE etc., are actively engaged in the development and standardization of IoT architecture and protocols. In a general perspective and most common layered architecture is compromised on four layers as shown in the Figure 1.3.

If I compare this layered architecture with the TCP/IP stack, I can analyze the similarity among them. Yes, indeed it's highly fortunate that the same traditional security threats are also possible to IoT devices. Conventional threats [1], especially to the perception layer (Physical layer of TCP/IP stack), can exploit the whole IoT network due to the insecure installation of sensors and actuators, moreover the lack of security provided to such devices by the manufacturers [136].

1.4 IoT Security

Securing heterogeneous IoTs and transmitting data on a large scale and in a distributive way is very challenging. In fact, sharing data contains a large amount of private information, preserving information security on the shared data is an important issue that cannot be neglected. The Figure 1.4 shows

[1] Threats that already exist to unconstrained networks and devices

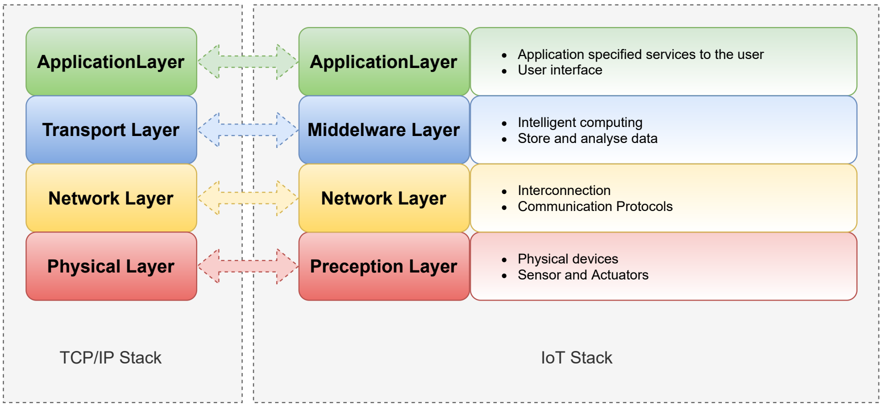


Figure 1.3: General IoT Architecture.

the main elements of interest for IoT security and several typical scenarios for interconnection and the inclusion of security features.

Application platforms or data storage servers, and network and security management systems are shown at the center of the network. These central systems gather data from sensors, send control signals to actuators, and are responsible for managing the IoT devices and their communication networks. At the edge of the network are IoT-enabled devices, some of which are quite simple constrained devices and some of which are more intelligent unconstrained devices. Protocol conversion and other networking services on behalf of IoT devices are the conventional jobs of the *GW*. Shading devices indicate the system is secure, Typically, GWs implement secure functions, such as Transport Layer Security (TLS) and Internet Protocol Security (IPsec). Unconstrained devices may or may not implement some security capability. Constrained devices generally have limited or no security features. However, any constrained or unconstrained devices attached to the GW are outside the zone of security established between the GW and the central systems. Unconstrained devices can communicate directly with the center and support security functions. However, constrained devices that are not connected to GWs have no secure communications with central devices.

The IoT is perhaps the most complex and least developed area of network security. The reason for this is that chip manufacturers have strong incentives to produce their products as quickly and cheaply as possible with their firmware and software. Their focus is on the functionality of the device

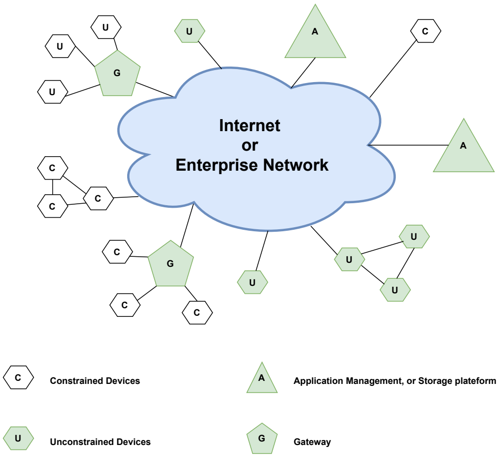


Figure 1.4: Elements of Interest in The Contexts of IoT Security.

itself rather than the security features as this increases the cost of the device. The end-user may have no way to patch the system, and if they do, they know little about when and how to patch. The result is that the hundreds of millions of Internet-connected devices in the IoT environment are vulnerable to attack.

1.4.1 IoT Security Requirements

Like conventional networks [2], the essential security requirements such as *confidentiality, authenticity, integrity, accountability*, and *availability* are also mandatory to protect IoT network data and services. These five basic security functions must consider while designing IoT networks because these functions play a main role to mitigate security attacks. Figure 1.5 depicts these important functions. I explain briefly each function.

1. **Confidentiality**: This function frames two related concepts:

   - *Data Confidentiality:* It ensures that private or confidential information is not made available or disclosed to unauthorized individuals.

[2] Networks that are not formed with constrained nodes

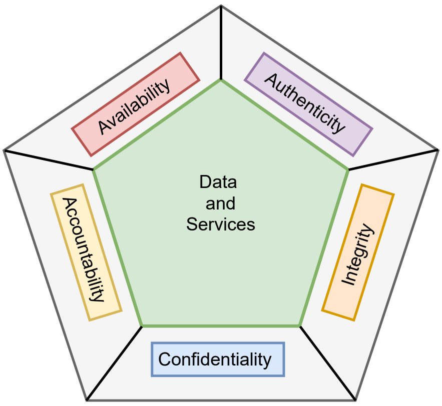


Figure 1.5: Essential IoT Network Requirements

- *Privacy:* It ensures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

2. **Integrity**: This function also covers two related concepts:
    - *Data Integrity:* It ensures that information (both stored and on-air) and programs are changed only in a specified and authorized manner.
    - *System Integrity:* It ensures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

3. **Accountability:** Its objective is to generate the requirement for actions of an entity to be traced uniquely to that entity. It supports the assurance that someone cannot deny something, also called non-repudiation. Moreover, it also supports deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.

4. **Authenticity:** It ensures that the communication from one IoT object to another is genuine, that is, a malicious object cannot masquerade as a trusted network object. It is a property of being genuine and being able to be verified and trusted, confidence in the validity of the transmission, a message, or the message originator.

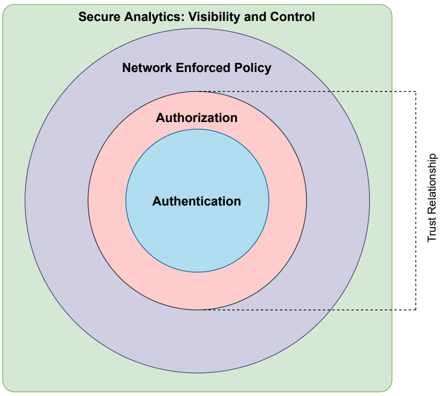


Figure 1.6: Secure IoT Framework.

5. **Availability:** It ensures that the desired network services are available even in the presence of denial-of-service (DoS) attacks. To attain this requirement some cloning functional devices are used in the IoT network. Cloned devices satisfy and provide the running services in case of any failure happen with any device. To ensure this requirement, I can make IoT network more robust and resilient. I will discuss resilience and robustness in the Chapter 4 in more detail.

1.4.2 Secure IoT Framework

Under core security requirements umbrella mentioned in the Section 1.4.1, there are other *IoT Security and Privacy Requirements* defined by International Telecommunication Union (ITU). Moreover, they also published *Y.2067, Common Requirements and Capabilities of a Gateway for the Internet of Things Applications, in June 2014*, specific security functions that the GW should implement. Cisco also playing a key role in the development of IoT World Forum Reference Model, and has developed a framework for IoT security, the detailed discussions are mentioned in [136]. In their White Paper on IoT security, Cisco proposes a secure IoT framework that defines the components of a secure facility for an IoT that includes all the levels, as shown in Figure 1.6.

Authentication: It includes the elements that initiate the determination of access by first identifying the IoT devices. Unlike typical devices on enterprise networks that can be identified by a human credential (e.g., username and password or token), the IoT endpoints are fingerprinted with identifiers that do not require human interaction. Such identifiers include RFID, x.509 certificates, or the MAC address of the endpoint.

Authorization: It controls a device's access to the entire network fabric. This element includes access control. Together with the authentication layer, it defines the necessary parameters for the exchange of information between devices and between devices and application platforms and enables the execution of IoT-related services.

Network Enforced Policy: Includes all elements that securely route and transport endpoint traffic across the infrastructure, whether it is control, the management, or actual traffic.

Secure analytics, including visibility and control: This component includes all the features needed to centrally manage IoT devices. First, this includes IoT device visibility, which simply means that centralized management services are securely aware of the distributed IoT device collection, including the identity and attributes of each device. Building on this visibility, there is the ability to exercise control, including configuration, patch updates, and threat mitigation.

1.5 Security Related Issues and Challenges in IoT

IoT devices have many security issues and challenges related to the protocol used at each layer, and not only that, but they also have management and control issues and challenges. I summarize some of them and present in the Table 1.1, but the recent detailed work is discussed in [16], [73], [132] and [137]. If I say what are general challenges in IoT then I would enlist them as

1. *QoS:* IoT networks must provide satisfactory services (e.g., service time, service availability, service delay, service load, service priority, etc.).

2. *Availability:* IoT networks must be robust to face any kind of malicious activity or technical fault.

Table 1.1: Security issues or attacks with respect to IoT layers

IoT Layers	**Security Issues/Attacks**
Application Layer	Data access and security authentication issues, data protection and recovery problems, spear-phishing attack, software vulnerabilities, attacks on reliability, and clone attack.
Middleware Layer	Making intelligent decision processing huge data, malicious-code attacks, multi-party authentication, handling suspicious information.
Network Layer	Cluster security problems, DoS attacks, spoofed, altered or replayed routing information.
Perception Layer	Node capture, fake node, mass node authentication, cryptographic algorithm, and key management mechanism.

3. *Reliability:* How long the object is usable and application robustness in the face of uncertain information.

4. *Scalability:* IoT systems must have the ability to support an increasing number of connected devices, users, and analytical capabilities without any degradation in QoS.

5. *Security:* This challenge needs to effectively deal with authentication, authorization, access control, trust, and privacy requirements without negatively impacting usability.

The object of this thesis is to make resilient IoT systems by analyzing various protocol behavior and machine learning approaches and providing solutions. I present and discuss current standard problems in chapters 3, 4, and 6 and proposed solutions to make resilient IoT systems. In Chapter 5, I present an approach of Federated Learning (FL) which is the technique of Machine Learning (ML) to increase the availability of the virtual function for the IoT devices.

1.6 Thesis Contributions

The main contributions of the thesis are listed below;

- I present a novel design for WSNs, to address availability and other challenges by shifting mandatory functionalities of LLNs from unreliable to reliable and stable domains.

- The key features of protocols, such as IEEE 802.15.4, 6LBR, and Virtual DoDAG root, are moved from an uncertain network to a more stable and controlled network.

- Synchronization of (EPs) (i.e., Gateways Routers) that provide IPv6 Backbone Router (6BBR) and DoDAG Root functions.

- The use of Virtual DoDAG at the Fog Layer, which aids in the synchronisation of many DoDAG roots.

- The use of a fog layer promotes adaptability and helps restricted devices for time-critical tasks.

- Assuring network functions and stating that my design is more robust since I have synchronised numerous EPs.

- Due to LLN peculiarities 6LoWPAN-ND protocol is not practical in mesh-under, I analyze the 6LoWPAN-ND protocol in multiple scenarios, where it reduces the multicast and increases the unicast transmission in a drastic way.

- In terms of reliability and robustness the IPv6-ND protocol lacks the network resilience because it works in a reactive way. I addressed the below mentioned questions with obtained results in terms of IoT resiliency. e.g.,

  1. Is the 6LoWPAN-ND protocol necessary for a particular LLN?
  2. What are the benefits and drawbacks of adopting 6LoWPAN-ND protocol?
  3. What would happen if we do not employ 6LoWPAN-ND protocol for LLNs and continue to use the IPv6-ND protocol?

- The Implementation of the FL method to anticipate network VFs usage in order to protect individual privacy;

- Formulation of the SP increasing revenue issue by taking into account Service Requests (SRs) having varying priorities and, as a result, varying price and value. In addition, if all of the high priority lows have been fulfilled, the SP can take data SRs with lesser importance.

- Based on the studied FL and the previously supplied VFs forecasting system, provide a VFs placement strategy and an appropriate matching-based SRs allocation procedure.

- Extensive computer simulation trials were used to evaluate the suggested approach's performance and compare it to a centralised Chaos Theory (CT)based prediction strategy.

- Present a lightweight handshake (registration procedure) and encryption to safeguard PDM data.

- Develop a list of additional performance measures that might be useful for performance evaluation of IoT systems.

- Application of IRTF HPKE framework to provide confidentiality and integrity to PDM data.

1.7 Thesis Organization

The organization of this thesis is as follows: In the Chapter 1, I state the problems that I work on during my Ph.d. research. Moreover, an overview of IoT applications, standard architectures, and requirements at each layer of the IoT stack also discussed. In the Chapter 2, I present an existing work from the relevant standard organization and their ongoing work. 3, describes the resilience in general and for IoT environment. Moreover, my proposed architecture addresses the availability of the gateway node in the context of core functionalities running on it. In the Chapter 4, I discuss the importance of the 6LoWPAN-ND protocol, why it is necessary for LLNs, and how it affects the robustness and reliability of the network. Moreover, I also compare it with traditional classical IPv6-ND.

In the Chapter 5, Federated Learning approach is applied to analyze the revenue of network service providers. To achieve the availability of virtual function, cross-layer frameworks consisting of virtual network feature placement, user demand prediction through the federated learning paradigm, and user allocation are realized to maximize the service provider's profit.

The HPKE security operations (KEM, KDF, and AEAD) application over PDM Destination Option, known as PDMv2 discussed in the Chapter 6.

The Chapter 7 encompasses the two research projects where I worked during my Ph.d. Finally, the Conclusion is drawn in Chapter 8, which summarizes the overall work presented in the previous chapters. It concludes that how security in terms of confidentiality, integrity, robustness, reliability, and availability is important to make IoT systems resilient.

Chapter 2

Relevant Standards

This chapter provides an overview of IoT existing standardized protocols that are used in IoT devices. Detailed information on specific layers or industry-specific protocols, plus a comparison of popular protocols, in the context of IoT resiliency. Moreover, the discussion also includes that in which domain different working groups are doing research and how their protocols fit in the TCP/IP stack and beneficial for the IoT systems. What kind of security services that existing standards and ongoing research work making the IoT networks resilient.

2.1 Introduction

IoTs covers a wide range of industries and use cases, ranging from a single, limited device to massive, cross-platform deployments of embedded technologies and cloud systems connected in real-time. It's all tied together by numerous existing and new communication standard protocols that enable devices and servers to communicate with each other in new, more connected ways. At the same time, dozens of alliances and coalitions are forming in hopes of unifying the fragmented and organic IoT landscape.

Robustness is the property of being strong and healthy in constitution of any kind of communication network. IEEE 802.15.4 standard for Low-Rate Wireless Personal Area Network (LR-WPAN) has a strong mechanism which addresses the robustness. Moreover, RPL also provide the robustness in WPAN at the Layer three. Considering the Layer four, off course Transmission Control Protocol (TCP) is more robust than UDP and reliable. But it depends on the application which transport protocol is required for the purpose of communication. A robust network can be achieved, if low level protocols provide strong loop free routing, error correction and acknowledgement mechanisms. I have reviewed this requirement in literature where standards provide some mechanisms to achieve the robustness dynamically.

This chapter provides an overview of IoT existing standardized protocols that are used in IoT devices. Detailed information on specific layers or industry-specific protocols, plus a comparison of popular protocols, in the context of IoT resiliency.

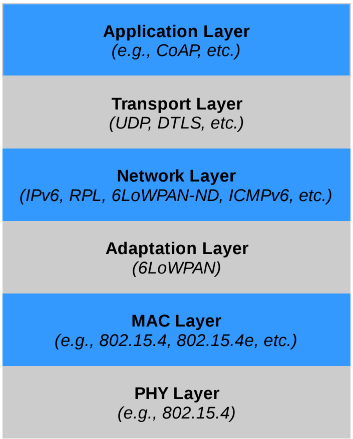


Figure 2.1: TCP/IP Stack with Existing Standard Protocols For IoT

2.2 Existing Standard

At the development of protocol for IoT networks, there are many working groups in various authoritative organizations. IEEE defines PHY and MAC layer protocols, and the standard known as IEEE 802.15.4[1]. IETF initial working group called 6LoWPAN[2] gave the foundation to utilize IPv6 over IEEE 802.15.4 standard then they closed that group. Successor of the 6LoWPAN working group defines specifications for running IPv6 over multiple constrained L2 (Layer Two) technologies that use a base 6LoWPAN

[1] https://www.ieee802.org/15/about.html

[2] https://datatracker.ietf.org/wg/6lowpan/email/

stack. This group known as IPv6 over Networks of Resource-constrained Nodes (6lo)[3]. Another IETF group focused on routing solutions, called Routing Over Low power and Lossy networks (ROLL) [4].

2.2.1 IEEE 802.15.4

It is a technological standard that specifies how low-rate WPANs should be operated. It is maintained by the IEEE 802.15 working group, which developed the first standard in 2003 and specifies the physical layer and media access control for LR-WPANs [65]. The most available standards that are studied, implemented in IoT devices, and simulators are [66] and [67]. The most recent standard is published in 2020 [68] by IEEE. It is the fourth revision of IEEE Std 802.15.4 from the beginning. It adds two more PHY amendments and one MAC amendment. Like old version, this standard keep four basic frames (Beacon, Data, Acknowledgement and MAC command frame) and supporting *star* and *peer-to-peer* topology. Moreover, it can also form a cluster tree where a single node works as super Personal Area Network (PAN) Coordinator. It is based on two types of devices called Full Function Device (FFD) and Reduced Function Device (RFD). FFD is capable of serving as a PAN coordinator or a coordinator, whereas, RFD is not capable of serving as either a PAN coordinator or a coordinator. PAN coordinator is a SPoF entity in the standard for both star or peer-to-peer topologies, so the resilience is of the protocol is challenging. Chapter 4 will throw more light on it.

Robustness in IEEE 802.15.4 Standard

The LR-WPAN by adopting IEEE 802.15.4 standard employs various mechanisms to ensure robustness in the data transmission at MAC level. These mechanisms are

1. **CSMA-CA Mechanism:** Standard uses two types of channel access mechanism depending of the network configuration.

    (a) *Slotted Channel Access:* This method is used in beacon-enabled networks. Each time a device wishes to transmit data frames or MAC commands, it shall wait for a random period. If the channel

[3] https://datatracker.ietf.org/wg/6lo/about/

[4] https://datatracker.ietf.org/wg/roll/about

is found to be idle, following the random back-off, the device shall transmit its data. If the channel is found to be busy, following the random back-off, the device shall wait for another random period before trying to access the channel again, but Acknowledgment frames shall be sent without using a Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) mechanism [65].

(b) *Unslotted Channel Access:*

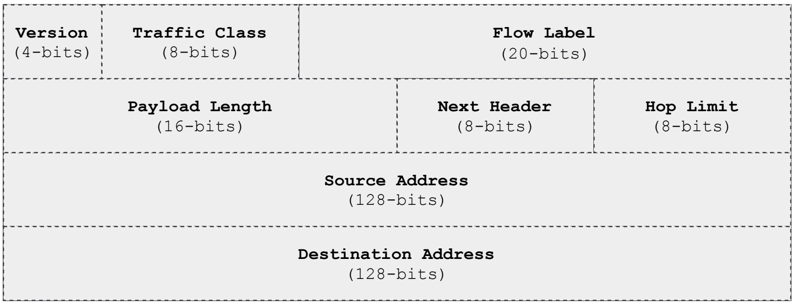
This method is used in Non-beacon enabled networks, where back-off slots are aligned with the start of the beacon transmission. Each time a device wishes to transmit data frames during the Contention Access Period (CAP), it shall locate the boundary of the next back-off slot and then wait for a random number of back-off slots. Following this random back-off, if the channel is busy, the device will wait for another random number of back-off slots before attempting to access the channel again. The device can start broadcasting on the next available back-off slot boundary if the channel is idle. Without the use of a CSMA-CA protocol, acknowledgement and beacon frames must be delivered [65].

2. **Frame Acknowledgment:** Successful reception and validation of a data or MAC command frame may optionally be acknowledged with an acknowledgement. If the receiving device is unable to process the received data frame for any reason, the message is not acknowledged. If the sender does not receive an acknowledgement after a certain time, it assumes that the transmission was unsuccessful and retries transmitting the frame. If no acknowledgement is received even after several attempts, the sender can either terminate the transaction or try again. If confirmation is not required, the sender assumes that the transmission was successful [65].

3. **Data Verification:** In order to detect bit errors, an Frame Check Sequence (FCS) mechanism, employing a 16-bit ITU-T standardized Cyclic Redundancy Check (CRC), is used to protect every frame [65].

2.2.2 IPv6

Like Internet Protocol version 4 (IPv4), IPv6 also works as a best effort protocol, but it provides way more *flexibility* and *scalability* to IoTs [61] [42]. It provides various types of addresses as shown in the Figure 2.3. There were four Internet Protocol (IP) connectivity related problems mentioned in [84] such as devices in a Low power Wireless Personal Area Network (LoWPAN) make network auto configuration and statelessness highly desirable. IPv6 has a solutions of Stateless Address Autoconfiguration (SLAAC) [140]. Second, large number of devices poses the need for a large address space. IPv6 has a solution (2^{128} - bits). Third, given the limited packet size of LoWPANs, yes, IPv6 address format allows subsuming of IEEE 802.15.4 addresses if so desired. fourth, interconnectivity to other IP networks including the Internet, of course IPv6 has this ability. The IPv6 header format is depicted in Figure 2.2.



Version (4-bits) | **Traffic Class** (8-bits) | **Flow Label** (20-bits)

Payload Length (16-bits) | **Next Header** (8-bits) | **Hop Limit** (8-bits)

Source Address (128-bits)

Destination Address (128-bits)

Figure 2.2: IPv6 Header

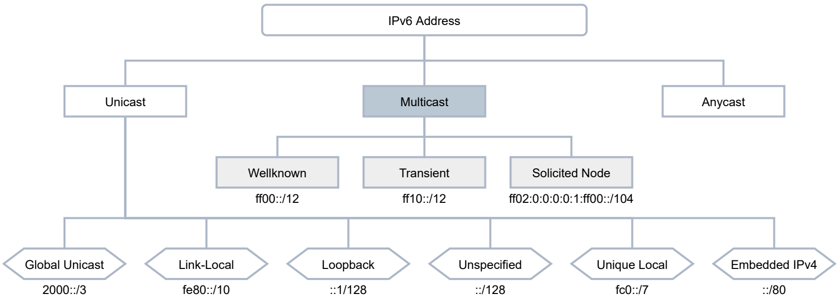


Figure 2.3: IPv6 addresses types

2.2.3 ICMPv6

IPv6 uses the Internet Control Message Protocol (ICMP) as defined for IPv4 in [113], with a number of changes. The resulting protocol is called Internet Control Message Protocol version 6 (ICMPv6) and has an IPv6 Next Header value of 58 [38]. This standard provides set of control messages used in ICMPv6, such as error and informational messages. This protocol is significantly utilized by neighbor discovery and routing protocols.

2.2.4 IPv6-ND

Neighbor Discovery Protocols (NDPs) are used to resolve virtual address/es (i.e., IP address/es) into physical address (i.e., MAC address) of the device (i.e., communication interface) or vice versa. In IPv4 uses Address Resolution Protocol (ARP) however IPv6 use IPv6-ND. In IPv6 all determined addresses are stored as information in neighbor cache. IPv6 hosts can automatically locate default routers on link using two ICMPv6 messages such as Router Solicitation (RS) and Router Advertisement (RA). For the address resolution, Duplicate Address Detection (DAD), and Neighbor Unreachability Detection (NUD) Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages are used. The Redirect message is used by the routers to inform a host of a better first-hop node on the path to a destination. These five messages can carry five option header for various purposes explained in the base standard [109]. These option headers are Source Link-Layer Address Option (SLLAO), Target Link-layer Address Option (TLLAO), Prefix Information Option (PIO), Redirected Header (RH), and MTU.

IPv6-ND core functions are enlisted below

1. **Router Discovery:** Enables hosts to find on-link routers.

2. **Prefix Discovery:** Enables host to determine what destinations are available on-link and which ones are available through a router.

3. **Parameter Discovery:** Enables host to learn network parameters such as link MTU or Hop limit.

4. **Address Auto-configuration:** Provide a way to hosts to auto-configure their interface addresses.

5. **Address Resolution:** Resolve IP layer address to Link Layer address (like ARP).

6. **Next-hop determination:** Algorithm to determine whether destination is on-link or off-link and subsequently how to reach the destination.

7. **Neighbor Unreachability Detection (NUD):** How nodes determine that a neighbor is no longer reachable.

8. **Duplicate Address Detection (DAD):** How a node determines whether or not an address it wishes to use is already in use by another node.

9. **Redirect:** How a router informs a host of a better first-hop node to reach a particular destination.

These fundamental operations are important in IPv6 networks, but they aren't designed for non-transitive wireless connections since their reliance on the standard IPv6 link model and heavy usage of multicast render them inefficient and often impossible in a low-power, lossy network. Furthermore, the IETF's IPv6 over 6LoWPAN study defines 6LoWPANs like IEEE 802.15.4. Due to energy saving, this and other comparable connection technologies use minimal or no multicast signaling. Furthermore, the wireless network may not fully adhere to the standard IP subnets and IP connections concepts. However, in the case of mesh-under topologies, this is not the case, as I discovered throughout my study. In the Chapter 5, I will go through everything in depth.

2.2.5 6LoWPAN

It is a shim layer between MAC and IP layers. It is also known as 6LoWPAN Adaptation Layer, as depicted in the Figure 2.1. The main concept of this shim layer is the use of stateless or shared context compression of network (IPv6) and transport (UDP) layer header fields. Due to this layer we can increase payload size at transport, network and even at MAC layer.

IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals-RFC4919

Why this layer is required? well as discussed in the Section 2.2.2 the inherent attributes of IPv6 to increase the scalability and flexibility, IETF needs to cope with the IPv6 minimum MTU requirements, which is 1280 octets but the underlaying layer such as IEEE 802.15.4 supports maximum PHY Service Data Unit (PSDU) of 127 bytes. So maximum frame size at the MAC layer is 102 Bytes (MAC Security+Payload). If MAC layer security imposes foster overhead as defined in [68]. Hence, there is essential need for the 6LoWPANs adaptation layer to provide fragmentation and reassembly to support minimum MTU size for IPv6. Figure 2.4 explains the headers and payloads variations. So while designing this layer, three primary elements considered, as explained below;

Security	MAC header	Payload	PSDU
Bytes			
21-AES-CCM-128	25	81	127
13-AES-CCM-64		89	
9-AES-CCM-32		93	

Table 2.1: Payload Variation According to MAC Layer Security

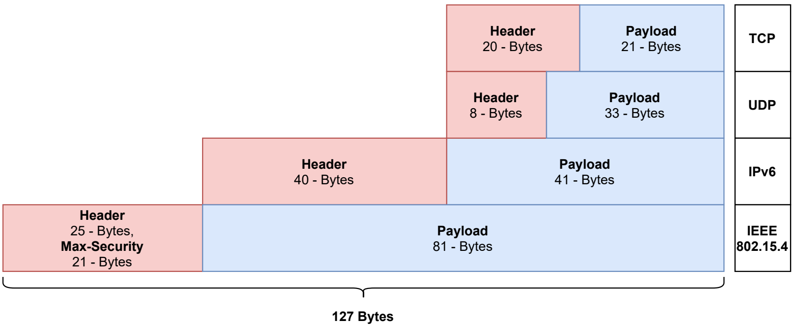


Figure 2.4: Headers and Payloads Length Calculation

6LoWPAN has three primary elements:

1. **Header compression:** IPv6 header fields are compressed by assum-

ing usage of common values. Header fields are elided from a packet when the adaptation layer can derive them from link level information carried in the 802.15.4 frame or based on simple assumptions of shared context.

2. **Fragmentation:** To meet the IPv6 minimum MTU requirement, IPv6 packets are divided into numerous link level frames.

3. **Layer 2 forwarding:** The adaption layer can carry link level addresses for the ends of an IP hop to support layer two forwarding of IPv6 datagrams. Alternatively, the IP stack might use Layer 3 forwarding to achieve intra-PAN routing, with each 802.15.4 radio hop acting as an IP hop.

There are various characteristics defined in initial standard [84], as shown in the Table 2.2.

Table 2.2: Payload Variation According to MAC Layer Security

Characteristics of LoWPANs	
Addressing	It supports both short (16-bit) or extended (64-bits) media access control addresses.
Low Bandwidth	With respect to the physical layer as defined in IEEE 802.15.4 standard data rates for 2.4 GHz (Global) is 250 kbps, 40 kbps, and 20 kbps are for, 915 MHz (US), and 868 MHz (EU), respectively.
Topology	There are two topologies, star, and mesh. Paths in the network may change due to the link quality, but the topology remains the same.
Low Power	Some or all devices are battery operated.
Low Cost	Devices used in such networks are typically linked with sensors, switches, etc. This drives some of the other characteristics such as low processing, low memory, etc.
Location	Typically, the location of the devices not predefined, because they tend to be deployed in an ad-hoc manner. Sometimes its hard access the location of these devices. Additionally, these devices may move to new locations [84].
Unreliability	Devices are very unreliable because of verity of reasons, e.g. battery drain, physical tampering, uncertain radio connectivity, etc.
Sleep Time	Due to the environment, some devices sleep periods increases and they are unable to communicate during their sleep periods. The reason is to save the battery time.

Transmission of IPv6 Packets over IEEE 802.15.4 Networks-RFC4944

On top of IEEE 802.15.4 networks, [105] provides the frame structure for transmission of IPv6 [41] packets, as well as the generation of IPv6 link-local addresses and statelessly auto-configured addresses. As discussed above, an adaption layer is created because IPv6 requires support for packet sizes significantly bigger than the maximal IEEE 802.15.4 frame size. It also specifies the header compression algorithms needed to make IPv6 function on IEEE 802.15.4 networks, as well as the requirements for packet delivery in IEEE 802.15.4 meshes. It opens a door to develop a routing protocol at layer-2 level by defining Mesh, Fragmentation, Broadcast headers. However, a complete mesh routing definition remains an open issue.

Compression Format for IPv6 Data-grams over IEEE802.15.4 Based Networks-RFC6282

[64] updates [105] by defining 6LoWPAN Internet Protocol Header Compression (6LoWPAN-IPHC) compression for IPv6 header, addresses (multicast and unicast), Next Headers (Hop-by-Hop, Routing, Fragment, Destination Options and Mobility headers) called LoWPAN Next Header Compression (LoWPAN-NHC) and UDP header compression framework, known as UDP LoWPAN-NHC.

6LoWPAN_Header Compression1 (6LoWPAN-HC1) in [105] is most effective for link-local unicast addresses, so well known link-local prefixes and Interface Identifier (IID) derived from IEEE 802.15.4 MAC. So in this case, both addresses may be completely elided.

Second, link-local are not used for application-layer data traffic, so the actual value presented in [105] compression mechanism is limited.

Third, routable addresses used when for off-link (route-over) communication required within 6LoWPAN. To carry prefix in-line for routable addresses, 6LoWPAN-HC1 requires both IPv6 Source Address and IPv6 Destination Address. IID of routable address must be carried in-line. Where Mesh Addressing header is not used. 6LoWPAN-HC1 requires 64 bits for the IID when carried in-line and cannot be shortened even when it is derived from the IEEE 802.15.4 16-bit short address. 6LoWPAN-HC1 requires full 128-bit address to be carried in-line, when destination is IPv6 multicast address.

As a result, [64] defines an encoding format 6LoWPAN-IPHC for effective compression of:

1. Unique Local IPv6 Addresses
2. Global/Unicast IPv6 Addresses,
3. Multicast IPv6 Addresses

Compression is based on shared state within contexts. Also introduces a number of additional improvements over the header compression format defined in [105]. Compression formation of IPv6 header illustrated in the Figure 2.5.

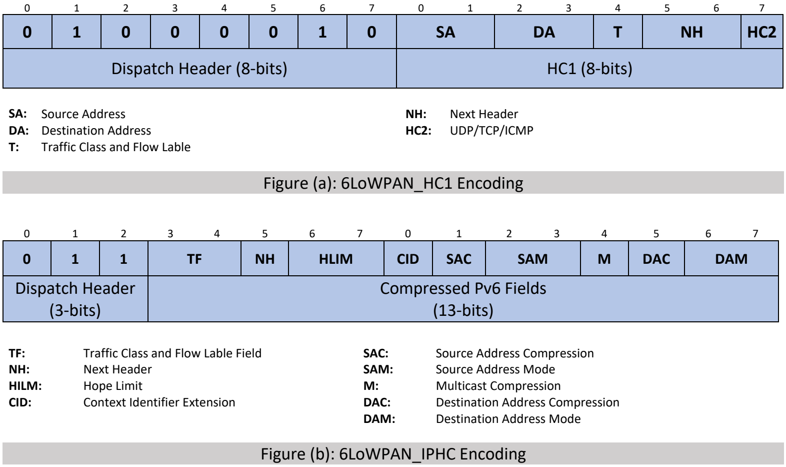


Figure (a): 6LoWPAN_HC1 Encoding

Figure (b): 6LoWPAN_IPHC Encoding

Figure 2.5: Compression variation from RFC4944 to RFC6282

6LoWPAN-GHC: Generic Header Compression for 6LoWPANs-RFC7400

In [64], a method defined for header compression in 6LoWPAN packets. The [27] presents a new specification for every new kind of header that needs to be compressed. Furthermore, [64] does not define an extension scheme like the Robust Header Compression (ROHC). This leads to the difficult situation that 6LoWPAN header compression must be reopened and rechecked every time a new header is considered in the 6LoWPAN/roll/CoRE group of IETF working groups. Although [64] was successfully published, but the underlying problem remains unresolved. This standard keep follow up on [64] and Next Header Compression (NHC) concept and adds slightly less efficient but far more general form of compression for headers of any kind, and even for header-like payloads as exhibited by routing protocols, Dynamic Host Configuration Protocol (DHCP), etc.

2.2.6 6LoWPAN-ND

The traditional IPv6-ND has many short comings if I consider the IoT environment. For example due to the lossy nature of wireless communication and change in radio environment, IPv6-link node-set may change due to external physical factors and Nodes appear to be moving without necessarily moving physically. Second, there are two types of link-layer addresses in LoWPAN, such as 16-bit short addresses and 64-bit unique addresses. Third, link-layer payload size is less than 100 bytes (as discussed in Section 2.2.5); thus, header compression is mandatory and very useful. Fourth, it was not designed for non-transitive wireless links. So due to this intermittent link behaviour along with heavy use of multicast make it inefficient and sometimes impractical. So considering these LoWPAN characteristics and IPv6-ND limitations, some optimizations and extensions are useful and necessary for LoWPAN and other homogeneous low-power networks. IETF presented two consecutive standards [130] and [143] to resolve the said shortcomings.

Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)-RFC6775

Optimization in IPv6-ND gives a new name to the protocol called 6LoWPAN-ND [130], which provides following the following optimizations and extensions to IPv6 Neighbor Discovery:

1. Eliminate periodic or unsolicited RA. But it may exchange between routers.

2. No need to perform DAD

   - If Extended Unique Identifier - 64 bits (EUI-64)-based IPv6 addresses are used.

   - DAD is optional if Dynamic Host Configuration Protocol version 6 (DHCPv6) is used to assign addresses.

3. New address registration mechanism between hosts and routers. That stops NS multicasting, support sleeping hosts, and enables same IPv6 prefix(es) to be used across 6LoWPAN. Moreover, it provides host-to-router interface for DAD.

4. New RA options such as 6LoWPAN Context Option (6CO), 6LoWPAN Capability Indication Option (6CIO), Authoritative Border Router Option (ABRO), PIO, and SLLAO

5. New DAD mechanism for multihop by using Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC).

6. New method to control the flooding of configuration changes in LoWPANs by using PIO, 6CO, and ABRO.

7. New default protocol constants are introduced, and some IPv6-ND protocol constants are tuned.

Registration Extensions for 6LoWPAN-ND-RFC8505

[143] is the successor standard of [130]. It modifies existing options and updates the associated behaviors. For example, it improves the registration technique in 6LoWPAN routers by updating Address Registration Option (ARO) into Extended Address Registration Option (EARO). It provides an enhancement to the registration capabilities and mobility detection for different network topologies, including the routing proxy neighbor discovery in an LLN by tailoring and updating option header called 6CIO[5]. This specification does not introduce any new options however it improves nodes energy and mobility along with Extended Duplicate Address Confirmation (EDAC) and Extended Duplicate Address Registration (EDAR) in route-over topology. This protocol works totally collapsed in the mesh-under topology, that I will discuss and show proves in the Chapter 5.

IPv6 Backbone Router-RFC8929

The standard [141] amends [130] and [143] to allow routing registrars known as “Backbone Routers” or “6LoWPAN Backbone Routers (6BBRs)” to proxy services for IPv6-ND. They are installed placed along a backbones wireless edge and federate several wireless links into a single Multi-Link Subnets

[5] This option header is tailored from [27]

(MLSN). It proposes an MLSN with a central backbone that federates edge (LLN) links, with each link offering its own protection against rogue access and packet tempering or replaying. The usage of conventional IPv6-ND on the backbone, in particular, necessitates the trust of all nodes and the prevention of rogue access to the backbone at all times.

2.2.7 RPL Routing Protocol-RFC6550

RPL is a proactive distance-vector IPv6 routing protocol which is designed for LLNs. LLNs are a class of network in which both the routers and their interconnect are constrained. It provides three mechanisms of routing:

1. Point-to-Multipoint traffic (from the central control point to the devices inside the LLN are supported.

2. Multipoint-to-Point traffic (from devices inside the LLN towards a central control point)

3. Point-to-Point traffic is also available.

Internet Assigned Numbers Authority (IANA) defines that RPL messages are carried via ICMPv6 messages with a type value of 155. The Code field identifies the type of RPL control message. There are five control messages used by RPL that form the Spanning tree such as DODAG Information Solicitation (DIS), DODAG Information Object (DIO), DODAG Advertisement Object (DAO), DAO-ACK, and Consistency Check.

RPL Security Modes

Confidentiality and integrity are supported by the RPL for the messages. It is designed such that link-layer mechanisms can be used when available and appropriate in case of their absence, RPL can use its own mechanisms. It has three basic security modes.

1. Unsecured: There are no security procedures in place when RPL control messages are transmitted. This mode does not imply that the RPL network is insecure; it could be meeting application security requirements using other existing security primitives, such as link-layer security.

2. Preinstalled: Nodes contain pre-installed keys that enable them to process and generate secured RPL messages before joining the RPL instance, as the mode name implies.

3. Authenticated: As in pre-installed mode, nodes have pre-installed keys, however the pre-installed key can only be used to join an RPL Instance as a leaf in this mode. Obtaining a key from an authentication authority is required to join an authenticated RPL Instance as a router. However, the method for obtaining this key is not specified in the specification.

There is a secure variant for each RPL message. Integrity and replay protection, as well as secrecy and delay protection, are provided by these variations. RPL networks are no different from other networks in terms of security. Yes, they are vulnerable to both passive and aggressive eavesdropping attempts. Because of the additional security constraints imposed by ad-hoc networks and their cost aims, these networks may be the most challenging to secure.

RPL Robustness

It is built to work with a variety of link layers, including confined, potentially lossy, or often used with highly constrained host or router devices, such as low-power wireless or PLC (Power Line Communication) technologies. In LLNs, it provides sufficient robustness such as:

1. *DODAGVersionNumber:* It is a sequential counter that the root increments to create a new version of a Destination-Oriented Directed Acyclic Graph (DODAG). The RPL version number changes if the topology changes, whether physically or logically (objective function changes). The `RPLInstanceID`, `DODAGID`, and `DODAGVersionNumber` is used to uniquely identify a DODAG Version. The DIO message is used to send this information. DIO spreads throughout the network, causing all other nodes to join the newly established topology.

2. *Destination Advertisement Object Acknowledgement (DAO-ACK):* If a child node wants to join a DODAG, it sends the DAO message to the root or parent node. On the reception of the DAO message (root or parent node), the DAO-ACK message is sent which tells the confirmation or rejection joining. Each DAO message holds the DAOSequence

number, echoed in the DAO-ACK by the recipient. It is used to correlate a DAO message and a DAO-ACK message. If the DAO-ACK message is lost in the path for any reason, then the child node sends another DAO message. This acknowledgment feature makes the network more certain, reliable, and robust.

3. *Checksum:* In ICMPv6, 16-bits are used to identify errors and RPL messages utilize ICMPv6 messages to share routing information. So, indirectly it provides the bit error check to RPL messages.

2.2.8 IPv6 Performance and Diagnostic Metrics (PDM) Destination Option-RFC8250

This standard defines optional headers included in each packet that give sequence numbers and timing information as a foundation for measurements in order to analyze performance issues. These measurements can be evaluated either in real time or afterwards. The PDM Destination Options header is defined in this standard. It also includes field limitations, computations, and the use of PDM measurements. It offers several advantages, such as real measure of actual transactions, the ability to span organizational boundaries with consistent instrumentation, not requiring time synchronization between session partners, and the ability to handle all transport protocols (TCP, UDP, the Stream Control Transmission Protocol (SCTP), etc.) consistently. PDM provides the ability to quickly determine if the (latency) problem is in the network or in the server (application). That is, it is a quick way to perform triage. Such important advantages are available but without securing the data.

2.3 Security Solutions in Standards

In this section, I would like to discuss my findings about the available security methods provided at each layer of TCP/IP stack by well-known standardized protocols.

2.3.1 MAC Layer Security (Layer-2)

I focus on the IEEE standards to meet security requirements as discussed in Chapter 1. IEEE developed the 802.15.4 standard for wireless sensor networks. They introduced three security modes that a device can enable or disable. Four security services are also provided, all of which work only in secure mode. Below are the details of the security modes and services.

- **Security Modes:** The following are the three security modes.

  1. *Unsecured Mode:* No security services are provided by devices operating in unsecured mode.
  2. *ACL Mode:* In this mode, restricted security services for communication with other devices. The higher layer may decide to reject frames depending on whether the lower layer MAC indicates that a frame should come from a particular device. Since in this mode, no cryptographic protection is provided in the sublayer MAC. The higher layer should implement other mechanisms to ensure the identity of the sending device. The service provided in Access Control List (ACL) mode is access control.
  3. *Secured Mode:* In this mode, the device is fully secured at the MAC level.

  The following are the security services that a device can achieve.

- **Security services :** The security mechanisms in this standard are based on symmetric keys provided by higher layer processes. The management and creation of these keys is the responsibility of the implementer. The security provided by these mechanisms requires that the keys are generated, transmitted, and stored in a secure manner.

  1. *Access Control Service:* Access control is a security service that allows a device to select the other devices with which it wants to communicate. When the access control service is provided for in this standard, a device shall maintain in its ACL a list of devices from which it expects to receive frames.
  2. *Data Encryption Service:* In this standard, data encryption is a security service that uses a symmetric cipher to protect data from being read by parties without the cryptographic key. The data can be encrypted with a;

     - Key shared by a group of devices (typically stored as the default key)

  - Key shared between two peers (typically stored in an individual ACL entry).

  In this standard [67], data encryption may be provided on Beacon Payloads, Command Payloads and Data Payloads.

3. *Frame Integrity Service:* Frame integrity is a security service that uses a Message Integrity Code (MIC) to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. integrity may be provided on Data Frames, Beacon Frames, and MAC Command Frames.

4. *Sequential Freshness Service:* It uses an ordered sequence of inputs to reject frames that have been replayed. When a frame is received, the freshness value is compared to the last known freshness value. If the freshness value is newer than the last known value, the check passed and the freshness value is updated to the new value. If the freshness value is not newer than the last known Freshness value, the check failed. This service provides evidence that the received data is newer than the last data received from this device, but it does not provide an accurate sense of time.

2.3.2 Network Layer Security-(Layer-3)

IPv6 has several benefits over IPv4 such as scalability, efficient multicast routing, better QoS support, easier autoconfiguration (no need for DHCP), built-in authentication and privacy support, flexible options and extensions, etc. Due to these vital benefits IPv6 is more suitable for IoT devices [42]. To make the IoT network heterogeneous, I need to provide IP connectivity for IoT devices. This will allow me to access IoT devices globally. Moving from a homogeneous to a heterogeneous network carries a high risk of violating network security. The security features of IPv6 are described in Security Architecture for Internet Protocol in RFC-2401 [75]. IPv6 provides two security headers that can be used individually or together, known as Authentication Header (AH) and Encapsulating Security Payload (ESP) [74]. For the Next Header field, 8 bits are defined in the standard. The Next Header fields 051 and 050 are used for AH and ESP. It would be wastage of time to go into more detail about the two services, since Internet Protocol Security (IPSec) is not intended for the constrained devices. Moreover, in addition to confidentiality and integrity, IPSec provides end-to-end security with authentication and replay protection services, but the problem is that it consumes more energy for processing and memory, which is not acceptable for the constrained devices. We must assume that all constrained devices in the IoT network are sufficiently secure at the second layer, such as IEEE802.15.4 standard provides a Advanced Encryption Standard (AES) security standard [67], or a light weight security protocol needed to secure data and services at network level.

2.3.3 Transport Layer Security (Layer-4)

The transport layer is responsible for end-to-end communication. Two well-known protocols are used to transfer the application data. It is a key layer in Open Systems Interconnection (OSI) and in the TCP/IP stack that ensures reliability in case of TCP. TCP is a connection-based and reliable protocol, while UDP is connectionless and unreliable. Reliability means that the data must be in-order on delivery. It depends on the application requirements which transport layer protocol must be chosen. Two types of security can be implemented in this layer, TLS for TCP traffic and Datagram Transport Layer Security (DTLS) for UDP traffic.

1. **Transport Layer Security (TLS):** It evolved from Secure Socket Layer (SSH), which was developed by Netscape. After that, IETF developed its first version 1.0 in 1999. The latest version is TLS-1.3 [116], which is used by many web service providers. TLS consists of two phases: The first is *handshake*, in which the two parties perform an authentication function and establish an encryption key for data transmission. The second is *Data Transfer* , in which the two parties use the encryption key to encrypt all transmitted data. It provides three main functions: *Encryption*, *Authentication*, and *Message Integrity.* It protects web applications from various attacks, such as data breaches and Distributed Denial-of-Services (DDoSs) attacks.

2. **Datagram Transport Layer Security (DTLS):** The main purpose of DTLS is to construct TLS over UDP [117]. Datagram transport does not require or provide reliability. In addition, unlike TCP, UDP does not provide connection and disconnection setups, congestion control, fragmentation, flow control, rate control, and back-haul capability

checking features. For many IoT use cases, these functions are not required, but can be implemented at the application level if needed.

In an IoT environment where most networks are wireless and ad-hoc, UDP traffic may perform poorly due to channel access, intermittent connectivity, packet loss, congestion, and packets not being received properly. In this case, TLS cannot perform well. Considering features of fragmentation and reassembly on the other side is particularly important and necessary in an IoT environment. In addition, TLS cannot be used directly in datagram environments for the following five reasons [118] [147]:

- TLS does not allow independent decryption of individual records. Since integrity checking depends indirectly on a sequence number, integrity checking of record N+1 is based on an incorrect sequence number and therefore fails if record N is not received. DTLS solves this problem by adding explicit sequence numbers.

- The TLS handshake is a cryptographic handshake with a locking function. Messages must be sent and received in a specific order. This is incompatible with reordering and loss of messages.

- Not all TLS 1.3 handshake messages (such as the NewSessionTicket message) are acknowledged. Therefore, a new acknowledgement message must be added to detect message loss.

- Handshake messages are potentially larger than any given datagram, thus creating the problem of IP fragmentation.

- Datagram transport protocols such as UDP are susceptible to abusive behavior in the form of denial-of-service attacks against non-participants and require back-channel capability checking to be built into the handshake using cookies.

Two parameters have been added to the TLS for datagram transmission to address these issues. The first is the sequence number, while the second is the Epoch field. Beginning with the TLS Record Layer, the sequence number field permits decryption of individual records (no steam ciphers, such as RC4), and the epoch field is introduced to identify key changes in the TLS Record Layer.

2.4 Ongoing Standards

In this section, I discuss work that is not yet standardized but still in progress to become standardized. Research about IPv6 over different link layer technologies (constrained node networks) are in progress by different IETF working groups. PDM which was totally unsecured in its first release [45] but now it is implementing the robust security framework such as HPKE with some additional network measuring metrics.

2.4.1 Hybrid Public Key Encryption (HPKE)

This ongoing research work is performing by the Crypto Forum Research Group (CFRG) in the Internet Research Task Force (IRTF). This work provides HPKE as a full framework where it uses encapsulation instead of the encryption of the symmetric key. Traditionally we encrypt the symmetric key with the public key but in the HPKE method, we can generate the symmetric key and its encapsulation with the public key. HPKE is robust and flexible framework because it includes KEM [81, 85], KDF [81] and AEAD [44, 111], which gives to the user to change the key sizes and way more functionalities [23]. It provides four modes such as;

1. Encryption to a public key.
2. Authentication using a pre-shared key,
3. Authentication using an asymmetric key.
4. Authentication using both a pre-shared and an asymmetric key.

Because this is a work in progress, not all KEMs will support all authorized variations. They implement the system with commonly used and efficient primitives including Elliptic Curve Diffie-Hellman key agreement, KDF based on HMAC message authentication code (HKDF), and Secure Hash Algorithm2 (SHA2). The detail discussion about HPKE and its functional operations discussed in chapter 7.

2.4.2 IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option

As discussed in section 2.2.8 that IPv6 Performance and Diagnostic Metrics Version1 (PDMv1) (i.e., [45]) defines an optional Destination Option (DO) header embedded in each packet to provide sequence numbers and timing information as a basis for measurements. Since this data is sent in clear text, this may allow malicious actors to obtain information for subsequent attacks. So in the recent work presented in [46] defines a lightweight handshake (registration procedure) and encryption to secure this PDM data. I am working on the security model which is based on the HPKE. The details of the work will be discussed in Chapter 7.

2.4.3 IPv6 over Constrained Node Networks (6lo) Applicability and Use-cases

The applicability of IPv6 over 6lo node networks is addresses in [62], which also includes implementation examples. Various connection layer technologies, such as ITU-T G.9959 (Z-Wave), Bluetooth Low Energy, DECT-ULE, MS/TP, NFC, and PLC, are utilized as examples in addition to IEEE Std 802.15.4. This ongoing work is aimed at a group of people who want to learn about and assess operating end-to-end IPv6 over constrained node networks for local or Internet access. If this practical work would become successful even though the network relies on [143] and [158] which are already facing SPoF problem.

2.4.4 Transmission of IPv6 Packets over PLC Networks

Power Line Communication (PLC), has been widely used to support Advanced Metering Infrastructure (AMI), particularly smart meters for energy. The ongoing work in [63], describes how IPv6 packets are transported over constrained PLC networks, such as ITU-T G.9903, IEEE 1901.1 and IEEE 1901.2.

2.4.5 Transmission of IPv6 Packets over Near Field Communication

Near Field Communication (NFC) is widely used in smartphones and other portable devices to establish radio communication by touching them together or bringing them into very close contact. It is based on current Radio Frequency Identification (RFID) standards such as ISO/IEC 14443 and FeliCa, and encompasses communications protocols and data exchange formats.

The [37] defines that how IPv6 is transmitted over Near Field Communication (NFC) using 6LoWPAN techniques by considering ISO/IEC 18092 and those defined by the NFC Forum.

2.4.6 IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP

The ongoing work in [55] specifies a method which enable IPv6 mesh over Bluetooth Low Energy links established by using the Bluetooth Internet Protocol Support Profile. But the work does not define any routing protocol to be used in an IPv6 mesh over Bluetooth LE links.

2.5 Conclusion

Approximately all of the link-layer technology are based on the tree topology, where a root node is managing and controlling data and control messages. Moreover, established and ongoing works are constantly facing the SPoF issue by utilising [143] and [158] on the top of any link layer. The standard [141] gives a solution by deploying multiple 6BBRs at the edge of LLN. But the availability of the network remains critical because the link-layer technologies are based on tree topology. Most of the ongoing work use Registration Ownership Verifier (ROVR) which is derived from the link-level device address. In case of address spoofing, any node connected to the network and aware of a registered address to ROVR mapping could perform address theft and impersonation attacks. For such attacks [141] provides address-protected neighbor discovery method.

Chapter 3

Toward Resilient Wireless Sensor Networks: A Virtualized Perspective

The paper that I discuss in this chapter has been published in the Sensors as part of the Special Issue Energy-Efficient Resource Allocation for beyond 5G and IoT Systems and is available online: https://www.mdpi.com/1424-8220/20/14/3902. Investigation of multiple standardized protocols by IETF and IEEE are discussed in the context of the availability and resilience of the IoT networks. Moreover, results ensures the investigated availability problem and an architecture is advise to improve the availability of the IoT networks.

3.1 Introduction

The system of either directly or indirectly interrelated computing devices, including mechanical, electrical, electronics or digital machines and animals or people, connected to human daily life is known as the IoT or IoE. IoT has the unique feature of generating and transferring data over the Internet without requiring human-to-human or human-to-computer interaction, enabling distributed, resilient and autonomous systems. The application of IoT in society spans different areas, such as commercial (medical and health-

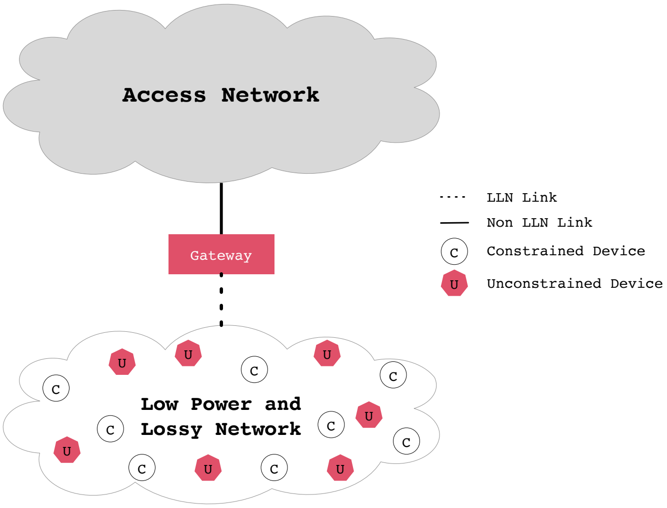


Figure 3.1: Abstract view of elements of interest in the Internet of Things (IoT) paradigm.

care, transportation, Vehicle-to-everything (V2X) communications, building and home automation), military (battlefield and ocean of things), infrastructure (metropolitan scale deployments, energy management and environmental monitoring), consumer (smart home and handicapped), and industrial applications (agriculture and manufacturing).

IoT is primarily driven by low-cost constrained devices, in which the main constraints are in the memory and computational capacity available to each device. Moreover, IoT devices are usually meant to be run on a battery or by energy-harvesting: thus, low energy consumption is the main goal. In order to keep the energetic cost to the minimum, IoT devices use specialized communication protocols, called LLNs.

Figure 3.1 shows a logical representation of a generic WSN. A gateway device is required at the edge of the WSN domain to provide Internet connectivity to all of its attached constrained and unconstrained devices. Unconstrained devices usually have more capabilities and access to more reliable power sources.

As with conventional networks, the essential security requirements (confidentiality, authenticity, integrity, accountability and availability) are also mandatory for WSNs [136]. In particular, they might be different depending on the application scenario; however, they are always important. A security breach can lead to severe consequences, ranging from the loss of users' personal data to the safety of whoever relies on the WSNs capabilities. The last point is especially important in applications such as Smart Cities or Industrial Internet of Things (IIoT). Securing devices and their communications in a distributed way on a large scale, preserving information security on the collected data and controlling information are very challenging and important issues that cannot be neglected.

In this chapter, I analyze the core requirements of IoT systems, while outlining the major critical points which have not yet been addressed by the standards and proposing an architecture to mitigate the most severe risks that are actually affecting IoT networks. The proposed architecture generally corresponds to the others already discussed in standards, but there are many different features that I highlight that can improve the resilience of the IoT systems.

The rest of the chapter is organized as follows: in the next section, I explain different scenarios and their requirements in terms of their availability, scalability, security and management. In Section 3.5, applicable standards for the IoTs with the integration of classical standards are discussed. The proposed architecture is finally presented in Section 3.7. Finally, I highlight several open issues and directions for future work in Section 3.8.

3.2 Motivation

The IoTs has been one of the main focus areas of the research community in recent years, the requirements of which help network administrators to design and ensure the functionalities and resources of each device. Generally, two types of devices—constrained and unconstrained devices—are typical in the IoT environment. Devices with limited resources—for example, sensors and actuators—are known as constrained devices. Unconstrained devices includes gateways or border routers. Such devices are challenging in terms of their deployment because of their connectivity, channel selection, multiple interfaces, local and global address assignment, address resolution, remote access, mobility, routing, border router scope and security. To deal with these services, the availability of the IoT system ensures that the desired network services are available even in the presence of denial-of-service attacks, and the use of the system has become a difficult but mandatory task for network designers. To this end, I present a novel design for WSNs to address these challenges by shifting mandatory functionalities from unreliable to reliable and stable domains.

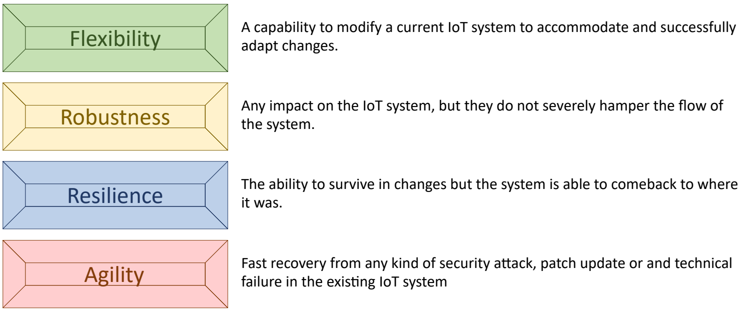
3.3 Contribution

The main contribution of my work consists in addressing the core network requirements for IoT systems and pointing out several guidelines for the design of standard virtualized protocols and functions. In addition, I propose a novel architecture which improves IoT systems, lending them more resilience and robustness, together with highlighting and some important open research topics.

- I move the core functionalities of protocols—i.e., IEEE 802.15.4 [67], 6LoWPAN Border Router (6LBR) [130, 143] and the Virtual DODAG root [158]—from an uncertain network to a more stable and controllable network.
- I achieve the coordination of egress points (Egress Points (EPs)) (i.e., gateways/edge routers), which perform the functionality of the 6BBR [142] as well as the DODAG root.
- I use the Virtual DODAG for the Fog Layer, which helps in the coordination among multiple DODAG roots.
- I use the Fog Layer to increase scalability and help constrained devices achieve time-critical applications.
- I synchronize multiple EPs; furthermore, I ensure network services and claim that my architecture is more resilient.

3.4 Service Requirements

IoT networks are an enabling technology for the so-called Fourth Industrial Revolution (Industry 4.0), Smart Cities, Smart Health, etc. It is envisioned that IoT systems will be based on heterogeneous network types, along with Cloud and Fog-based systems [136]. As a consequence, I can expect a wide



A capability to modify a current IoT system to accommodate and successfully adapt changes.

Any impact on the IoT system, but they do not severely hamper the flow of the system.

The ability to survive in changes but the system is able to comeback to where it was.

Fast recovery from any kind of security attack, patch update or and technical failure in the existing IoT system

Figure 3.2: Main capabilities used to improve availability.

range of technologies; e.g., Fifth Generation (5G), Bluetooth (802.15.1), LiFi (802.15.7r1), Wireless Fidelity (WiFi) (802.11), WPANs (802.15.4), Ethernet, etc. Despite the wide variety of systems, IoT systems will almost always share the same general concept shown in Figure 3.1, where a number of unreliable devices are connected to the Internet, providing a globally reliable service by means of redundancy, cooperative functions, etc. As an example, multihop systems are based on the idea that, even if a node fails, other paths can be quickly established.

In the present chapter, I am interested in the overall system reliability; i.e., the determination of the actual problems that can affect the service as a whole. Without loss of generality, I can list the following requirements that are common to any IoT system.

3.4.1 Availability

As with conventional networks, the essential security requirements—i.e., confidentiality, authenticity, integrity, accountability and availability—are also mandatory for IoT systems. Availability ensures that the desired network services are available even in the presence of Denial-of-Service (DoS) attacks. As mentioned in Figure 3.2, it is important for network administrators and designers to consider this requirement by ensuring flexibility, robustness, resilience and agility capabilities.

3.4.2 Scalability

This requirement describes the ability to scale the entire network by ensuring stability and competitiveness when demand is increased. According to [29], one network size hardly fits for all scenarios, since requirements are typically time-variant. However, one size of the network cannot solve all future market demands. It may become difficult and challenging to ensure the same services with the same QoS after scaling the existing network. Ensuring the scalability requirement in IoT systems also become very challenging when dealing with heterogeneous devices, because the heterogeneity of devices encompasses different applications, protocols, mobility, hardware etc.

3.4.3 Security

As in conventional networks, the five security functions—i.e., confidentiality, authentication, accountability, availability and integrity—are also mandatory for IoT systems [136]. I discuss availability independently, because it becomes very challenging in SPoF scenarios that I highlighted in Section 1. IoT is certainly the most complex and still open area of network security because embedded devices are concerned with vulnerabilities and there is no good way to patch them. The chip manufacturers adopt its firmware and software to optimize their incentives. Device chips are selected by the vendors based on features and price, and the vendors perform certain manipulations if anything is required by the chip software and firmware; thus, only device functionality is their main focus. On the other hand, end users are usually not able or allowed to patch the system, or if they can, then they have limited information about when and how to patch. As a result, hundreds of millions of Internet-connected devices in the IoT are vulnerable to attacks [126]. Sensors have definitely become vulnerable, because they allow attackers to inject malicious code and data into the network. This is also a threat for actuators, where the attacker can manipulate the operation of machinery and other devices in the IoT system.

3.4.4 Management

This is a process of administrating and controlling devices and their functionalities in the entire IoT ecosystem. Generally, there are two categories of management: first is network management, where the network administrator performs analysis and maintains the performance of entire IoT system, and the second is network security, where security-related process are managed.

Network Management

Network management is widely considered one of the hard problems of networking and continues to be the focus of much research even in IoT systems. The management of hundreds of millions of heterogeneous, small objects becomes very challenging due to their heterogeneous nature. This includes performance management, fault analysis, location management, mobility management, the provisioning of QoS and any kind of service management.

Network Security Management

Securing and controlling heterogeneous IoT devices is very challenging, because IoT security operation and management includes routine IoT security evaluation, security logs and the automatic identification of the security events based on the best practice polices. The security management platform (centralized or distributed) can be provided for policy configuration, policy orchestration and policy execution. Nevertheless, it is essential to ensure homogeneous security polices for the entire IoT system.

3.5 Relevant Standards for IoT

In this section, I discuss the standard technologies that are relevant for my scenario. As stated previously, I can partition the network from a topological point of view into two main parts:

- The wireless segment, in which devices use unreliable communication systems and are mostly resource-constrained, and
- The wired segment, where I can assume I am able to use high-speed, reliable, and secure networking technologies.

Referring to Figure 3.1, gateway devices reside between the two parts (wired and wireless). Without loss of generality, I can assume that the wired section is reliable and secure, thanks to the use of the Ethernet, Software-Defined Networkings (SDNs), network virtualization, etc. In contrast, I will assume that the wireless segment is a multihop IoT system, enabled by any

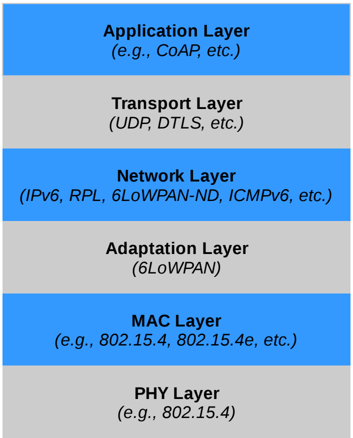


Figure 3.3: IoT-oriented IP protocol stack

multihop technology; e.g., Bluetooth Mesh, IEEE 802.15.4, etc. For the present discussion, however, the exact physical and MAC standard is not relevant, as my focus is on upper layers.

In the following, I will assume that the IoT system is based on IP standards; in particular, IPv6. As shown in Figure 3.3, the IP stack for IoT systems is largely identical to the "normal" IP stack, with some notable exceptions that can affect the system resilience.

The use of IPv6 for resource-constrained networks and devices can lead to some inefficiencies. For this reason, a set of "adaptation" protocols have been defined by the 6Lo IETF Working Group. In particular, 6LoWPAN [64, 105] allows the compression of the IPv6 header, while 6LoWPAN-ND [130] mainly helps in optimizing Neighbor Discovery. 6LoWPAN is (almost) a stateless compression system, and it does not cause particular issues; in contrast, 6LoWPAN-ND can lead to network failures, as I will show below. Another point worth analyzing is the routing protocol, as it can be a potential source of issues.

3.5.1 Neighbor Discovery Optimization

Traditional IPv6-ND [109] is used for router discovery, address resolution, DAD and redirecting messages, along with prefix and parameter discovery. 6LoWPAN-ND optimized this protocol for the LLNs [130]. In an

LLN, devices are classified according to their role: 6LoWPAN Node (6LN), 6LoWPAN Router (6LR), and 6LBR. A 6LN is a “normal” device, while 6LR refers to devices which are able to relay messages and 6LBR is a device responsible for managing the network. A very important difference between IPv6-ND and 6LoWPAN-ND is that IPv6-ND is completely distributed, while 6LoWPAN-ND is a centralized protocol: all the IPv6-ND functionalities normally using multicast messaged are substituted by a request–reply mechanism between 6LNs and the 6LBR, with 6LRs acting as relays.

The introduction of a request–reply system helps the LLN, as multicast messages are not efficient in LLN and they might be not supported at all in some architectures. On the other hand, a centralized entity is a potential security and reliability problem.

It is worth noting that 6LoWPAN-ND is also responsible for network prefix dissemination, DAD and IP address registration, and the network parameters must be periodically refreshed. As a consequence, each 6LN must have a stable connection with the 6LBR. An erratic connection or an incorrect configuration of the 6LoWPAN-ND timers can lead to network failures.

3.5.2 Network Routing

Routing is usually the critical point in a resilient network. As a matter of fact, the Internet is also “resilient” thanks to its decentralized routing approach. However, the SDN paradigm has recently emerged as a viable alternative to “classical” routing approaches. SDN enables, among other things, better and finer-grained traffic engineering, faster network reconfiguration, per-flow routing, etc.

No matter which approach is used—SDN or IP routing—the implications for network resiliency are evident: a failure in a forwarding device might jeopardize the network. For this reason, it is worth recalling the basic ideas behind both approaches.

- **IP-based routing:** IP routing heavily depends on the type of network being considered. In the wired part, there are well-known protocols, such as RIP, OSPF, EIGRP, etc., whose resilience has been extensively studied. As a matter of fact, the resilience is in this case almost entirely dependent on the routing protocol convergence time.

  Wireless multihop networks require special routing protocols, as every node (not only the routers) actually participate in the forwarding scheme. Although many routing schemes has been proposed for multihop and ad-hoc networks, RPL [158] is a routing protocol built to fulfill the specific IoT scenarios. As a consequence, I will focus on RPL in the following discussion.

In RPL, the network topology is oriented toward a sink (or, in RPL terms, a root node). All the paths are built to originate from the sink node, and its existence is central for the whole network (RPL enables also other kind of paths, but for the sake of brevity, I will not consider them in the present discussion). Although RPL builds redundant network paths to prevent node and link failures, and although the recovery time from network disruption is limited, the root node is a potential issue. It is possible to have multiple different RPL roots in an IoT network, but this is not usually a good solution, as it increases the network complexity and the memory requirements in the nodes.

- **SDN forwarding:** The SDN approach decouples the routing function from the forwarding function. In an SDN network, all the forwarding nodes only keep a forwarding table, which is built by a centralized controller. The controller creates the forwarding tables by having a complete topological knowledge of the network and possibly other data such as the switches' queue occupancy, link resource utilization, etc.

  The SDN approach was initially proposed for wired networks, and it is now a well-known technology used in many scenarios. Due to the benefits offered by SDN, there have been several recent proposals to extend it to wireless links [53]. Nevertheless, one central requirement for SDN is to have a secure, reliable and fast link between any switching device and the SDN controller. This limitation makes its applicability in wide multi-hop networks quite problematic, if not entirely impossible.

3.6 Related works

As discussed in Section 3.4, network resilience depends on four elements: availability, scalability, security and management. In order to fulfill these requirements, each network element should be resilient to a variety of attacks, and the network itself should be self-healing. As noted previously, a single IoT device can be attacked; however, a powerful network design should prevent a case in which a single device failure can disrupt the network as a whole. As a consequence, I believe that the very first requirement is to avoid or mitigate the presence of an SPoF in the network.

The most recent research contributions toward network resilience in IoT networks are summarized in Table 3.1. We considered here only works focusing on protocols above layer 2, and I will assume, without loss of generality, that an unmodified IEEE 802.15.4 standard [67] is being used. This assumption is justified by the fact that using a “custom” L2 protocol usually leads to increased device costs and long-term device resupply issues.

Table 3.1: State-of-the-art contributions and limitations.

Paper	**Year**	**Contribution**	**Av**	**Sc**	**Se**	**Ma**	**SPoF**
[108]	2020	A central control that jointly manages end-to-end, both the wired segments and the Industrial IoT domain.	○	◐	○	◐	○
[145]	2020	An aggregator based RPL for an IoT-Fog based power distribution system with 6LoWPAN.	◐	●	○	◐	◐
[15]	2019	SD-NFV based architecture to reduce the end-to-end delay and strengthen energy depletion in motes.	○	○	○	◐	○
[50]	2019	Provide multi-gateway synchronization protocol ByzCast to increase data availability and improves fault and intrusion tolerance in WSNs.	◐	◐	●	◐	●
[102]	2018	NFV based Operating room Innovation Center (OPIC) to investigate time and non-time critical applications.	◐	◐	○	◐	◐

Table 3.1. ***Cont.***

Paper	Year	Contribution	Av	Sc	Se	Ma	SPoF
[83]	2018	Provide synchronization scheme for multiple gateways to increase network capacity and proposed a scheme to reduce the energy waste and TSCH enhancement.	◐	●	○	◐	◐
[51]	2018	Provide interoperability in home automation system for Fog computing applications based on MQTT and ZigBee-WiFi Sensor Nodes.	○	◐	○	◐	○
[70]	2018	A role based security controller architecture to strengthen the security of IoT.	○	○	●	◐	○
[25]	2018	An end-to-end indoor air quality monitoring (IAQM) system provide interoperability and backup support in case of connection failure IP or radio.	◐	◐	○	◐	○
[124]	2017	Provide synchronization algorithm for multiple gateways to increase the availability and reliability of critical applications.	◐	◐	○	◐	◐
[71]	2017	Cost effective scheme for the selection of gateways and adaptation mechanism is used to increase the system capacity to cope dynamic change.	◐	○	○	◐	○

Table 3.1. ***Cont.***

Paper	Year	Contribution	Av	Sc	Se	Ma	SPoF
[86]	2016	Provide energy-efficient services, fault tolerance, load balancing and resource management.	◐	◐	◐	●	○
[56]	2016	An hierarchical SDN approach provide security and handle communications between clusters by an SDN cluster head managed by an SDN controller.	◐	●	◐	●	◐
[93]	2016	Cloud-based security architecture for medical WSNs, where Access Control supports complex and dynamic security policies.	○	◐	◐	◐	○
[76]	2015	Overly architecture for WSN based fire monitoring system that relies on a constrained application protocol, where a single WSN is shared by multiple applications.	○	◐	○	◐	○
[133]	2015	Secured cross layer architecture for IoT to improve security management by Adaptive Interface Translation Table (AITT).	◐	○	●	◐	○
[26]	2015	A new scheme that provide address configuration and context management and their distribution in 6LoWPAN-based architecture.	◐	◐	○	◐	○

Table 3.1. *Cont.*

Paper	Year	Contribution	Av	Sc	Se	Ma	SPoF
[110]	2015	Architecture facilitate interoperability, support low power operations, offers service discovery, registration and authentication mechanisms for IoT.	○	◐	◐	◐	○
[54]	2015	Provide a mechanism for fault tolerance at gateway and provide traffic management to avoid gateway being a bottleneck.	○	◐	○	◐	○
[94]	2015	A new design of gateway with the integration of 6LoWPAN adapter layer in a Network Adapter Driver (NAD) of computer.	○	○	○	◐	○
[53]	2015	A stateful approach to make programmable sensor nodes by reducing the amount of information exchanged between sensors and SDN controllers.	○	◐	○	◐	○
[157]	2015	Efficient-Neighbor Discovery that advertise reachability to a registered addresses and BBR solve the problem of node mobility.	○	◐	○	◐	○
[30]	2015	An automatic monitoring and tracking system for patients, biomedical devices within hospitals, that provide visibility of the motes and perform information management.	○	○	○	◐	○

Table 3.1. ***Cont.***

Paper	**Year**	**Contribution**	**Av**	**Sc**	**Se**	**Ma**	**SPoF**
[114]	2014	Analyzed different solutions for the integration of WSNs and Internet and provide Gateway solution for localization and tracking application.	○	○	○	◐	○
[104]	2014	A model for an Area Sensor Network (ASN) that connects heterogeneous networks and provide interoperability & scalability.	○	●	○	◐	○
[58]	2014	Provide dynamic and distributed load balancing scheme for multiple gateways to achieve global load fairness, network capacity, and reliability.	◐	◐	○	◐	◐
[31]	2014	Architecture for smart campuses and focused on data collection from sensors and its storage in the Cloud.	○	◐	○	◐	○
[43]	2014	Architecture based on multiple GWs and improve ND proxy, routing support, mobility and reliability for data delivery in 6LoWPANs.	●	◐	○	●	◐

Av = Availability, **Sc** = Scalability, **Se** = Security, **Ma** = Management, **SPoF** = Single Point of Failure;

○= no, ◐= partial, ●= yes.

From Table 3.1, I can see that the SPoF issue is still an open problem.

On the other hand, fulfilling the network requirements that I addressed in Section 3.4 is ongoing research because it depends on dynamic changes in the market. These requirements are directly proportional to the market demands [136]. In the literature review, our selection criteria spanned SPoF, core network requirements and different emerging technologies such as SDN, Network Function Virtualization (NFV), and the Fog/Cloud.

The architectures presented in [43,50,56,58,83,102,124,145] addressed the SPoF problem and provided solutions to increase the scalability and improve the availability; however, the rest of the papers did not address this problem and adopted different mechanisms to ensure other network requirements.

Some proposals, such as the work presented in [15, 53, 56, 70, 108], were facilitated by the SDN approach to improve the network requirements, but only the work presented in [56] presented a novel idea to solve the SPoF. In addition to SDN technology, in [15] , the authors took NFV technology into account and tried to improve the end-to-end delay without considering the importance of the Fog Layer for IoT devices. Instead of Fog and SDN, in [102], the authors utilized the NFV and presented a limited solution for the SPoF problem as well as achieving network requirements other than security.

The work presented in [51, 93, 145] deployed the Fog Layer near the IoT domain; in this way, they solved the network management problems and increased the scalability. Some research contributions highlighted other aspects in current standards; for example, in [157], the node registration process is dealt with and mobility issues solved by introducing backbone boarder routers BBRs at the edge of the LLN. In contrast, in [26], the authors introduced a node address configuration method and provided a context (CO) dissemination scheme within 6LoWPAN. In [43], the authors closely addressed the same SPoF problem in the standard [130] that I am addressing regarding the synchronization of 6LoWPAN gateways to solve the SPoF and other services, but their solution is protocol-dependent, which limits its scalability.

3.6.1 Why neither SDN or “Classical” Approaches can be the Solution (Alone)

SDN and classical protocols have their respective limitations. As a matter of fact, the SDN controller, PAN Coordinator, 6LBR and DODAG Root are all SPoFs, and their operations are vital for a system’s resiliency. There are no standardized protocols for the North–South and East–West interfaces;

moreover, the network becomes more unreliable and vulnerable when more than one functionality is running on a single node.

SDN is widely considered to be a possible method to improve network reliability. However, the whole reliability is based on the assumption that a reliable and secure channel is available between the SDN controller and the SDN switches. This assumption is valid for wired networks, where TCP and TLS can be used. In the IoT domain, this cannot be assumed; moreover, the SDN controller itself must be located in a high-security zone to prevent attacks on its availability.

All classical protocols in [67, 130, 143, 158] are standardized, but they basically based on the *Tree* topology, where a single root is responsible for all the management services; for this reason, they suffer from multiple SPoF elements, leading to a global lack of resiliency.

We believe that both approaches can be used to ensure network resiliency, as they are effective on different network segments. However, when used in tandem, they can successfully ensure network resiliency.

3.7 Proposed Architecture

As discussed in previous Sections, the main issues in LLN reliability are related to devices whose functionality is essential for the network management (routing, address management, etc.). A targeted attack on one of these devices can jeopardize the network. Our proposal aims at mitigating this risk by moving the critical functionalities into a section of the network which is easier to manage and control and by providing redundancy and resiliency through virtualization [127]. In this way, I am also able to solve the SPoF problem and achieve the core requirements previously discussed. In this section, I present our architecture, properly integrating current standards, and some necessary discussion to strengthen the 6LoWPAN architecture.

Without loss of generality, I will focus on the 6LBR, RPL Root, and PAN Coordinator functionalities. Traditionally, these three functions are implemented in the same device, which also acts as a gateway between the LLN and the Internet. Even though these functionalities might be split in different devices, this does not help to increase the resiliency of the network. In contrast, it creates a burden for the management, as multiple devices might be compromised, and the loss of even one of them will affect the whole system.

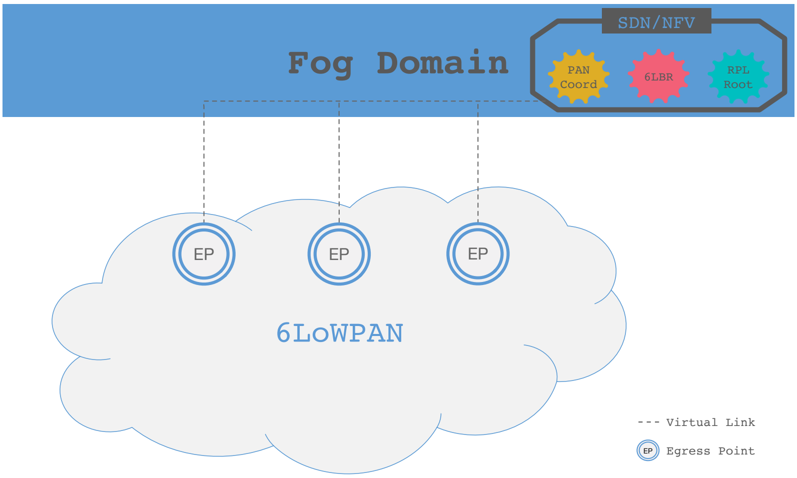


Figure 3.4: Fog-based LLN architecture.

Our proposal relies on moving all the critical functionalities in the Fog domain, where they can be properly protected from attacks, and they can obtain more resilience thanks to the use of the NFV. Our architecture is based on three levels; i.e., the Fog domain, access network domain and LLN domain.

As shown in Figure 3.4, EPs are installed at the edge of the LLN domain. Each EP is connected to the Fog domain through an access network, which can be assumed to have low latency and high reliability. Now, I will present the functionalities and requirements against each domain.

3.7.1 LLN Domain

The LLN Domain is composed of all the LLN devices, and I assume that it uses the latest IETF proposals for LLNs. This includes, but is not limited to, 6LoWPAN [64, 105], 6LoWPAN-ND [130, 143] and RPL [158]. These standards ensure proper resilience in the network, provided that the physical topology of the network allows redundant paths for the EPs.

To this end, it is important to ensure that the failure of a small portion of the devices leads to network partitioning. In other terms, the physical network topology should be as meshed as possible, eventually deploying nodes in which the network does not satisfy the necessary redundancy. To analyze this case, it is possible to apply graph theory; e.g., to ensure that there are no nodes that have a betweenness centrality [28] significantly different than the others. Under these assumptions, it is safe to consider the network as a whole as resilient, because even if a device is removed, the routing can quickly recover from the failure. Inside the LLN, security and resiliency might also be improved by using MAC-level encryption, address shuffling [112] and other techniques to prevent attacks on single devices. However, I consider these aspects as beyond the scope of the current discussion.

3.7.2 Access Domain

The role of the this domain is to connect EPs to the Fog domain. Without loss of generality, I can assume that it can be considered as more reliable and secure by keeping in mind the nature of the LLN and its related standards. As an example, the access domain might rely on 5G, Ethernet or any system that allows the EPs to be connected to the Fog domain. The only requirement for the access network is to be able to configure a virtual point-to-point link between each EP and the Fog-based network functions. This can be accomplished though well-known network management procedures and can involve the use of SDN, encrypted tunnels, etc.

The EPs play a central role in the proposed architecture, which is shown in Figure 3.5. With respect to a normal LLN EP, where it acts as a gateway between the LLN and the Internet, in our architecture, the EP simply forwards the packets from the LLN to a set of devices implementing the necessary functionalities (i.e., PAN Coordinator, 6LBR, RPL root, etc.) within the Fog domain. We refer to these in the following as LLN root functions, and they will be discussed in Section 3.7.3.

From an architectural point of view, EPs behave as the first hop set of nodes connected to the LLN root functions. To this end, it is necessary to have the following:

1. A virtual 802.15.4 interface.
2. A reliable and secure link between the LLN node and the LLN root functions.

The virtual 802.15.4 interface is necessary to enable seamless communication between the EPs and the devices implementing the LLN root functions.

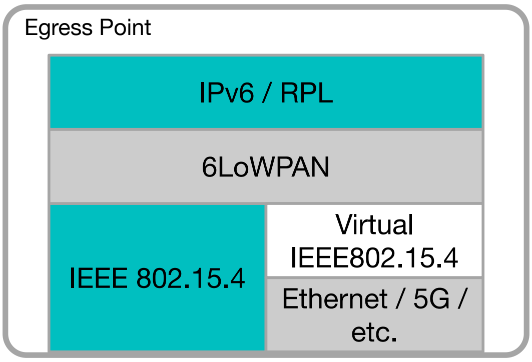


Figure 3.5: Egress Point architecture

The link between the EP and the LLN root functions can be, for example, a direct Ethernet link or, in a more complex environment, an encrypted virtual link; i.e., a Virtual Private Network (VPN) tunnel.

The IEEE 802.15.4 standard-based frames, which originated from the virtual IEEE 802.15.4 interface, are tunneled through the above-mentioned link toward the LLN root functions, where they are handled as if they originated from a normal LLN device.

3.7.3 Fog Domain

This domain is built and managed according to the best practices for Fog systems and can be enhanced by using SDN and NFV technologies. In particular, SDN can help in providing flexible and resilient network management, link resilience and traffic engineering, while NFV allows us to replicate and dynamically allocate resources to the LLN root functions, implemented as NFV elements. As a result, I can safely assume that the LLN root functions implemented in the Fog are scalable and resilient. Moreover, thanks to the computational capacity of the Fog domain, I can assume that the LLN root functions are protected through proper security systems; i.e., firewalls and Intrusion Detection Systemss (IDSs).

The LLN root functions implemented in the Fog domain are described below.

Virtual PAN Coordinator

The tasks of the PAN Coordinator (i.e., devices association and disassociation, beacon generation, etc.) are implemented in the Fog and are practically identical to those of a “real” PAN Coordinator. Its connection to the real LLN is ensured by the EPs, which behave as LLN nodes.

Virtual 6LBR

The role of the 6LBR is to bridge the 6LoWPAN network (where IPv6 headers are compressed) and the IPv6 network. Toward this end, the virtual 6LBR behaves as a normal 6LBR thanks to the virtual IEEE 802.15.4 interface present in the EPs.

Virtual RPL Root

Similar to the virtual 6LBR, the virtual RPL Root is practically identical to a traditional RPL root. The only difference is in how the metrics of the links between the root and the EPs are measured.

The virtual links between the RPL root and the EPs are different from the links present in the LLN. As a consequence, it is necessary to define how the metrics [151] of these links are collected. As an example, the “hop count” metric might be kept identical, while the “link reliability” metric might require a new definition.

To further enhance the routing resiliency, it is possible to implement the virtualization suggested RFC 6550 [158], where a virtual RPL root is implemented in the non-LLN section of the network and each gateway between the LLN and non-LLN domain is orchestrated by the Virtual RPL root. The standard proposes two scenarios, as shown in Figure 3.6. Our proposed architecture fully enables the first scenario foreseen in the standard, shown in the figure as Scenario 1. On the contrary, Scenario 2 (which is also foreseen by the standard) does not seem to enhance the network resiliency, as the RPL root (also acting as a virtual root) cannot be fully protected by the Fog domain.

Virtual 6LoWPAN Backbone Router

A 6BBR [142] enables seamless connectivity between an LLN and an IPv6 network, behaving as a routing registrar that provides proxy-ND services [143].

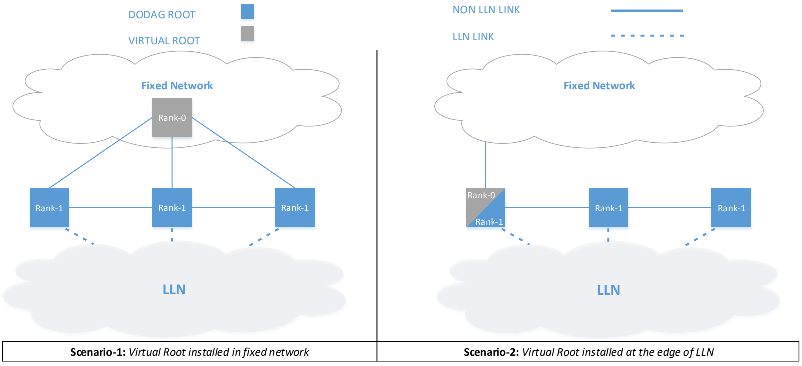


Figure 3.6: Virtual DODAG Root placement scenarios

The use of a 6BBR allows us to avoid the use of a prefix for each LLN, and I believe that its use will be widespread in future LLN deployments. In the proposed architecture, the 6BBR can also be implemented as a network virtual function, further increasing the reliability and scalability of the large enterprises such traffic management, health and medicine, agriculture etc.

3.7.4 Architecture Evaluation

To evaluate the effectiveness of our architecture, I performed some tests using the well-known ns-3 simulator (`https://www.nsnam.org`). In the simulation setup, I considered two scenarios: the first simulation scenario is shown in Figure 3.7, which is, as highlighted in Section 3.6, the classical conventional network layout; the second scenario represents our proposed architecture, and it is shown in Figure 3.8.

Table 3.2: Simulation parameters.

Parameter type	**Value**
Radio Range	About 100 m
802.15.4	Beaconless, always on
Propagation Model	Log-Distance
6LoWPAN Compression	IPHC - RFC 6282
RPL Constants	as per RFC 6550

The conventional network layout (Figure 3.7) is indeed composed of one gateway, acting as PAN Coordinator, 6LoWPAN endpoint and RPL root, while the proposed network layout (Figure 3.8) includes multiple EPs and an NFV based node in the Fog domain performing the PAN Coordinator, 6LoWPAN endpoint and RPL root functions. The simulation parameters to evaluate both scenarios are shown in Table 3.2.

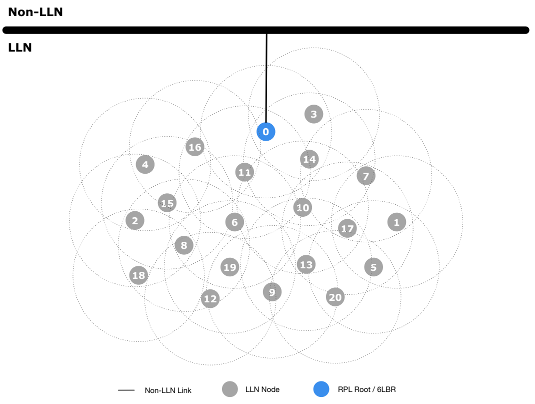


Figure 3.7: Conventional network setup.

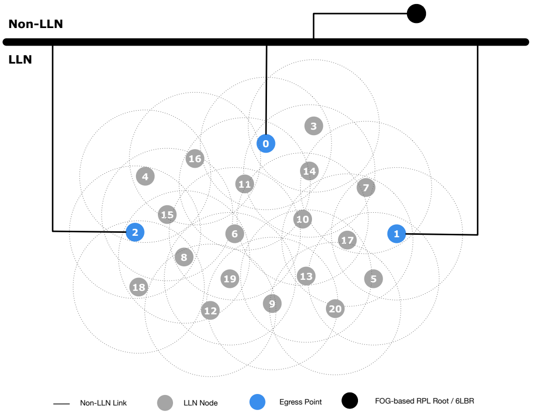


Figure 3.8: Proposed network setup.

In order to simulate an attack (or a device failure), I added some artificial noise to the signal received and sent by the device being attacked. This is

Table 3.3: Conventional setup, with nodes ranked over time.

	Root										**Sensor Nodes**										
Time [s]	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	1	-	-	2	-	-	-	-	-	-	-	2	-	-	2	-	-	-	-	-	-
2	1	4	-	2	4	5	3	3	4	-	3	2	-	-	2	4	3	4	5	-	-
3	1	4	-	2	4	5	3	3	4	-	3	2	-	-	2	4	3	4	5	-	-
4	1	4	-	2	4	5	3	3	4	-	3	2	-	4	2	4	3	4	5	-	-
5	1	4	-	2	4	5	3	3	4	-	3	2	-	4	2	4	3	4	5	-	-
6	1	4	-	2	4	5	3	3	4	5	3	2	-	4	2	4	3	4	5	-	-
7	1	4	-	2	4	5	3	3	4	5	3	2	-	4	2	4	3	4	5	-	-
8	1	4	-	2	4	5	3	3	4	5	3	2	-	4	2	4	3	4	5	-	-
9	1	4	6	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
10	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
11	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
12	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
13	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
14	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
15	1	4	5	2	4	5	3	3	4	5	3	2	5	4	2	4	3	4	5	4	5
											Root Node is Attacked										
16	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

equivalent to adding a jammer in the proximity of the attacked device or to physically tampering with the device antennas. The noise is powerful enough to prevent any signal being correctly sent or received by the device under attack.

In the first experiment, I used the conventional network setup as shown in Figure 3.7, where Node 0 (i.e., Gateway) is connected to the Internet. The attack is performed at Node 0 at second 15. After the attack, as shown in Table 3.3, all the sensor nodes become unreachable.

In the second experiment, I used the proposed network setup shown in Figure 3.8, where Nodes 0, 1 and 2 are EPs, and the NFV based node in the Fog domain is connected through reliable non-links. The results are presented in Table 3.4. In this experiment, I attacked the EPs (Node 2, 1 and 0), respectively, at seconds 15, 30, and 45. To better understand the network recovery time, I increased the RPL DODAG version after the attack detection. This effectively detached all the nodes from the root and forced a new join procedure. As shown in the table, the network quickly recovers a fully functional state after each failure, and it becomes unavailable only when the last EP is successfully attacked.

The tests fully confirm the effectiveness of the proposed architecture. As a matter of fact, in the conventional network setup (Figure 3.7), an attack on the gateway node completely disables the whole LLN. Moreover,

Table 3.4: Proposed network setup, with nodes ranked over time.

	Root										**Sensor Nodes**											
Time [s]	**NFV**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	1	2	2	2	-	-	-	-	-	3	-	-	3	-	-	3	3	-	3	-	-	-
2	1	2	2	2	3	-	3	-	-	3	-	-	3	-	-	3	3	-	3	-	-	-
3	1	2	2	2	3	-	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	-
4	1	2	2	2	3	-	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	-
5	1	2	2	2	3	-	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	-
6	1	2	2	2	3	-	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	-
7	1	2	2	2	3	4	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	5
8	1	2	2	2	3	3	3	4	-	3	5	4	3	-	4	3	3	4	3	3	4	4
9	1	2	2	2	3	3	3	4	3	3	5	4	3	-	4	3	3	4	3	3	4	4
10	1	2	2	2	3	3	3	4	3	3	5	4	3	-	4	3	3	4	3	3	4	4
11	1	2	2	2	3	3	3	4	3	3	5	4	3	5	4	3	3	4	3	3	4	4
12	1	2	2	2	3	3	3	4	3	3	5	4	3	5	4	3	3	4	3	3	4	4
13	1	2	2	2	3	3	3	4	3	3	5	4	3	5	4	3	3	4	3	3	4	4
14	1	2	2	2	3	3	3	4	3	3	5	4	3	5	4	3	3	4	3	3	4	4
15	1	2	2	2	3	3	3	4	3	3	5	4	3	5	4	3	3	4	3	3	4	4
											EP 2 is Attacked											
16	1	2	2	2	3	-	-	-	-	-	-	4	-	-	-	-	-	-	3	-	-	-
17	1	2	2	2	3	-	-	-	-	-	5	4	-	-	4	-	-	-	3	-	6	-
18	1	2	2	2	3	-	-	-	-	-	5	4	3	-	4	-	-	-	3	-	6	-
19	1	2	2	2	3	5	3	-	3	-	5	4	3	-	4	-	5	4	3	-	6	-
20	1	2	2	2	3	5	3	-	3	-	5	4	3	6	4	-	5	4	3	-	6	-
21	1	2	2	2	3	5	3	-	3	-	5	4	3	6	4	-	5	4	3	-	6	-
22	1	2	2	2	3	5	3	-	3	6	5	4	3	6	4	4	5	4	3	7	6	4
23	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	4	5	4	3	6	5	4
24	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
25	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
26	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
27	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
28	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
29	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
30	1	2	2	2	3	5	3	4	3	5	5	4	3	6	4	3	5	4	3	6	5	4
											EPs 1 and 2 are Attacked											
31	1	2	2	2	-	-	-	-	-	-	-	-	3	-	-	3	-	-	-	-	-	-
32	1	2	2	2	3	-	-	-	-	-	-	4	3	-	-	3	-	-	5	-	-	-
33	1	2	2	2	3	-	-	4	-	-	-	4	3	-	-	3	-	-	5	-	-	-
34	1	2	2	2	3	-	6	4	4	-	6	4	3	-	5	3	-	-	5	-	5	-
35	1	2	2	2	3	-	6	4	4	-	6	4	3	-	5	3	-	-	5	-	5	-
36	1	2	2	2	3	-	6	4	4	-	6	4	3	-	5	3	-	-	5	-	5	-
37	1	2	2	2	3	5	6	4	4	-	6	4	3	-	5	3	-	4	5	-	5	-
38	1	2	2	2	3	5	6	4	4	6	6	4	3	6	5	3	5	4	5	7	5	6
39	1	2	2	2	3	5	6	4	4	6	6	4	3	6	5	3	5	4	5	7	5	6
40	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
41	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
42	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
43	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
44	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
45	1	2	2	2	3	5	6	4	4	5	6	4	3	6	5	3	5	4	5	6	5	6
											EPs 0, 1, and 2 are Attacked											
46	1	2	2	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
47	1	2	2	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

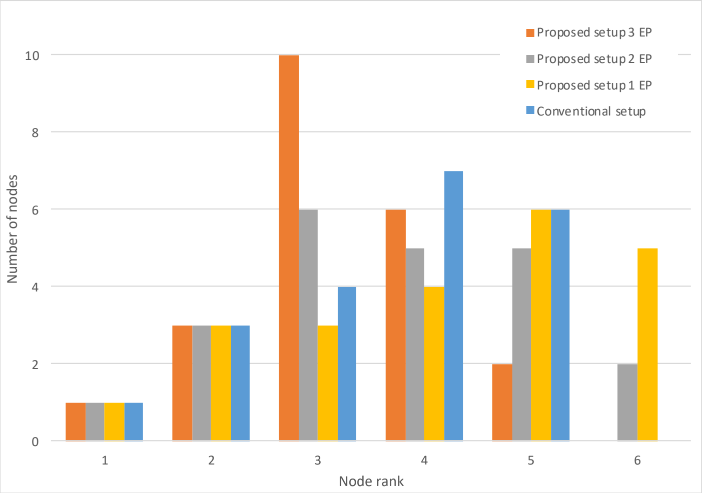


Figure 3.9: Histograms of node ranks when VDR is not located at the Fog layer

since the LLN nodes are disconnected, they could start to perform recovery mechanisms (i.e., find a new network), enabling the attacker to perform a secondary attack such as a gateway impersonation. On the contrary, in our proposed architecture, an attack on one (or more) EPs has limited or no effect on the network operations. It is not necessary to remark that an attack on all the EPs is far more difficult to perform, and an adverse event (i.e., a failure) involving all the EPs at the same time is even less likely.

To date, the standard [158] defined only rank 1 for the root node and rank 0 is not defined. First experiment results as show in the Figure 3.9, all the EPs have the same rank i.e., one. When the network converges at node rank 3 most of the nodes are visible to the root nodes. In this experiment the Scenario-2 from the Figure 3.6 has considered. At node rank 5, conventional setup and proposed setup with 1 EP has the same equal number of nodes connectivity.

Figure 3.10 which is our proposed setup with multiple EPs along with Virtual DoDAG Root (VDR) installed separately at fog layer (consider the Scenario-1 of the Figure 3.6), shows the node rank histogram with a varying number of working EPs. It is evident that the introduction of the EP functionality has another benefit on the network, other than increased resiliency:

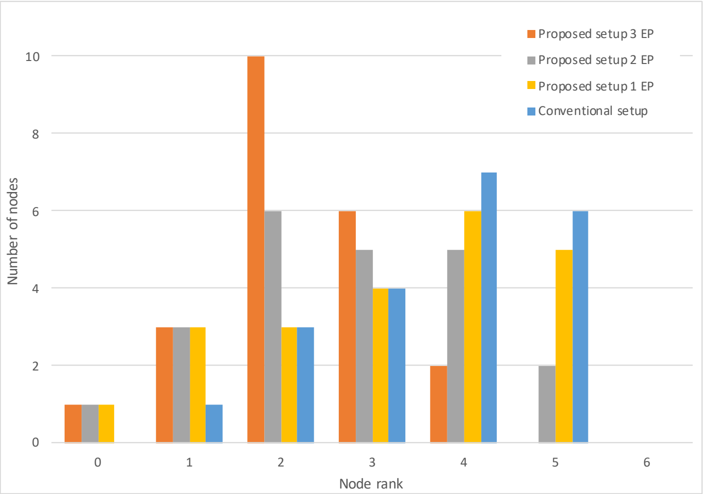


Figure 3.10: Histograms of node ranks when VDR is located at the Fog layer

the LLN maximum rank is smaller, with more nodes concentrated on ranks very close to those of the EPs (rank 2 nodes). This has many potential benefits for the network: less energy consumption, less latency, etc. As a matter of fact, the possibility to use EPs goes beyond the pure network resilience, as they can become both an optimization parameter in the LLN (EP number and position), and a further security measure; e.g., by using "sleeping" EPs which are activated when an attack is detected. One point that has not been studied in the present work, and that will be the subject of future research, is the network convergence time. RPL (which is the major driver for the network stability and convergence) is an extremely complex protocol, whose convergence is dependent on multiple parameters and which can be influenced by multiple factors (e.g., variable-period trickle timers, network failure detection methods, traffic patterns, network topology, objective functions, etc.). As a consequence, the study will have to involve multiple scenarios with varying numbers of nodes in different topologies (both synthetic and real).

3.8 Conclusion

The utilization of IoT devices in our social life and ongoing research work is rapidly increasing, but this work also brings some important challenges. Some of them are required to fulfill the demands of the market, and they need to be addressed in way that strengthens IoT systems in the design process while developing it. All the related standards, IEEE 802.15.4 [67], 6LoWPAN-ND [130, 143] and RPL [158], are based on the Tree topology, where the "root" node is responsible for all management services. Beside this, when a single node—i.e., a gateway—becomes a root node for an LLN, then it represent a single point of failure. Keeping in mind this problem, our proposed architecture provides significant scalability and availability by connecting multiple EPs to the Fog domain. The Fog domain also gives the opportunity to fulfill the network and security managements.

Our proposed architecture provides a way to deploy a resilient network to fulfill the core requirements and solution of decoupling of different services of root nodes. The simulation results fully confirm the validity of the approach. Moreover, the proposed architecture enables enhanced security countermeasures; e.g., by using "ghost" EPs, which are activated when an attack is detected.

Chapter 4

Is 6LoWPAN-ND necessary? (Spoiler alert yes)

The LLNs are based on constrained devices. Energy conservation is one of the main constraint and the traditional IPv6 Neighbor Discovery was not designed nor suitable to cope it, because non-transitive wireless links, and the use of heavy multicast transmission make it inefficient and sometimes impractical (or outright impossible) in a LLN. A delicately work has been done by the IETF to optimize the IPv6 Neighbor Discovery (NDP) protocol, and the result is 6LoWPAN-ND. Unfortunately, 6LoWPAN-ND is not usually implemented in the commercial, open source or proprietary IoT operating systems or simulators. In this chapter, I explore if there is a real need of 6LoWPAN-ND protocol for a LLN. What would be the benefits or drawbacks if we implement this protocol? What will happen if we do not adopt this protocol for LLNs? And why it has not been widely adopted?

4.1 Introduction

According to the market analysis forecasts, IoT systems are going to experience exponential growth. Even without citing one forecast, in particular, the usual figure is in the range of billions of devices in the next 10 years. Contrary to computer systems, or even smartphones, IoT devices are usually small devices, with limited resources, low cost, and long operative life. As counter-intuitive it might be, the long operative life is a problem, as wrong or suboptimal design choices can not simply be phased out by “natural ob-

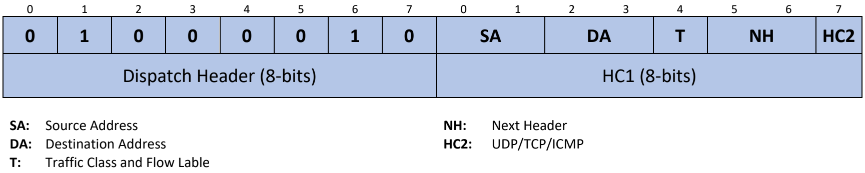
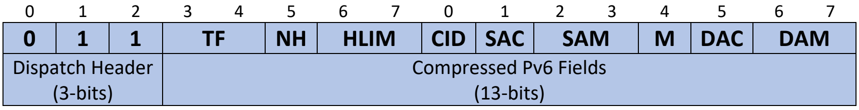


Figure 4.1: 6LoWPAN-HC1

solescence", and software patches are a problem as well, due to the limited support from vendors. Hence, it is important to evaluate carefully what protocols are optimal in what scenarios. Avoiding this kind of analysis leads to systems that are "working", but exhibits issues in terms of scalability, uneven resource consumption, etc.

IoT systems can be roughly split into two broad categories: devices equipped with an IPv6 stack, and devices needing a *Gateway* to be connected to the Internet. In this chapter, we will focus on the first kind. Among the IoT devices using the IPv6 stack 'natively', further classification can be made according to the kind of network they can use. Devices that can use 'IPv6-friendly' networks, like WiFi or 5G, and devices using the so-called LLNs, like Bluetooth, IEEE 802.15.4, Long Range Wide Area Network (LoRaWAN), etc.

A LLN might have a number of differences, both evident, and less evident, from a normal network. The most relevant (for the present discussion) are 1) short frames - usually unable to carry efficiently IPv6 packets, 2) potential lack of efficient support for multicast or broadcast frames (or no support at all), 3) potential lack of uniqueness of MAC level addresses in the LLN, etc. In order to overcome the limitations imposed by LLNs, IETF Working Groups (WGs) (in particular the 6LoWPAN and 6lo Working Group) devised a number of protocols. The most known is the 6LoWPAN, a *shim* layer, as shown in the Figure 4.4 is hiding the LLN MAC layer from the IPv6 layer, offering, for example, header and packet compression. This shim layer also called the 6LoWPAN *Adaptation Layer*. It solves the problem of IPv6 MTU requirement by providing the facility of fragmentation [64, 105]. The 6LoWPAN-HC1 [105], as shown in the Figure 4.1 is partially obsoleted and to cope its limitations, 6LoWPAN-IPHC [64], as shown in the Figure 4.2, provides compression techniques for link-local, global, and multicast IPv6 addresses. It also supports stateless and state-full encoding.



TF:	Traffic Class and Flow Lable Field	**SAC:**	Source Address Compression
NH:	Next Header	**SAM:**	Source Address Mode
HILM:	Hope Limit	**M:**	Multicast Compression
CID:	Context Identifier Extension	**DAC:**	Destination Address Compression
		DAM:	Destination Address Mode

Figure 4.2: 6LoWPAN-IPHC

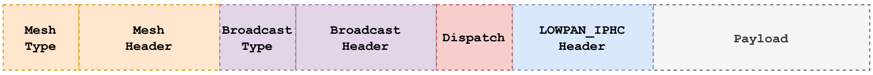


Figure 4.3: A LoWPAN encapsulated LOWPAN_IPHC compressed IPv6 datagram.

But it is obligatory that a LoWPAN encapsulated LOWPAN_IPHC compressed IPv6 datagram that requires both mesh addressing and a broadcast header to support mesh broadcast or multicast must follow the sequence as illustrated in the Figure 4.3. Moreover, ESC Dispatch Code points and guidelines can be found in [32].

Later on IETF developed the Neighbor Discovery protocol for 6LoWPANs known as 6LoWPAN-ND [130, 143]. Because the conventional IPv6-ND [109] is based on heavy multicast and made for non-LLNs and it becomes inefficient and sometimes impractical for LLNs due to their energy conservation. Moreover, IPv6-ND was not designed for non-transitive wireless links. Beside peculiarities of IPv6 over LLN MAC Layer 802.15.4, The IPv6 provide several benefits e.g., Reachability along with scalability of LLN devices towards the Internet, ability of interconnectivity to other IP networks including the Internet [84], auto-configuration mechanism, called SLAAC [140]. The IPv6 address format allows subsuming of IEEE802.15.4 addresses, as explained in [105] and [64]. Such vital features of the IPv6, LoWPAN known as 6LoWPAN in literature. In this chapter, my discussion encompasses in terms of both protocols reliability and robustness. For this reason I explain what is 6LoWPAN-ND and why there is a need for this protocol in LLNs? What are the benefits or drawbacks if network administrator adopt it or not?

The rest of the chapter is organized in this way; In Section 4.2, in-

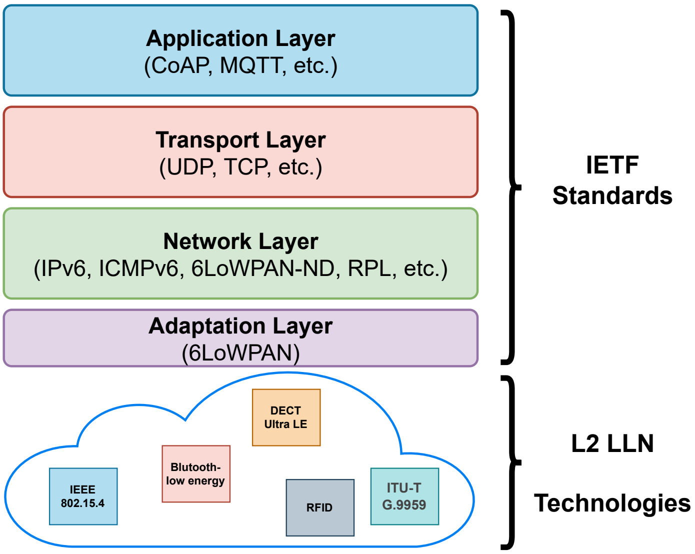


Figure 4.4: IETF 6lo Stack

depth discussion encompasses to address the said questions. Moreover, I also made our survey on open-source, proprietary, and commercial tools (e.g., simulators and operating systems) that have implemented 6LoWPAN and 6LoWPAN-ND or even consider its implementation, discussed in Section 4.3. The Section 4.4, discusses the complexity of 6LoWPAN-ND, for example node categories, Neighbor Cache Entry (NCE) states, and their optimal variations and *address registration mechanism* that encompasses link-local and global addresses registration.

In the Section 4.5, I explain the simulator description, simulation scenarios along with obtained results and detailed discussion. In the last Section 4.7, I discuss both ND protocols in terms of their reliability and robustness.

4.2 6LoWPAN-ND Standard

The ultimate communication between nodes (hosts and routers) requires the link-layer address. For this reason, nodes use *Neighbor Discovery (ND)* to determine the link-layer addresses of their neighbors. Hosts also require to use ND to find neighboring routers that are willing to forward packets on their behalf. Moreover, nodes use the ND protocol to deliberately keep track of their neighbors reachability or unreachability states and accordingly maintain their NCE's.

IPv6 based non-LLNs networks use the classical IPv6-ND reactive protocol that relies heavily on multicast operations. It provides multiple core dynamic functionalities to form a network, e.g., *Router Discovery, Prefix Discovery, Parameter Discovery, Address Resolution, NUD, DAD*, *Redirect*, and etc,. All operations achieved with the help of ICMPv6 [38] and with the utilization of specific multicast addresses. As discussed in the Section 4.1 about the LLN devices characteristics, wireless links are sometimes non-transitive and nodes are energy conservative, so the IPv6-ND perform inefficient and impractical to deal with LLN devices.

The *Neighbor Discovery Optimization* for 6LoWPAN's, shortly called 6LoWPAN-ND based on two IETF standards RFC6775 [130] and RFC8505 [143][1]. In addition to IPv6-ND ICMPv6 options, 6LoWPAN-ND defines more new options. It categorise three types of devices namely 6LN, 6LR, and 6LBR. It also defines two types of topologies *mesh-under* and *route-over*, Moreover, for route-over topology it present an extension to DAD message. The 6LoWPAN-ND protocol carries the functionalities of IPv6-ND but in an optimised way to work over LLNs.

The main functions of 6LoWPAN-ND are to proactively establish the Neighbor Cache Entry (NCE) in the 6LN's and 6LBR and to prevent address duplication. For this reason, it introduces a new unicast *Address Registration* mechanism with the help of NS and NA by appending EARO, that came up with reducing the use of multicast messages compared to the classical IPv6-ND protocol. For route-over topology, it defines the ICMPv6 typed DAR and DAC messages between 6LR's and the 6LBR.

4.2.1 6LoWPAN-ND Node Types

Standard define there kind of node types where 6LR and 6LBR are the routing nodes, and 6LN (only) is a non-routing node. 6LN term also used for 6LR, but when they don't act as a router, for example in mesh-under topology.

1. **6LoWPAN Node (6LN):** Any host or router participating in a LoW-

[1] RFC-6775 is partially obsoleted by RFC-8505.

PAN. This term is used according to the situation, either a host or router. For example, in mesh-under, all 6LR's considered as a host, instead of a router.

2. **6LoWPAN Router (6LR):** An intermediate router having the ability to forward and route IPv6 packets and also send and receive RA's and RS's. They only participate in route-over topologies.

3. **6LoWPAN Border Router (6LBR):** A border router located at the edge of 6LoWPAN. Its responsibility includes to disseminate network configuration information. It is an authoritative node and it propagates information to the 6LoWPAN network it is serving e.g., PIO, 6CO, ABRO, 6CIO and registration of the global addresses.

4.2.2 6LoWPAN-ND Network Topology

6LoWPAN-ND defines two types of topologies, mesh-under and route-over, where 6LBR is the central repository of all the Registered Addresses in its domain and the source of truth for uniqueness and ownership. We will see the difference between both topologies and follow of messages.

1. **Mesh-under:** A topology where all nodes are connected to a 6LBR through a mesh using link-layer forwarding. In this topology, all IPv6 hosts in a LoWPAN are only one IP hop away from the 6LBR. As depicted in the Figure 4.5, 6LN sends RS messages for the router discovery similar way presented in IPv6-ND protocol, but in response it receives a unicast RA message along with several node configuration options. The PIO and SLLAO options tailored from the IPv6-ND protocol. On successful configuration, 6LN sends NS messages for the address registration, this message also gives the knowledge of address duplication. Once the 6LBR complete registration then it will send NA message to 6LN with a status code. There are several status codes regarding the registration phase that can be found in [143]. For example address duplicate, Cache full etc,. It is obligatory for 6LN to perform a second registration against it global address, so again the same procedure NS(SLLAO)/NA(EARO with Status) will be called but this time for global address registration. Once the registration is done then data communication can be start between both nodes.

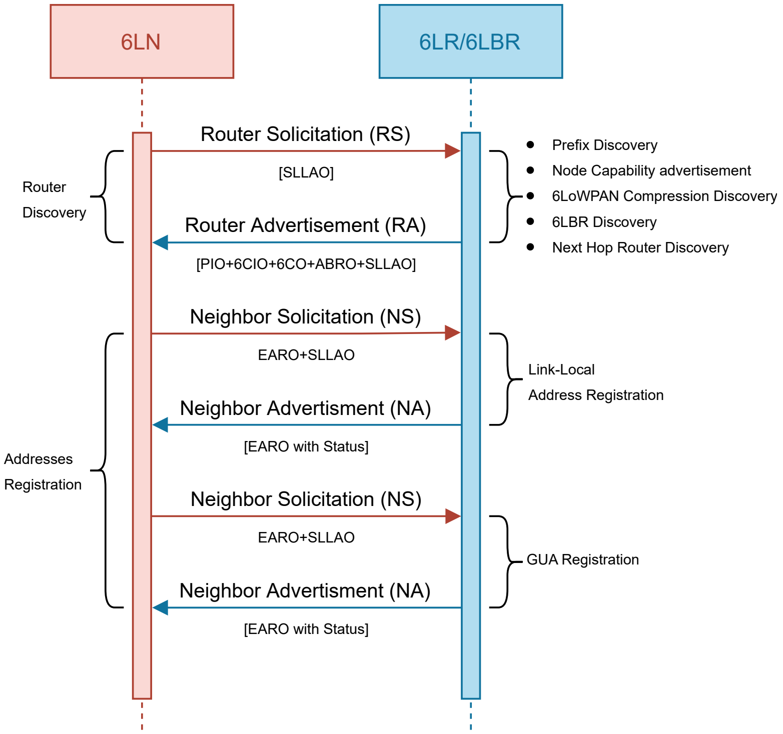

6LN
6LR/6LBR
Router Solicitation (RS)
[SLLAO]
Router Advertisement (RA)
[PIO+6CIO+6CO+ABRO+SLLAO]
Neighbor Solicitation (NS)
EARO+SLLAO
Neighbor Advertisment (NA)
[EARO with Status]
Neighbor Solicitation (NS)
EARO+SLLAO
Neighbor Advertisment (NA)
[EARO with Status]
Router Discovery
Addresses Registration
Prefix Discovery
Node Capability advertisement
6LoWPAN Compression Discovery
6LBR Discovery
Next Hop Router Discovery
Link-Local Address Registration
GUA Registration

Figure 4.5: Message exchange between 6LN and 6LR/6LBR in Mesh-under

2. **Route-over:** In this topology all hosts are connected to the 6LBR through the use of intermediate IP routing. Hosts are typically multiple IP hops away from a 6LBR. The route-over topology typically consists of a 6LBR, a set of 6LR's, and hosts.

   In this topology, 6LN act same as discussed above, but this time 6LR will also perform the GUA registration with 6LBR node. This registration accomplish with the help of DAR and DAC ICMPv6 message exchange. When 6LR sends the DAR it copies the EARO fields into DAR message, and append its own link layer address (LLA) as an option (SLLAO), instead of 6LN, as shown in the Figure 4.6. Once the registration become successful at 6LBR, then is will send a unicast to DAC message to 6LR. The 6LR also register the global address and send NA with a successful status code against NS that is previously re-

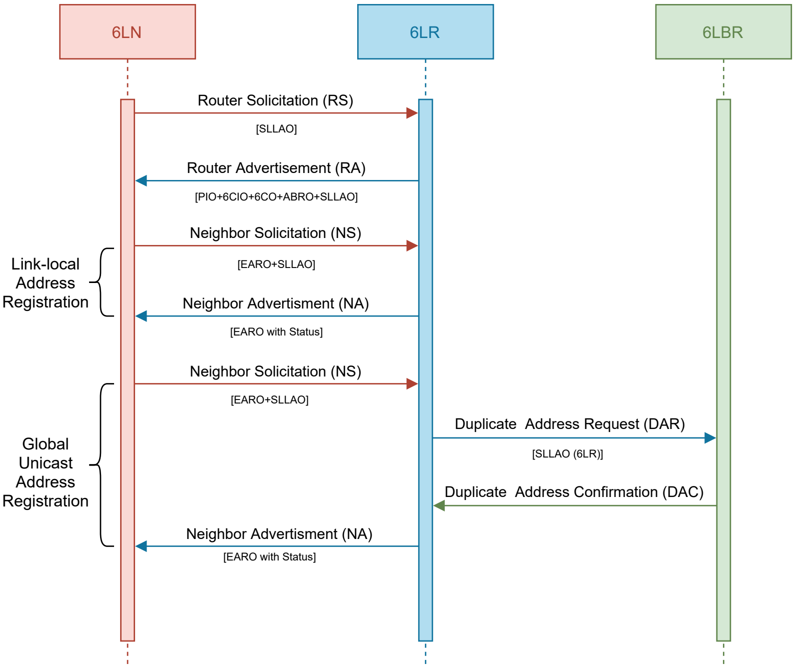

6LN
6LR
6LBR
Router Solicitation (RS)
[SLLAO]
Router Advertisement (RA)
[PIO+6CIO+6CO+ABRO+SLLAO]
Neighbor Solicitation (NS)
[EARO+SLLAO]
Link-local Address Registration
Neighbor Advertisment (NA)
[EARO with Status]
Neighbor Solicitation (NS)
[EARO+SLLAO]
Duplicate Address Request (DAR)
[SLLAO (6LR)]
Global Unicast Address Registration
Duplicate Address Confirmation (DAC)
Neighbor Advertisment (NA)
[EARO with Status]

Figure 4.6: Message exchange between 6LN and 6LR/6LBR

ceived from 6LN. These new link-local and global address registration mechanism introduced by the 6LoWPAN-ND and as a consequence, DAD is no longer necessary, and address resolution might be substituted by a request to the router.

4.2.3 New Neighbor Discovery Options

To deal with complexity of 6LoWPAN natures, there are some new options defined in [130] and [143]. The 6LoWPAN-ND also tailored previously defined options from [109] and [27] standards.

1. **Extended Address Registration Option (EARO):** Routers require direct reachability to their neighboring hosts through their set of IP addresses and corresponding link-layer addresses. This thing needs to be maintained as the radio reachability changes. EARO has the ability to accomplish this task by appending in unicast NS messages

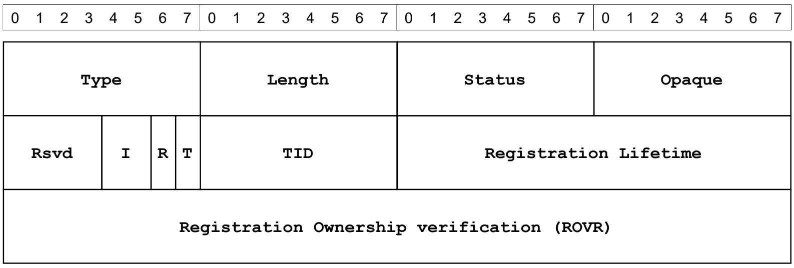


Figure 4.7: Extended Address Registration Option Format

sent by hosts. It also plays the role of NUD to determine that it can still reach a default router (See Section 4.4). It helps routers to maintain reliably NCEs. It is also included in corresponding NA messages with a Status field indicating the success or failure of the registration. In [143] introduces some new fields such as 2-bit *I-flag*[2], *R-flag*, *T-flag*, *Opaque*, *Transaction ID* (TID) and *Registration Ownership Verifier* (ROVR).

The I-field is a two-bit field. Only zero value is used to indicate that the Opaque field carries an abstract index that is used to decide in which routing topology the address is expected to be injected. In that case, the Opaque field is passed to a routing process with the indication that it carries topology information, and the value of 0 indicates default [143]. When R-flag is set, the Registering Node requests that the 6LR ensure reachability for the Registered Address. When it is not set, indicates that the Registering Node is a router and that it will advertise reachability to the Registered Address via a routing protocol (such as RPL [158]). The T-flag indicates the presence of the TID field. The Opaque field carries the information where the registration is relayed to another process, e.g., to advertised by a routing protocol. The TID field is used to know the current location of a registering mobile device. Whereas ROVR provides the correlation between multiple attempts to register the same IPv6 address. This option is always host initiated and its size varies from 64, 128, 192, or 256 bits. Its format is shown in the Figure 4.7.

[2] Currently, it works as a flag because all other values of are reserved and MUST NOT be used.

2. **6LoWPAN Context Option Format (6CO):**

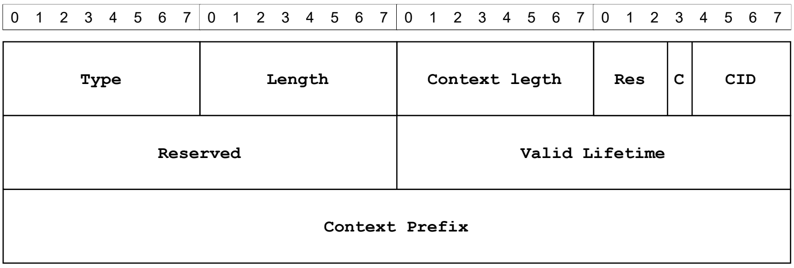


Figure 4.8: 6LoWPAN Context Option Format

It carries prefix information for LoWPAN header compression. The header compression applies to all IPv6 addresses. The Prefixes can be remote as well as local to the LoWPAN. Multiple contexts are identified by a CID field. Standard allows using prefix context of any length or an address (/128). But only up to 16 6COs may be carried in a unicast RA message. The format of the 6CO message is shown in Figure 4.8.

3. **6LoWPAN Capability Indication Option (6CIO):** Initially the 6LoWPAN Generic Header Compression (6LoWPAN-GHC) [27] standard defines this option, and later on [143] tailored it and defines five new capability bits for use in the 6CIO, for use in IPv6-ND messages. The G-flag indicates that the node is capable of GHC. The D-flag indicates that the 6LBR supports EDAR and EDAC messages. The L-flag indicates that the node is 6LR. B-flag indicates that the node is 6LBR. P-flag indicates that the node is Routing Registrar and E-flag indicates that the node is IPv6-ND Registrar, meaning that it supports registrations based on the EARO. The format of the 6CIO message is shown in Figure 4.9.

4. **Authoritative Border Router Option (ABRO):** It is required when RA messages are used to disseminate PIOs, 6CIOs, and context information across the entire route-over topology. To reliably add and remove prefixes from the 6LoWPAN, it is obligatory for all nodes to include this option along with PIO and 6CO. This obligation indicates that the authoritative 6LBR is adding and removing PIO and 6CO, otherwise the message will be discarded. Version number fields indicate

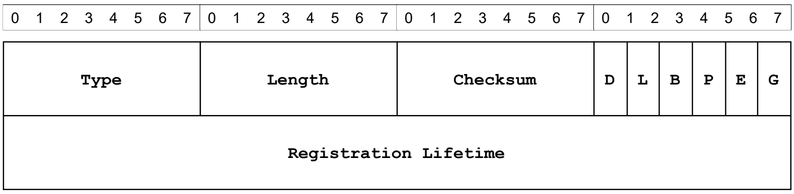


Figure 4.9: Capability Indication Option Format

the latest advertisement. With multiple 6LBRs participating in the 6LoWPAN, then they would have separate version number spaces and option needs to carry the IP address of the 6LBR that originates set of information. The format of the ABRO message is shown in Figure 4.10.

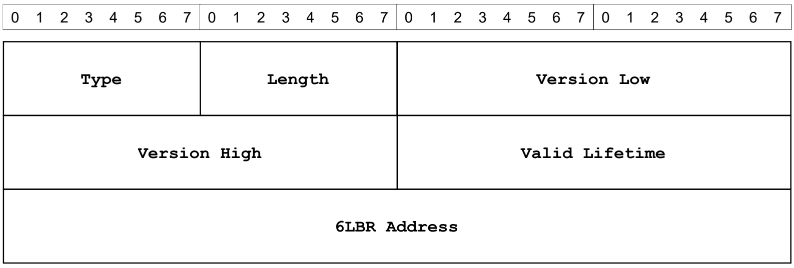


Figure 4.10: Authoritative Border Router Option Format

4.2.4 New Neighbor Discovery Messages

As per RFC4862 [140], Duplicate Address Detection (DAD) MUST be performed on all unicast addresses prior to assigning them to an interface, regardless of whether they are obtained through stateless auto-configuration, DHCPv6, or manual configuration. But in 6LoWPAN-ND standard nodes do the registration of their unicast link-local and global addresses by using NS(EARO+SLLAO)/NA(EARO+Status code) as shown in Figure 4.5 and Figure 4.6, so this procedure does not allow nodes to follow the old procedure, because registration procedure implicitly does the typical DAD operation. For multihop implicit DAD messages between 6LR and 6LBR, the standard introduces two new ICMPv6 messages DAR and DAC with new Type numbers 157 and 158, respectively. Both messages share a common base format as shown in Figure 4.11. The ICMPv6 Code fiels [38] for Duplicate Address messages is split into two 4-bit fields. First is known as *Code Prefix* and second is called *Code Suffix*. The use-case of the Code Prefix field is not yet defined so it remains zero and ignore by the receiving node. The Code Suffix defines four values. 1, 2, 3, and 4 denoting that node supports ROVR size of 64, 128, 192, or 256 bits. Remaining fields hold the same definition and processing as in the EARO.

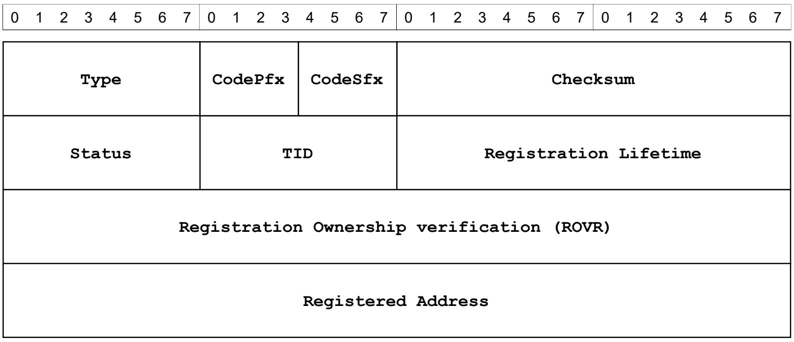


Figure 4.11: Extended Duplicate Address Message Formats

1. **Extended Duplicate Address Request (EDAR):** When 6LR receives the NS(EARO+SLLAO), it copies the EARO fields into EDAR and includes its own SLLAO, and sends it to 6LBR. The registered Address field carries the host IPv6 address from the IPv6 Source field in the NS that contained the EARO sent by the host. While sending EDAR, 6LR put its own unicast address into the IPv6 Source field.

2. **Extended Duplicate Address Confirmation (EDAC):** This message triggers in response to EDAR by the 6LBR. The Status filed holds the zero value that indicates the positive confirmation and for how much time the registration holds at the 6LBR, indicated in Registration Lifetime filed against the provided ROVR and Registered Address. On the reception of EDAC at intended 6LR, all fields copies from EDAC into EARO and issue a NA(EARO) to the intended host who initiated the request for address registration by sending NS(EARO+SLLAO). In case of registration failure, all possible steps should be followed according to the EARO Status Codes Table 1 mentioned in [143].

With this basic introduction and analogy of the 6LoWPAN-ND protocol with its proactive approach towards LLNs, now we are at this point that what if we don't use the 6LoWPAN-ND protocol then how classical IPv6-ND can help us with its reactive behavior to form a reliable and robust 6LoWPANs? What are the other benefits that 6LoWPAN-ND or IPv6-ND protocols are providing? What are their pros and cons in terms of reliability, robustness, performance, and overhead? For this reason, my next discussion throws some light on the importance of the 6LoWPAN-ND and its redounded and conflicting functions with other IETF standards.

4.2.5 Is 6LoWPAN-ND necessary?

Some work in literature and even in the *Simulators* and *Operating Systems (OS)* (Particular made for the IoT motes) do not implement the 6LoWPAN-ND protocol, and they use the upper layers logic to obtain the basic information that a 6LoWPAN-ND provides. For example, according to the Section 6.2 of [158], the DIS message is similar to RS as specified in IPv6-ND [109]. Meaning that a node may use DIS message to probe its neighborhood for nearby DODAGs. Most of the simulators utilize only RPL for prefix dissemination, IPv6 ND for neighbor discovery, and IPv6 DAD for duplicate address detection, which is not correct. It must be stressed that, even if RPL DIO messages can carry a PIO option, and thus can be used to perform SLAAC, there is no support in RPL for ND or DAD. Hence, without 6LoWPAN-ND, the "normal" IPv6 procedures must be used.

Another factor where we can say that 6LoWPAN-ND is mandatory is because it allows us to push 6CO inside the network and 6LoWPAN-IPHC uses contexts to compress the IPv6 header having global addresses. Of course it is possible to configure contexts manually, but this prevents optimizations based on the network operations.

The RPL DIO message carries the PIO. It would be a conflict if a network is configured with RPL and 6LoWPAN-ND protocols. Since RPL+DIO and 6LoWPAN-ND+RA is redundant functions, it may lead to a miss-configuration of the entire network, especially if both PIO's are different. This point is still open, and in the future it would be useful to clarify the respective roles and use-cases. However, this will require a cooperation between both IETF groups *ROLL* and *6lo*.

Another point is what if there is a DHCPv6 [106] for the given 6LoWPAN? Well, the DHCPv6 is "unnecessary", as it would duplicate 6LoWPAN-

ND functionalities. e.g., the addresses registration mechanism is also performed by the 6LoWPAN-ND and address assignment, it couldn't do it "better" than SLAAC + 6LoWPAN-ND because compression works better with SLAAC. So there is no real need of DHCPv6.

Moreover, IETF standards [130] and [143] define mesh-under topology where all nodes are one IP hop away from 6LBR, in this scenario the significance of 6LoWPAN-ND protocol become reduced, because of Layer-2 forwarding. Interesting enough, I will show that this is exactly where current implementations falls short. Even if all the nodes are apparently 1-hop away from each other, this does not means that they are, and this misunderstanding leads to dire consequences.

4.3 6LoWPAN-ND Implementation Status

Keeping in mind the importance and complexity of the 6LoWPAN protocol, there are several simulators & operating systems that are still in the phase of implementation. Most of them has implemented the 6LoWPAN Compression [64,105] (We did not take into account the general header compression standard [27]) but few of them partially implemented 6LoWPAN protocol [130,143].
As shown in the Table 4.1 mostly open-source work has done in the implementation of the 6LoWPAN protocol. Moreover, it also gives the detail about how many simulators and operating systems have implemented 6LoWPAN Compression and 6LoWPAN-ND protocols.

Simulators from [5] to [9] has not implemented the 6LoWPAN-ND protocol except. It also implemented the 6LoWPAN-IPHC. [6] does not support RPL but one research work has implemented, where model itself integrates Contiki's [1] by using its resources (e.g., 6LoWPAN with HC1). IPv6/UDP header compression mechanism is abstract in nature and uses the Thread protocol [7] for routing by [9]

From [1] to [2] are the types of operating systems where mostly support 6LoWPAN stack except [2]. [1] has the lightweight implementation of RPL called RPL-lite but it does not support 6LoWPAN-ND. OpenWSN is a research project among multiple universities where they provide a firmware that supports IEEE802.15.4e and RPL in only non-storing mode. Their stack support 6LoWPAN but not 6LoWPAN-ND. The RiOT-OS [10] stack support (6LoWPAN-IPHC) and partially support 6LoWPAN-ND (multihop

Table 4.1: Supported features in simulators & operating systems.

Name	Type	License	6C	ND
[5] ns3	sim	Open Source	✓✓	–[1]
[6] OMNeT++	sim	Open Source	–[2]	–
[4] NetSim	sim	Proprietary	–	–
[9] QualNet	sim	Commercial	✓	–
[1] Contiki-NG	os	Open Source	✓✓	–[3]
[8] OpenWSN	os	Open Source	✓✓	–
[10] RiOT OS	os	Open Source	✓✓	✓
[11] Tiny OS	os	Open Source	✓	–
[3] Mbed-OS	os	Open Source	✓✓	✓
[12] Zephyr OS	os	Open Source	✓✓	–
[2] FreeRTOS	os	Open Source	–	–

6C = *6LoWPAN Compression* : – None, ✓ RFC4944, ✓✓ RFC6282
ND = *6LoWPAN-ND* : – None, ✓ RFC6775, ✓✓ RFC8505

[1] Independent work by us but soon it will be part of [5] official release.
[2] Not part of the official release. Independent work [78].
[3] Not part of the official release. Independent work [128].

DAD ([130], section 8.2) is still missing.). [11] is obsoleted, and rarely used currently, because It had implemented the [105] with some early drafts of [64] but not fully implemented.

The Mbed-OS [3] support three IP stacks, *LwIP stack, Nanostack, and External IP Module*, First and last are out of our discussion but with Nanostack, at the network level, it supports IPv6 with 6LoWPAN adaptation layer, where the use case is only Mesh networking and at *Border Router*. It supports configuration for RPL and 6LoWPAN-ND mesh networking.

The Zephyr OS [12], supports RPL as well as it also integrates an open-source Thread protocol implementation called OpenThread [7]. It supports 6LoWPAN-IPHC but does not support 6LoWPAN-ND. it still relies on conventional IPv6-ND protocol.

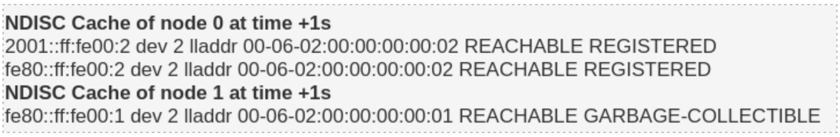
The FreeRTOS+TCP [2] is currently an IPv4 TCP/IP stack, but IPv6 functionality along with support for multiple network interfaces is available in a FreeRTOS Labs project.

4.4 6LoWPAN-ND Complexity

In this section, we will discuss the complexity of 6LoWPAN in terms of NCE states relationships between IPv6-ND and 6LoWPAN-ND, link-local and global *Address Registration* processes, and implicit and explicit NUD works.

4.4.1 NCE States

The 6LoWPAN-ND [130] standard has defined three NCE states (GARBAGE-COLLECTIBLE, REGISTERED and TENTATIVE) which are orthogonal to the states specified in the IPv6-ND [109]. These new states are not well explained and not even discussed in the successor IETF document [143]. For example how the states formed in 6LN and 6LBR? How NCE's changing their states before and after link-local and global address registration? Which node maintains what state on the reception of ND messages? Figure 4.12 depicts that how orthogonality of both standards works in 6LoWPAN's. Node 0 is a 6LBR and node 1 is a 6LN. If a node is not in the REGISTERED state then at *registering node*, IPv6-ND states will become irrelevant.



```
NDISC Cache of node 0 at time +1s
2001::ff:fe00:2 dev 2 lladdr 00-06-02:00:00:00:00:02 REACHABLE REGISTERED
fe80::ff:fe00:2 dev 2 lladdr 00-06-02:00:00:00:00:02 REACHABLE REGISTERED
NDISC Cache of node 1 at time +1s
fe80::ff:fe00:1 dev 2 lladdr 00-06-02:00:00:00:00:01 REACHABLE GARBAGE-COLLECTIBLE
```

Figure 4.12: NCE's States for 6LBR and 6LN

6LN NCE States

In the mesh-under, 6LN's function as a *registered node*, and only use the GARBAGE-COLLECTIBLE state which keeps the next-hop link-local address by maintaining the orthogonality of the IPv6-ND states. By definition, in mesh-under next-hop is one IP hop, hence the 6LBR, and network works as a star topology. There is no need to maintain states against global addresses, as per standard obligation. When a 6LN receives the RA or NA(EARO) with unsuccessful link-local address registration then it will not make any state from 6LoWPAN-ND so orthogonality of IPv6-ND states become irrelevant.

6LR NCE States

In route-over, 6LR's maintain GARBAGE-COLLECTIBLE states for their next-hop *parent* nodes just like 6LN's in mesh-under. Moreover, they also keep records of each of their child's link-local and global addresses by maintaining the REGISTERED state along with IPv6-ND states. The REGISTERED state is mandatory for both addresses. If link-local address registration is successful and global fails then the REGISTERED state against the link-local needs to remove and a node starts sending RS messages or goes on with pending RA messages processing. The TENTATIVE state is totally implementation-dependent and may only work when a 6LR is waiting for DAC for the registered node. Meanwhile, packets destined to the registered node must be dropped. Once GARBAGE-COLLECTIBLE (for parent nodes) or the REGISTERED (for child node) states formed then the IPv6-ND invoke to maintain the orthogonality over these states. As a matter of fact, the TENTATIVE state is redundant, and can be totally eliminated in an implementation without any consequence. This has been confirmed by the RFC editor.

6LBR NCE States

The 6LR acts like a replica of the 6LBR. However, 6LBR only keeps all NCE's as REGISTERED for both link-local and global addresses. The 6LBR is always a parent node in both topologies, so it will not create GARBAGE-COLLECTIBLE NCE. The IPv6-ND state only takes into account when a successful address registration done for each address.

4.4.2 Address Registration

This functionality is very complicated to manage by not only for registered nodes but also for the registering nodes. A 6LN may receive multiple RA's against a single RS transmission. Each RA includes multiple options to help nodes in the configuration setup. The recipient of multiple RA's must buffer all and processed on their turn one by one, because standard demands both link-local and global addresses registration with all one hop neighbors. In mesh-under both address registration performed by the 6LBR. In route-over, both link-local and global addresses must register to one hop parent neighbor's and for global addresses, it is mandatory to register with 6LBR.

Each 6LR help their child node for global addresses registration by using DAR and DAC messages [143].

Link-local Unicast Address (LUA) Registration

It is obligatory that all 6LN's register their link-local addresses to their respective parent nodes, from where they receive RA's. The reason for this rule is to handle the address duplication problem. A scenario related to this problem is shown in the figure 4.13. There is a chance of node B and C having the same MAC address, hence the same link-local address. Another aspect is that each node may receive multiple RA's from neighboring nodes. If a registering node fails the registration with any valid reason (i.e., EARO Status Codes) mentioned in [143] then registered node must remove all information that RA message includes from its buffer. A registered node is unable to process the next RA message until it receives the result of the previous registration.

Global Unicast Address (GUA) Registration

In mesh-under, it's not only depend on the status code in the NA(EARO), but also depend on the successful link-local address registration. Meaning that if the link-local address registration fails then no question for the global address registration. On the other hand, this registration process becomes more complex in the rout-over topology. For example 6LR has to wait for a successful DAC from the 6LBR, then it will register the registered node global address in its NCE, otherwise not.

4.4.3 Neighbor Unreachability Detection

Explicit Reachability Confirmation

In both topologies, registering node's perform NUD by sending normal NS/NA to check the registered node reachability on the reception of the NS(EARO) for link-local and global address registration. Registering node's may also do NUD's once in a while during the until the registration time out. But it also depends on the implementation. Reason for this procedure is to maintain uniqueness of addresses in the entire network. It is restricted for child nodes to do it.

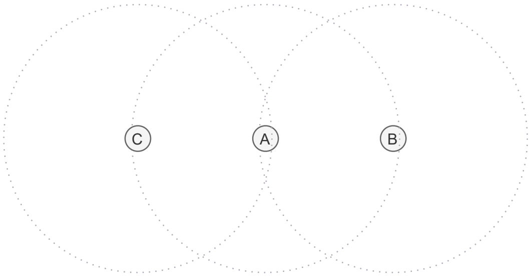


Figure 4.13: Link-local Unicast Address Registration

Implicit Reachability Confirmation

The exchange of NS(EARO) and NA(EARO) with successful registration of link-local or global addresses between child and parent node's provide implicit NUD. This is one of the reason for a child to not perform the explicit NUD.

4.5 Simulation and Results

I analyse the performance and overhead of the 6LoWPAN-ND [143] and IPv6-ND [109] in a non mesh-under and mesh-under scenarios. Where multiple 6LN's are connected to a single 6LBR to form a *star* topology. In order to evaluate the robustness and reliability of both standard protocols in a typical WSN, I deeply investigate their core functions and their the weaknesses by using ns-3 simulator.

4.5.1 Simulator Description

To evaluate both protocols in terms of performance, overhead, and their weakness, I set up the ns-3 simulators parameters as shown in the Table 4.2. I implemented the 6LoWPAN-ND module in the ns-3 simulator. The code is actually under review and will be made public in the near future.

4.5.2 Simulation Scenarios

To analyze the performance of both protocols in mesh-under scenarios. I construct very simple topologies. First, is the *grid* and second is the *cir-*

Table 4.2: Simulation parameters

Parameter Type	Value
Radio Range	About 100m
Distance between nodes	30m
802.15.4	Beacon-less, always on
Propagation Model	Log-Distance
6LoWPAN Compression	RFC 6282
6LoWPAN-ND	RFC 8505
IPv6-ND	RFC 4861
Mobility Model	Constant Position Mobility Model

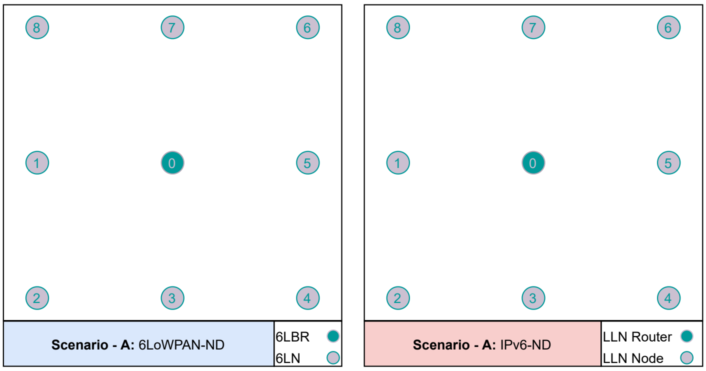


Figure 4.14: Grid Mesh-under Topology

cle topology. In both scenarios, nine nodes are participating where a single node in the center acting as 6LBR while running the 6LoWPAN-ND protocol, and in the case of IPv6-ND, it acts as LLN Router. The rest of the nodes are LLN nodes with respect to the running protocols. In the case of 6LoWPAN-ND, these nodes are called 6LNs, meaning that they will maintain the NCE according to the [143] and LLN nodes will follow the [109] to maintain the NCE. The grid and circle mesh-under topologies are shown in the Figures 4.14 and 4.15 respectively.

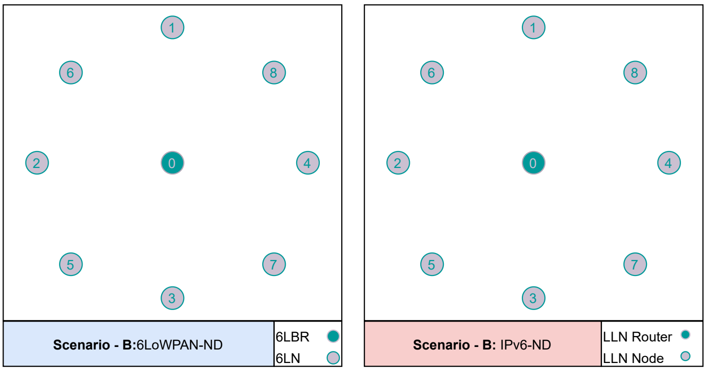


Figure 4.15: Circle Mesh-under Topology

4.5.3 Simulation Results

In the coming sections, first I explain the effects of the both protocols in both topologies while the Data Traffic is disabled (DTD). Later I discuss the behaviour of both protocols while enabling the data traffic also called Data Traffic Enabled (DTE).

4.5.4 Data Traffic Disabled (DTD) Mesh-under Grid Vs Circle Topology

The first result I obtain is about the Control Messages Count (CMC) with respect to time which is in seconds as shown in the Figure 4.16. I execute this simulation just for 40 seconds where I can see that 6LoWPAN-ND [143] is more chatty as compare to IPv6-ND [109]. Because both protocols in start send RS messages on all router multicast address but in case of 6LoWPAN-ND all nodes receive a unicast RA from the 6LBR but in case of IPv6-ND a single multicast RA update all LLN nodes in the vicinity.

Second reason of being chatty is that in 6LoWPAN-ND there is a node registration process where all 6LN nodes register their link-local and global unicast addresses (LUA and GUA) to 6LBR. While this process is not required in the IPv6-ND protocol. The registration is mandatory process as discussed in the Section 4.4.2.

In Figure 4.17, it is evident that by using the 6LoWPAN-ND protocol in mesh-under, the number of transmitted packet count (Tx) is far higher than

IPv6-ND. The standard [143] claims the reduction of multicasting for LLN's to maintain the battery for a long time, but on the other hand, it increases the unicast transmission. As we can see in the grid, the unicast count is 67 and the multicast count is 14, and in total CMC is 81 packets. Similarly in circle topology, unicast is 53 and 13 packets of multicast which leads CMC in a total count of 66 packets.

Another result that is illustrated in the Figure 4.18, where during the simulation time, 6LoWPAN-ND congest the whole network more than IPv6-ND protocol. For example, due to the nature of mesh-under and flooding, receive packet count (RxPktCount) is higher in both scenarios. Not only this, It is also depicted that at a certain maximum transmission (MAX Tx Pkt Count) and reception (MAX Rx Pkt Count) is higher as compare to IPv6-ND protocol. As a side note in the Figure 4.18, the packet count in grid topology is a bit higher than the circle, because, I changed the distance between nodes. In circle, distance between nodes is 45m and in grid remain the same (i.e,. 30m). Actually I found almost the same count in both scenarios while using the same distance between nodes, the reason behind this change is to check the variation of both protocols in dense and sparse networks. This would be count as my future work.

Moreover, the 6LoWPAN-ND is heavier in terms of Total Packet Size(TPS) as compare to the IPv6-ND protocol. In Figure 4.19, I calculate transmitted(Tx) and received(Rx) packets in terms of bytes for both protocols. The sum of Tx and Rx is derived from the total number of packets send and receive by all nodes. During the simulation run, at a certain point, maximum transmission in terms of bytes (MAX Tx TPS) is by 6LoWPAN-ND, which means nodes congest the whole network at this point. I would say here that if we increase the network scalability then this count will increase drastically, which may lead to a lot of collisions and packet drops.

Another important difference between both protocols as discussed above is about the registration and no registration of addresses (LUA and GUA). As said, IPv6-ND does not require the registration of both addresses, but 6LoWPAN-ND does. To check the behavior of 6LoWPAN-ND, the 6LBR keeps the addresses registered for 1 day, and every 3 hours each node performs the re-registration process. As illustrated in the Figures 4.20 and 4.21, all nodes perform registration on every 3 hours interval. The first registration was completed in a couple of seconds but in the figure, it shows within the first 16 minutes. On the other hand, there is just one single transmission

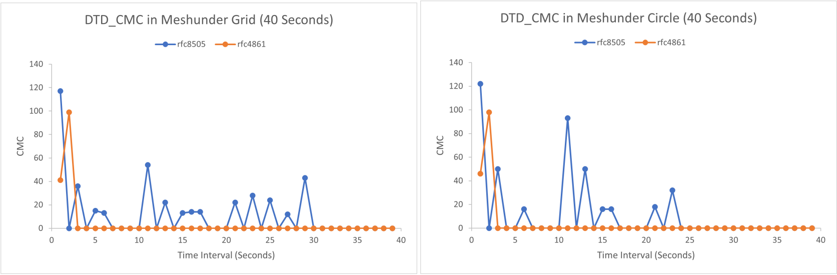


Figure 4.16: DTD-CMC in Mesh-under Grid and Circle

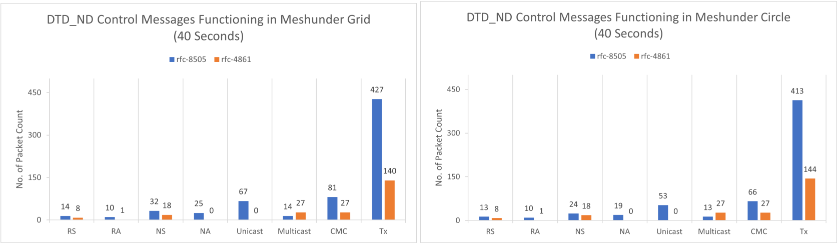


Figure 4.17: DTD-ND Control Messages in Mesh-under Grid and Circle

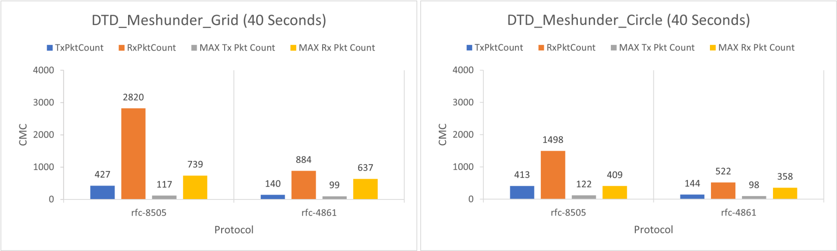


Figure 4.18: DTD-Tx,Rx, and MAX Count in Mesh-under Grid and Circle

of RA by using all node multicast address, which updates all the LLN nodes NCEs. Keep in mind that here all nodes start NS/NA transmission for the address resolution if the the NCE is not in REACHABLE state and packet arrives from the upper layer. But in 6LoWPAN-ND case, this address resolution process is not required and forcefully stopped by the standard [143]. In this case node will transmit packet to its one hop parent node instead of starting NS/NA(Address Resolution). Network administrator can increase

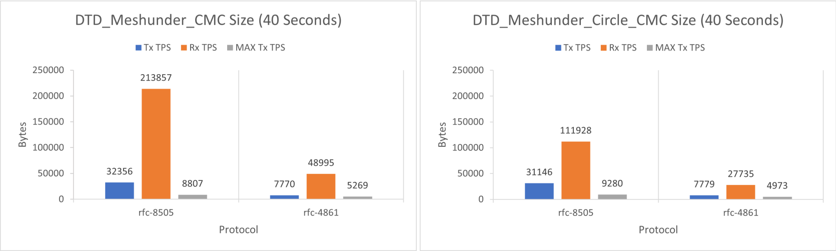


Figure 4.19: DTD-TPS in Mesh-under Grid and Circle

the address re-registration time to reduce the number of transmissions, but this all depends on the network requirements.

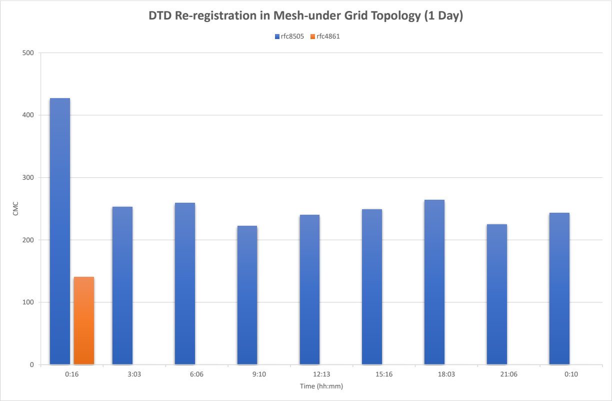


Figure 4.20: DTD Re-registration in Mesh-under Grid Topology

4.5.5 Data Traffic Enable (DTE) Mesh-under Grid Vs Circle Topology

Having the same scenarios, I investigated both protocols while enabling the data traffic. The parameters such as maximum packet size, inter packet interval, maximum packet count, and simulation stop time are 12-bytes, 1 second, 200 packets, and 300 seconds, respectively. With these parameters,

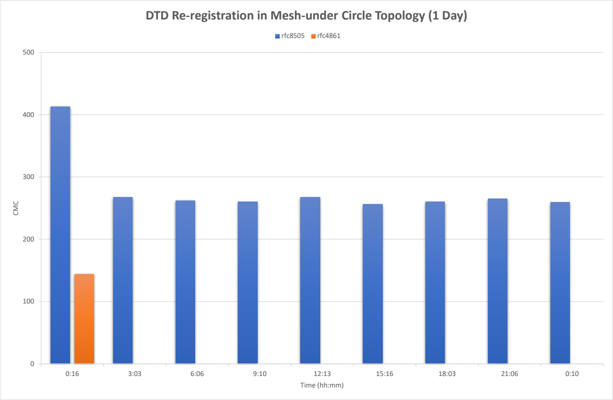


Figure 4.21: DTD Re-registration in Mesh-under Circle Topology

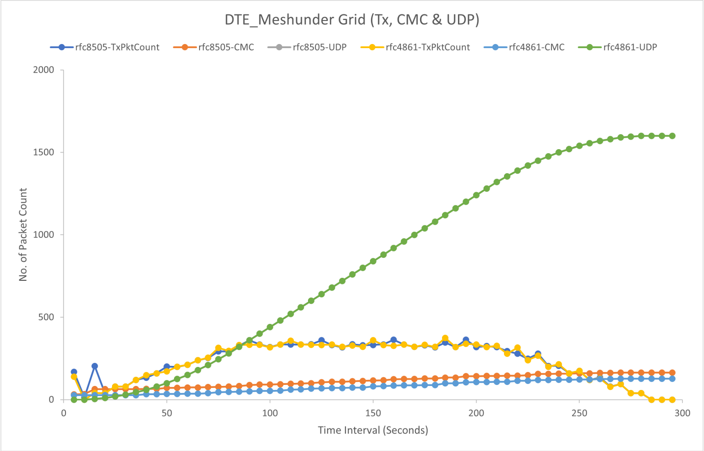


Figure 4.22: DTE-Mesh-under Grid (Tx,CMC and UDP)–300

I executed another simulation where all 6LNs and LLN nodes start sending UDP packets to 6LBR and LLN Router respectively.

In mesh-under grid (Figure 4.22) and mesh-under circle (Figure 4.23) are presenting the behaviour of the transmitted (Tx), CMC, and UDP packets. The UDP traffic does not make any effects but over all control messages

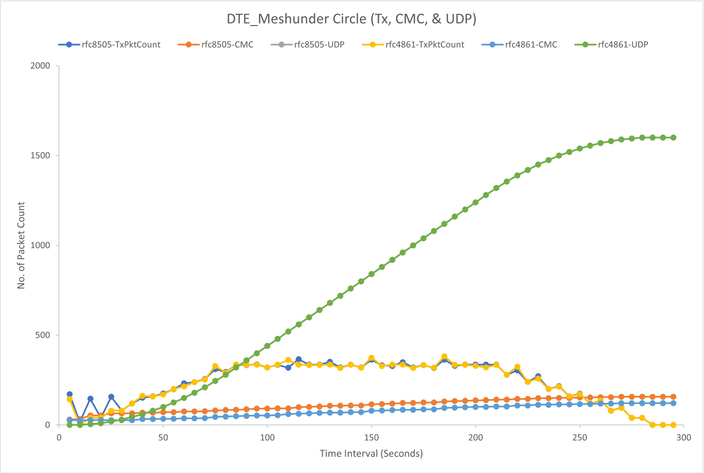


Figure 4.23: DTE-Mesh-under Circle (Tx,CMC and UDP)-300

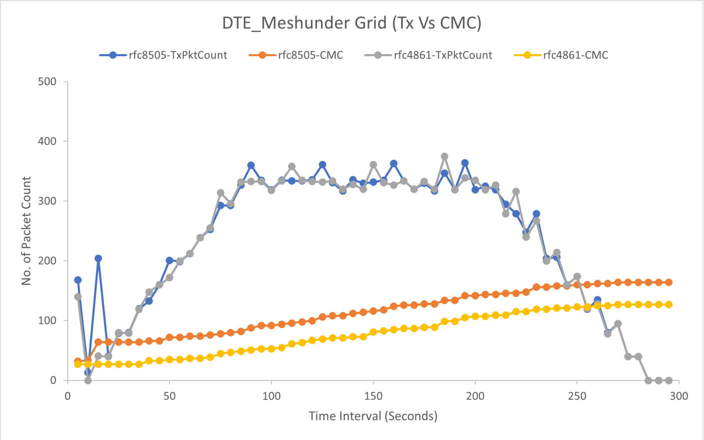


Figure 4.24: DTE-Mesh-under Grid-300

(CMC) in first 10 to 20 seconds are higher in 6LoWPAN-ND as discussed above (depicted in Figure 4.16), but later on they remain the same as IPv6-ND CMC. Due to mesh-under with flooding nature and proactive approach of 6LoWPAN-ND protocol, a burst of Tx also cause a lot of collisions in first 10 to 20 seconds but only in case of 6LoWPAN-ND protocol but as we can see that after 30 seconds both protocols acting like a same. Because

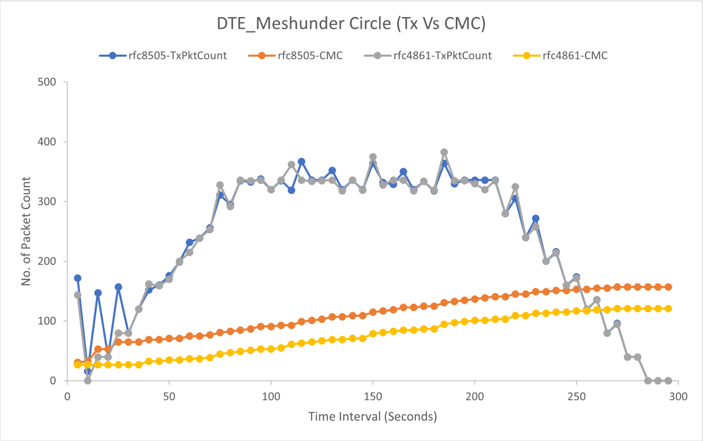


Figure 4.25: DTE-Mesh-under Circle-300

of the neighbor discovery REACHABLE state has expired as mentioned in IPv6-ND protocol [109]. The more clarity of both CMC and Tx counts can be seen in Figures 4.24 and 4.25.

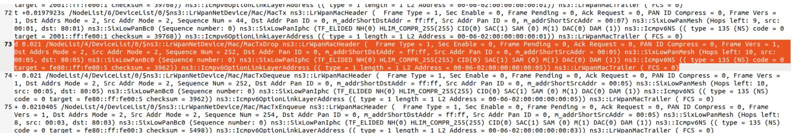

```
target = 2001::ff:fe00:1 checksum = 39768)) ns3::Icmpv6OptionLinkLayerAddress (( type = 1 length = 1 L2 Address = 00-06-02:00:00:00:00:01)) ns3::LrWpanMacTrailer ( FCS = 0)
t +0.0197923s /NodeList/6/DeviceList/0/$ns3::LrWpanNetDevice/Mac/MacTx ns3::LrWpanMacHeader (  Frame Type = 1, Sec Enable = 0, Frame Pending = 0, Ack Request = 0, PAN ID Compress = 0, Frame Vers =
1, Dst Addrs Mode = 2, Src Addr Mode = 2, Sequence Num = 44, Dst Addr Pan ID = 0, m_addrShortDstAddr = ff:ff, Src Addr Pan ID = 0, m_addrShortSrcAddr = 00:07) ns3::SixLowPanMesh (Hops left: 9, src:
00:01, dst: 80:01) ns3::SixLowPanBc0 (Sequence number: 0) ns3::SixLowPanIphc (TF_ELIDED NH(0) HLIM_COMPR_255(255) CID(0) SAC(1) SAM (0) M(1) DAC(0) DAM (1)) ns3::Icmpv6NS (( type = 135 (NS) code = 0
target = 2001::ff:fe00:1 checksum = 39768)) ns3::Icmpv6OptionLinkLayerAddress (( type = 1 length = 1 L2 Address = 00-06-02:00:00:00:00:01)) ns3::LrWpanMacTrailer ( FCS = 0)
d 0.021 /NodeList/4/DeviceList/0/$ns3::LrWpanNetDevice/Mac/MacTxDrop ns3::LrWpanMacHeader (  Frame Type = 1, Sec Enable = 0, Frame Pending = 0, Ack Request = 0, PAN ID Compress = 0, Frame Vers = 1,
Dst Addrs Mode = 2, Src Addr Mode = 2, Sequence Num = 252, Dst Addr Pan ID = 0, m_addrShortDstAddr = ff:ff, Src Addr Pan ID = 0, m_addrShortSrcAddr = 00:05) ns3::SixLowPanMesh (Hops left: 10, src:
00:05, dst: 80:05) ns3::SixLowPanBc0 (Sequence number: 0) ns3::SixLowPanIphc (TF_ELIDED NH(0) HLIM_COMPR_255(255) CID(0) SAC(1) SAM (0) M(1) DAC(0) DAM (1)) ns3::Icmpv6NS (( type = 135 (NS) code = 0
target = fe80::ff:fe00:5 checksum = 3962)) ns3::Icmpv6OptionLinkLayerAddress (( type = 1 length = 1 L2 Address = 00-06-02:00:00:00:00:05)) ns3::LrWpanMacTrailer ( FCS = 0)
- 0.021 /NodeList/4/DeviceList/0/$ns3::LrWpanNetDevice/Mac/MacTxDequeue ns3::LrWpanMacHeader (  Frame Type = 1, Sec Enable = 0, Frame Pending = 0, Ack Request = 0, PAN ID Compress = 0, Frame Vers =
1, Dst Addrs Mode = 2, Src Addr Mode = 2, Sequence Num = 252, Dst Addr Pan ID = 0, m_addrShortDstAddr = ff:ff, Src Addr Pan ID = 0, m_addrShortSrcAddr = 00:05) ns3::SixLowPanMesh (Hops left: 10,
src: 00:05, dst: 80:05) ns3::SixLowPanBc0 (Sequence number: 0) ns3::SixLowPanIphc (TF_ELIDED NH(0) HLIM_COMPR_255(255) CID(0) SAC(1) SAM (0) M(1) DAC(0) DAM (1)) ns3::Icmpv6NS (( type = 135 (NS)
code = 0 target = fe80::ff:fe00:5 checksum = 3962)) ns3::Icmpv6OptionLinkLayerAddress (( type = 1 length = 1 L2 Address = 00-06-02:00:00:00:00:05)) ns3::LrWpanMacTrailer ( FCS = 0)
+ 0.0210405 /NodeList/4/DeviceList/0/$ns3::LrWpanNetDevice/Mac/MacTxEnqueue ns3::LrWpanMacHeader (  Frame Type = 1, Sec Enable = 0, Frame Pending = 0, Ack Request = 0, PAN ID Compress = 0, Frame
Vers = 1, Dst Addrs Mode = 2, Src Addr Mode = 2, Sequence Num = 254, Dst Addr Pan ID = 0, m_addrShortDstAddr = ff:ff, Src Addr Pan ID = 0, m_addrShortSrcAddr = 00:05) ns3::SixLowPanMesh (Hops left:
8, src: 00:03, dst: 80:03) ns3::SixLowPanBc0 (Sequence number: 0) ns3::SixLowPanIphc (TF_ELIDED NH(0) HLIM_COMPR_255(255) CID(0) SAC(1) SAM (0) M(1) DAC(0) DAM (1)) ns3::Icmpv6NS (( type = 135 (NS)
code = 0 target = fe80::ff:fe00:3 checksum = 5498)) ns3::Icmpv6OptionLinkLayerAddress (( type = 1 length = 1 L2 Address = 00-06-02:00:00:00:00:03)) ns3::LrWpanMacTrailer ( FCS = 0)
```

Figure 4.26: NS(DAD) dropped in Mesh-under Grid Topology [rfc4861]

Another interesting fact found while running IPv6-ND over grid scenario, that is the drop case of NS(DAD) packet. This packet always sent by all participating LLN nodes into network having the destination address as Solicited Node Multicast Address (SNMA). As shown in Figure 4.26, packet dropped at node 4. This NS(DAD) must requires a response if and only if the receiving node has the same IP address. In my experiment all nodes have unique addresses (MAC and IP), but if we consider the practical environment where DHCP is not running and multiple LLN nodes rely on NS(DAD), then for sure IPv6-ND will fail and unable to provide the reliability to the system.

LUA is used for link-local communication and for multi-hop communication GUA is used. I developed a very simple topology to analyze both types of addresses with 6LoWPAN-ND and IPv6-ND. As shown in Figure 4.27,

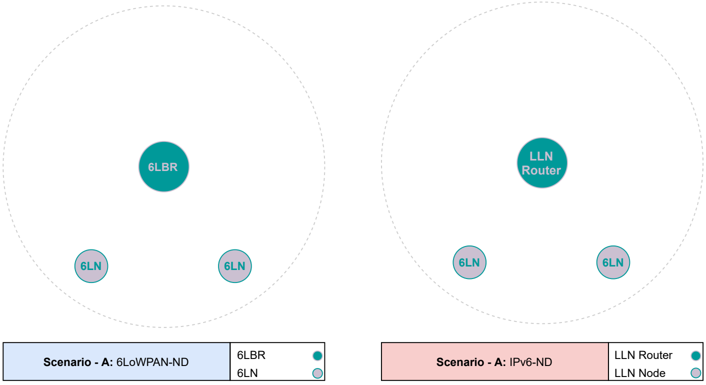


Figure 4.27: Data Communication using LUA Vs GUA in Mesh-under and Non Mesh-under

where two 6LNs are in the vicinity of 6LBR and they are also in the radio range of each other. The same scenario was deployed to analyze the IPv6-ND protocol, where LLN Router is managing the network, instead of 6LBR. While running IPv6-ND protocol on this very simple topology, where both 6LNs want to communicate with each other by using LUAs and GUAs. In both cases, they are able to do that. On the other hand, when I analyze both addresses types in the 6LoWPAN-ND scenario, the GUA communication works but through the 6LBR. But when they use LUAs for data communication, then 6LoWPAN-ND fails.

This point (at first sight counter intuitive) is extremely important, and it highlights some peculiarities of 6LoWPAN-ND. It is first necessary to remember that we did not use in our simulation any routing protocol, i.e., RPL is not present. Moreover, 6LoWPAN-ND mandates that two nodes can communicate directly if and only if they did perform a neighbor registration. This is necessary because, in general, there is no guarantee at all that the MAC address is unique in the LLN. Hence there is no guarantee that node A can communicate its neighbor B, because neighbor B could have a conflict with a third node C having the same MAC address as node A. Note that this consideration appeared in RFC8505, and is not present in RFC6775, mainly because RFC6775 is specifically targeted to IEEE 805.15.4, while RFC8505 applies to a generic LLN.

Assuming that there is no routing protocol, nodes A and B, especially in a mesh-under topology, can not know if they are at 1 physical hop or not. Hence, they will not start an address registration, thus preventing a direct communication. If RPL (for example) would have been used, then it would have been possible to recognize the physical 1-hop condition though the reception of DIOs, hence enabling an address registration, and the direct communication.

Summarizing, 6LoWPAN-ND does not provide the solution for this problem and forced the nodes to always register to their next hop router nodes (6LR/6LBR) and always use their next-hop router for data communication but not with LUAs. This is totally normal because LUAs are valid on the *physical* link, and even if the mesh-under will hide the physical topology from IPv6, the address validity is still bound to it. Hence, we can conclude that this apparently counter intuitive behaviour is, indeed, a consequence of the correct application of the standard. The summary of this experiment also illustrated in Table 4.3 and Table 4.4.

The interesting part is that by not using 6LoWPAN-ND, the nodes would communicate even through LUAs, and this is not only a violation of the standard, but also something that can work only in extremely specific scenarios, i.e., where it is possible to guarantee that there is no MAC address duplication in the LLN.

Table 4.3: Data Communication with Link-local Unicast Address (LUA)

Protocol	**Topology**	
	Mesh-under	**Non-Mesh-under**
rfc8505	Not Working	Not Working
rfc4861	Working	Working

4.6 NDP and DAD reliability

Beside the topics mentioned so far, in my simulations I found another point that must be mentioned, and (interestingly) is not stressed enough in either RFC6775 or RFC8505.

Both IPv6 DAD and ND are using multicast packets, and the standards rightfully identify this as a problem because the routing protocol could be

Table 4.4: Data Communication with Global Unicast Address (GUA)

Protocol	**Topology**	
	Mesh-under	**Non-Mesh-under**
rfc8505	Working	Working
rfc4861	Working	Working

not supporting multicast, and even if it is (like in the case of mesh-under), this leads to energy inefficiency.

However, there is another point to mention. Both protocols rely on the assumption that the probability that a message is not received by the destination is negligible, e.g., the target node will receive an DAD and will reply if there is an address conflict. Moreover, DAD use an *implicit* signalling: if there is no answer, then a positive outcome is assumed (there is no address conflict).

If the network has a high loss probability on the link, and this is a common assumption in LLN (after all, LLN stands for Low-power and *Lossy* Network), the probability to not receive a DAD sent as multicast are high. Hence, the whole DAD mechanism is to be considered unreliable.

This point alone should make it clear that 6LoWPAN-ND is not optional for LLNs, it is the only reliable solution.

4.7 Conclusion

The traditional classical IPv6-ND protocol was designed for serial links and shared transit media for example Ethernet. It works fine also for wireless non-LLNs, and it is based on a reactive approach (build the neighbor cache when it is needed). On the contrary, 6LoWPAN-ND uses a proactive approach with high transmission in the bootstrap phase.

The claimed purpose of developing 6LoWPAN-ND is to reduce the heavy multicasting that LLNs can not afford. The purpose of this research was to investigate the performance and overhead of 6LoWPAN-ND and IPv6-ND protocols. It also highlighted the loopholes in both protocols, such as NS(DAD) failure in IPv6-ND and LUA communication while using 6LoWPAN-ND. Moreover, the Table 4.5 conclude the effects of on-link flag in both

Table 4.5: Effects of on-link flag in Mesh-under and Non Mesh-under

Network Type	**Onlink Flag**	**Protocol**	**Elucidation**
Non Mesh-under	*SET*	*rfc8505*	Impossible to work
		rfc4861	Not working, hidden node problem
	NOT SET	*rfc8505*	Works and solves hidden node problem
		rfc4861	Lacks NDP and DAD delegations
Mesh-under	*SET*	*rfc8505*	Impossible to work
		rfc4861	Apparently functional, but DAD is unreliable.
	NOT SET	*rfc8505*	More chatty but DAD is reliable.
		rfc4861	Less traffic but DAD is unreliable.

protocols while using mesh-under and non mesh-under.

Overall, I can state that, despite the complexity of the protocol, the 6LoWPAN-ND usefulness has been so far heavily underestimated, and that the lack of it in the implementations is a major issue, as the IPv6 DAD and ND are not suitable for LLNs. Not only they do use multicast, which could be unsupported at routing level, but they also are unreliable on LLNs. Hence, it is mandatory to foster the 6LoWPAN-ND adoption, and I think that my work on the ns-3 simulator can be a step ahead toward a deeper understanding of the protocol usefulness by the IoT operating systems developers. Toward this end, I am working on publishing my results, present them to the relevant standardization bodies, and possibly write an informative RFC.

Chapter 5

Federated learning for IoE environments: A service provider revenue maximization framework

The paper that I present in this chapter has been accepted as part of the ITU Journal on Future and Evolving Technologies (ITU J-FET) special issue on Internet of Everything - Volume 2 (2021), Issue 5. It can be found on the following link: `https://www.itu.int/pub/S-JNL-VOL2.ISSUE5-2021-A01`. To protect data security and maximize service provider income, the chapter examines the use-case of federated learning to virtual functions demand prediction in IoE based edge-cloud computing systems. In addition, the chapter offers a virtual function placement based on the federated learning module's services demand forecast. A work allocation method based on matching is proposed. At the end of this chapter numerical findings verify the suggested technique while comparing it to a chaos theory prediction system.

5.1 Introduction

The emergence of new network paradigms such as Edge Computing (EC) [91, 96, 129, 144], for which the limitations typical of the cloud architecture have been bypassed moving computation nodes to the network edges close to the end users, has given rise to a wide range of challenges in many research areas [40, 134]. Consequently, several new issues, such as user mobility, heterogeneity in Quality of Service (QoS) or service requirements, massive volume of data, user privacy, diversity on data types and so on, have led to numerous efforts from both academia and industry in providing highly effective and efficient solutions [35, 36, 47, 48, 152]. In particular, there exists a significant branch of literature regarding possible solutions to improve EC Network (ECN) performance in order to guarantee a high level of user satisfaction and to provide dynamic and flexible network resource allocation and decision-making strategies. Within this context, the Internet of Everything (IoE) paradigm, in which people, process, data, and things are connected and exchange data,has given rise to systems with increasing complexity and applications involving strict real-time requirements and sensitive data [121], heterogeneous traffic. Generally speaking, heterogeneity in data flow types implies different QoS or service requirements. Furthermore, from a Service Provider (SP) perspective, such diversity triggers new data flow management policies, service provision costs and selling prices. In this respect, the SP revenue maximization is strictly related to the adopted management and administration policy.

Indeed, a proper resource exploitation planning is essential to guarantee elevated levels of network efficiency, user satisfaction and consequent high SP revenues, as highlighted by literature such as [148], [39]. In particular, having an a priori knowledge about the data flow service demand can be properly exploited to perform suitable resource infrastructure planning with maximum income. In order to pursue this objective, Machine Learning (ML) [18, 20, 79, 103, 163] has emerged by providing many techniques to perform data behavior interpretation and analysis. The ability of ML techniques in catching data trends, patterns and hidden features, has ensured its applicability to many problems. However, although the knowledge and extrapolation of user data characteristics positively impacts many application areas, it may result in being non-compliant with some specific user privacy constraints [98]. In this respect, if on the one hand the users' data analysis may lead to remarkable advantages in reference to the network resources planning and exploitation, on the other the user data gathering may trigger user dissention, due to privacy concerns and violation. Within this context, a data-manipulation framework able to collect users' data without contravening users' privacy is a priority. In this respect, Federated Learning (FL) [98, 135, 146, 152, 156, 160] has recently emerged as a promising tool to perform, locally on the users' devices, statistical and mathematical training models based on ML methodologies without losing users privacy constraints. The FL framework consists of the devices level, generally indicated in literature as clients, and a central server unit which aggregates and merges the data preliminary processed by the clients. Typically, FL has the following matters to face with [99]

- **Non-Independent Identically Distributed Data** The clients have different training datasets, therefore a single dataset cannot be considered representative of the other clients datasets;
- **Unbalanced Datasets** Different clients have different datasets, and each dataset may have a diverse number of elements in comparison to other clients datasets;
- **Large-Scale Distribution** The number of clients involved in the FL training procedure is generally higher than the amount of data processed at the client level;
- **Limited Communication** Mobile devices may or may not be available for data training and the computational capability or communication conditions could be poor.

The rest of chapter is organized as follows. In Section 5.5 an in-depth review of the related literature is presented. Section 5.6, discusses the problem statement, while in Section 5.7 the FL framework and the placement strategy are presented. Then, in Section 5.8 the experimental results are analyzed and the alternative CT predictive approach explained. Finally, the conclusions are presented in Section 5.9.

5.2 Motivation

In accordance with the IoE paradigm, millions of people and billions of devices are expected to be connected to each other, giving rise to an ever increasing demand for application services with a strict quality of service requirements. Therefore, service providers are dealing with the functional integration of the classical cloud computing architecture with edge computing networks. However, the intrinsic limited capacity of the edge computing nodes implies the need for proper virtual functions allocations to improve user satisfaction and service fulfillment. In this sense, demand prediction is crucial in services management and exploitation. The main challenge here consists of the high variability of application requests that result in inaccurate forecasts. Federated learning has recently emerged as a solution to train mathematical learning models on the users' site. This chapter investigates the application of federated learning to virtual functions demand prediction in IoE based edge-cloud computing systems, to preserve the data security and maximise service provider revenue. Additionally, the my research work proposes a virtual function placement based on the services demand prediction provided by the federated learning module. A matching-based tasks allocation is proposed. Finally, numerical results validate the proposed approach, compared with a chaos theory prediction scheme.

5.3 Contribution

In reference to the proposed contextualization, we have assumed here that sensitive user data may be derived from historical users functions utilization. In this perspective, sharing data about daily users habits may expose the users to undue risks. For this reason, the FL framework may represent a useful tool to counteract such a problem. However, a deep investigation of the privacy issues are out of the scope of this chapter. The chapter proposes the application of the FL framework, in order to forecast the service demands, without losing the user privacy constraints, in an IoE scenario. Moreover, on the basis of service demand forecasting, this chapter proposes a suitable Virtual Functions (VFs) placement both on the ECN and cloud. Summarizing, the contributions of this chapter are

- Application of the FL strategy to forecast the network VFs demand, in order to take into account the users privacy;
- Formulation of the SP maximum revenue problem, by considering Service Requests (SRs) with a different priority and hence, different cost and price. In particular, the SP can accept the data SRs with low priority if all the high priority flows have been satisfied;
- Proposal of a VFs placement strategy and a suitable matching-based SRs allocation algorithm based on the considered FL and the previously provided VFs forecasting scheme;

- Performance evaluation of the proposed approach and the comparisons with a centralized Chaos Theory (CT)-based prediction scheme, by resorting to extensive computer simulation runs.

5.4 Machine Learning Framework

Algorithms learn from data in the same way humans learn through their experiences. Machine learning is a new technology having a primary goal of learning parameter models that are based on the training data. The key thing here is to figure out how to re-create the learning process in the brain. Although there is no final standard for solving this type of problem, numerous recurring patterns have been identified among the suggested solutions over time. The whole learning process follows a consistent pattern, which includes data modification and feature extraction at the start. The learning algorithms utilize the characteristics to select a learning model and then look for its parameters. [131] divides Machine learning into three categories, as follows:

1. *Supervised learning*: The aim here is to assign a label to the data according to a model trained on a labeled dataset provided by the supervisor, who acts as a teacher. The labeled dataset is represented by a set of input and output parameters.

2. *Unsupervised learning*: This methodology relies on recognizing patterns and structures in available data rather than using labels or responses. Clustering and dimensionality reduction are basically two examples of this type of learning paradigm.

3. *Reinforcement learning*: This strategy falls somewhere in the middle of the previous two categories. The environment takes on the role of the instructor, providing hints to the learning system, which gets feedback depending on its responses.

As a result, the objective of the ML is to define parameter models based on certain training data. Despite the fact that classic machine learning models achieve exceptional efficacy, the learning methods are highly time consuming, and the models require the training data to be centralized on a single site, such as a datacenter. FL has recently emerged as a way to address machine learning's severe limitations. The new FL approach brings

ML down to the device level. According to this method, mobile phones work together to develop a common model based on data that has been taught on the device. By separating the learning process from data storage, federated learning enhances user privacy. Furthermore, instead of being calculated on centralized machines, machine learning models are computed on mobile devices. Because today's high-performance mobile phones are strong enough to execute machine learning models, this innovative computation paradigm is feasible.

A ML model may be identified as a loss function based on a data sample z and a parameter vector $\mathbf{w}$, i.e., $f_z(\mathbf{w})$, which represents the error imposed by the model based on the training data [155]. Let m be the number of end-devices (EDs), where each ED i having its own set of local data Ω_i, where i $= 1, ..., m$. [101, 155] is a definition of the collective loss function.

$$F_i(\mathbf{w}) = \frac{1}{|\Omega_i|} \sum_{z \in \Omega_i} f_z(\mathbf{w}), \tag{5.1}$$

where $|\Omega_i|$ denotes the number of items that belong to the group Ω_i. Then, from equation 5.1 follows that the global loss function across all the $|\Omega_i|$, $i = 1, ..., m$, is thus given by [101, 155].

$$F(\mathbf{w}) = \frac{\sum_{i \in \{1,\ldots,m\}} |\Omega_i| F_i(\mathbf{w})}{\sum_{i \in \{1,\ldots,m\}} |\Omega_i|}, \tag{5.2}$$

The immediate consequence of 5.1 and 5.2, as well as [155], is finding $\mathbf{w}^\star$ such that

$$\mathbf{w}^\star = arg\ min\ F(\mathbf{w}), \tag{5.3}$$

As a result, the FL technique consists of consecutive interactions between the client and server sides, with just a fraction of the EDs participating in the training process during each algorithm iteration round u. The overall framework may be summarized as follows:

- Analogously, each ED i engaged in the training process updates its local parameter vector $\mathbf{w_i}(u)$, which was created according to [154], on the basis of Δ_i;

$$\mathbf{w}_i(u) = \hat{\mathbf{w}}_i(u-1) - \alpha \nabla F_i(\hat{\mathbf{w}}_i(u-1)), \tag{5.4}$$

where α is the learning rate and $\hat{\mathbf{w}}_i(u-1)$ is the term $\mathbf{w}_i(u-1)$ after global aggregation.

- The weighted average is computed on the server side, as described in [100] and defined by

$$\mathbf{w}(u) = \frac{\sum\limits_{i\in\{1,\ldots,m\}} |\Omega_i|\mathbf{w}_i}{\sum\limits_{i\in\{1,\ldots,m\}} |\Omega_i|}. \tag{5.5}$$

In terms of client privacy, distributed data training that follows the federated learning rules has several advantages. In reality, the training procedure on the client's site allows users to secure their sensitive and private information when uploading the parameter vector $\mathbf{w}_i$ which does not expose the client to any privacy concerns, since given $\mathbf{w}_i$, to retrieve Ω_i is not easy.

5.5 Related Works

Recently, ML techniques have found extensive applications in big data analysis in fog/edge networks research area.

An overview of the ML techniques applied to fog is presented in paper [138]. Then, paper [138] investigates the ability of the ML strategies in detecting malicious attackers in fog networks, while paper [33] focuses on the ML solutions to evaluate the advantages deriving from an edge caching solution, taking into account user satisfaction perspective and energy efficiency. The improvement in sensing reliability and network latency is the aim of paper [164], in which the authors implement a multi-hidden multilayer convolutional neural network solution to provide data authentication in a mobile crowd-sensing environment. The tree decisions strategy combined with the k-nearest neighbors method is applied in [123], in which authors deal with the position-based confidentiality problem in high real-time industrial application scenarios.

In a different way, SP maximization is the objective of paper [161], in which a deep supervised learning approach is applied to perform the minimization of the total network cost. A fog blockchain network is analyzed in paper [95], which formulates a solution based on the auction theory, where deep learning is applied to the maximization of the edge computing SP revenue.

Additionally, distributed ML is adopted in papers [82, 87, 149, 150]. In paper [149], a distributed version of the well-known support vector machine method is implemented to investigate its applicability. The reinforcement learning, and more in depth the Q-learning algorithm, is applied in paper [82], in order to minimize the users' outage in heterogeneous cellular networks scenarios. The control in crowd-sensing problem is the main objective of paper [87], exploiting the human in the loop methodology to propose a hierarchical crowd sensing framework with the aim of reducing cloud congestion and promoting the balancing of the data traffic. Then, the distributed stochastic variance reduced gradient is applied in paper [150], in which a target accuracy is fixed, and the optimization of the number of collection points to make data analysis provided. Furthermore, paper [150] proposes the minimization of the amount of network traffic sent towards the collection points. In a different way, the maximization of SP profit in a Mobile Edge Computing (MEC) blockchain network has been studied in paper [95], in which an auction strategy combined with deep learning is formulated to perform edge resource allocation. Similarly, the auction theory is also applied to the profit maximization profit in [69], in which a novel combined optimal pricing and data allocation problem is solved with the Bayesian auction approach. The profit maximization in the cognitive virtual operator is addressed in paper [89], in which a dynamic network scenario is considered. Paper [89] develops a low complexity online control scheme to perform decisions about price and resource planning. A cloud allocation scheme for three classes of virtual machines is presented in [88], with the aim of maximising cloud provider profit.

Recently, FL has gained attention and papers [77, 98, 115, 152, 159, 162] provide its application to different contexts and situations. Paper [152] and paper [162] contextualize the FL in MEC networks, optimizing with the distributed gradient descent method the trade-off between local updates and global aggregations, formulating a loss function minimization problem, and introducing some resource constraints. Papers [152] maximize the number of clients involved in the aggregation process, aiming at minimizing the aggregation error. The MEC scenario is taken into account also in paper [162] which addresses the popularity content caching problem throughout the adoption of the hybrid filtering on stacked encoders to forecast content requests trend. Authors in [159] exploit the signal superposition property of wireless channels on the basis of which a novel aggregation data strategy for the over-the-air computation is presented. Furthermore, the model proposed in [98] is applied in [98] with the stochastic gradient descent algorithm as optimizer, aiming at training data in a distributed fashion by limiting the communication costs. The multi-task learning problem is solved with the FL and the novel Mocha context-aware optimization algorithm is presented in paper [135], while a blockchained FL architecture is proposed in [77]. Then, this architecture is designed to implement a distributed consensus strategy, by taking into account the blockchain end-to-end delay. Finally, a hybrid IoT-MEC network is considered for the application of FL in [115]. Paper [115] provides transmission and computational costs optimization, applying multiple deep reinforcement learning agents. Authors in [14] propose a QoE-driven delivery approach, in which there is cooperation between the Over-The-Top and Internet service providers, aiming at maximizing the revenue. Similarly, paper [49] addresses the economic aspects of a collaborative services management between Over-The-Top and Internet service providers. Consequently, authors propose an architecture to realize their collaboration, defining three different approaches on the basis of which the profit maximization of different customers is pursued. Then, the main objective of paper [13], is the investigation of the management procedures for multimedia services, proposing a collaborative zero-rated QoE approach to model the close cooperation between mobile network operators and the Over-The-Top service providers.

As summarized in Table 5.1, in contrast to papers [69,88,89], which provide profit maximization solutions without taking into account user privacy issues, we propose a revenue maximization framework based on data information elaborated locally on the users' devices, avoiding the typical privacy concerns of the other approaches. Hence, as in papers [77,98,115,152,159,162], we propose an FL-based framework by using the gradient descent algorithm as optimizer. The motivation for this conservative choice resides in the fact that more complex methods may result in prohibitive consumption of the End Users' (EU) hardware resources, which is a crucial point in the distributed data training problems. Furthermore, in contrast to the previous up-to-date works, this chapter contextualizes the application of the FL to the VFs deployment problem, by exploiting the FL framework to properly predict the application network demand, in order to maximize the SP revenue. Furthermore, a VFs placement and an SRs service allocation is provided to evaluate the actual validity of the proposed solution. In fact, the SRs service allocation algorithm, based on the matching theory, does not take into account the SP perspective, but only the users, i.e., the SRs, interests. Finally, to the best of our knowledge, this is the first paper which applies the FL to the SP revenue maximization problem, by considering even the users' perspective. The proposed approach performance has been evaluated by resorting to extensive numerical simulation and by providing comparison with the centralized CT-based predictive method.

Table 5.1: Literature Contributions

Standard Literature	**Paper contribution**
[69, 88, 89]	Proposal of a revenue maximization framework based on data information elaborated locally on the users' devices, avoiding the typical privacy concerns of the other approaches.
[77, 98, 115, 152, 159, 162]	Contextualization of the application of the FL to the VFs deployment problem, by exploiting the FL framework to properly predict the application network demand, in order to maximize the SP revenue.

5.6 Problem Statement

As an IoE reference scenario, we consider a single SP featuring an ECN constituted by $\mathcal{N}$ Computation Nodes (CNs) located at the network edges, and a more powerful cloud located far from the ECN. We suppose that all the CNs are equipped with a Central Processing Unit (CPU) with the same computational capability and number of available Storage Resource Blocks (SRBs) S. In a different way, the cloud is assumed to have a storage capacity of U SRBs, with $S < U$. In addition, we assume the availability of high speed wired links between CNs and from any CN to the cloud[1]. Furthermore, we guess that the ECN is able to support $\mathcal{T}$ different high priority service types, which are characterized by different provision costs and selling prices. Each service type $i \in \mathcal{T}$ has associated a QoS level expressed as a time deadline

[1] We have assumed that the connection towards the cloud is performed throughout the CN nearest to the SRs needing computation. Consequently, the communication latency cost between SRs and their nearest CN has no impact on the overall SR completion time and hence it has been neglected in defining 5.12.

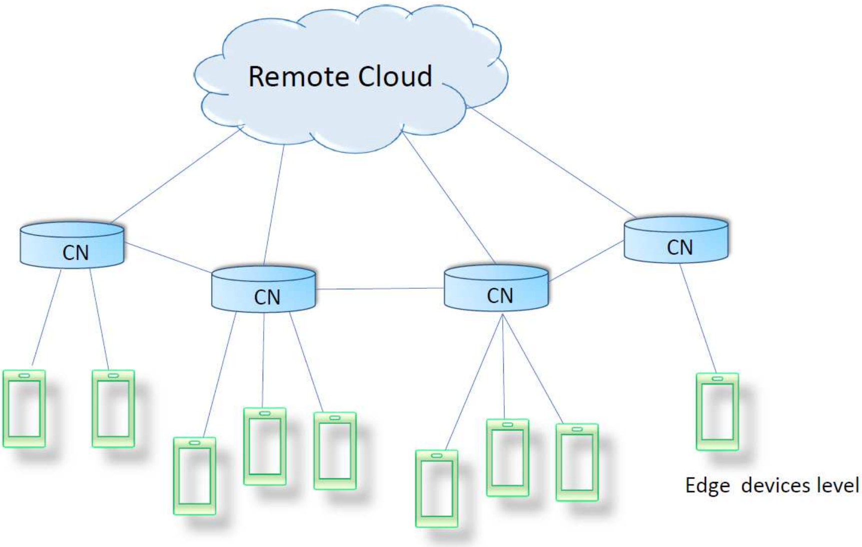


Figure 5.1: Hybrid cloud-fog network architecture

τ_i before which the type i service accomplishment has to be completed. In addition, we consider the presence of $\mathcal{M}$ service type requests with lower priority and without any time deadline constraint. The number of requests belonging to this class is indicated hereafter with y_j, with $j \in \mathcal{M}$.

Periodically, the SP updates the service demand and we assume that any new request does not arrive between two SP updates.

Let x_i be the number of SRs demanding for service i. We suppose that each SR is originated by an EU), and that an EU requires only one SR. Therefore, as a direct consequence, hereafter we assume interchangeable the SR and EU terms. Then, as regards the SP, the provision of a service has a cost mainly depending on x_i and following the model given by [153]

$$c(x_i) = \begin{cases} 0, & x_i = 0, \\ \beta_{c,i} + \beta_{l,i}\mu_i^{x_i}, & x_i > 0, \end{cases} \tag{5.6}$$

in which $\beta_{c,i}, \beta_{l,i}, \mu_i$ are real valued parameters whose value changes on the basis of the request type.

Similarly, the provision cost for providing y_j SRs of type j follows the

Table 5.2: Main symbols

Notation	Description
CN	Computation node
VF	Virtual function
FL	Federated learning
SRB	Storage resource block
SR	Service request
S	Number of SRBs per CN
U	Cloud SRBs
ECN	Edge computing network
$\mathcal{T}$	High priority requests
$\mathcal{M}$	Low priority requests
τ_i	Time deadline
x_i	Number of req. demanding for service i
y_j	Number of req. demanding for service j
$\mathcal{X}(x_i, q_i)$	SP revenue for the high priority req.
$\mathcal{Y}(y_j, z_j)$	SP revenue for the low priority req.
T_r	Service accomplishment time
$\omega_{z,h}$	Waiting time on the CN
$\omega_{z,C}$	Waiting time on the cloud

rule [153]

$$b(y_j) = \begin{cases} 0, & y_j = 0, \\ \alpha_{c,j} + \alpha_{l,j}\nu_j^{y_j}, & y_j > 0, \end{cases} \tag{5.7}$$

where $\alpha_{c,j}, \alpha_{l,j}, \nu_j$ are, also in this case, real valued parameters.

Moreover, for each service type with high priority, the SP revenue results ruled by the following relation

$$U(x_i, q_i) = \frac{\log(1 + x_i)}{q_i}, \tag{5.8}$$

with $q_i = |x_i - k_i|$, where k_i is the number of SRs for which τ_i has been respected. Then, the SP revenue for the low priority SRs is given by

$$U(y_j, z_j) = \frac{\log(1 + y_j)}{z_i}, \tag{5.9}$$

where z_j is the number of SRs among y_j accepted by the network for their service. Hence, the SP revenue, corresponding to the provision of the i-th and the j-th service type, can be expressed as

$$\mathcal{X}(x_i, q_i) = U(x_i, q_i) - c(x_i), \tag{5.10}$$

and

$$\mathcal{Y}(y_j, z_j) = U(y_j, z_j) - b(y_j), \tag{5.11}$$

respectively.

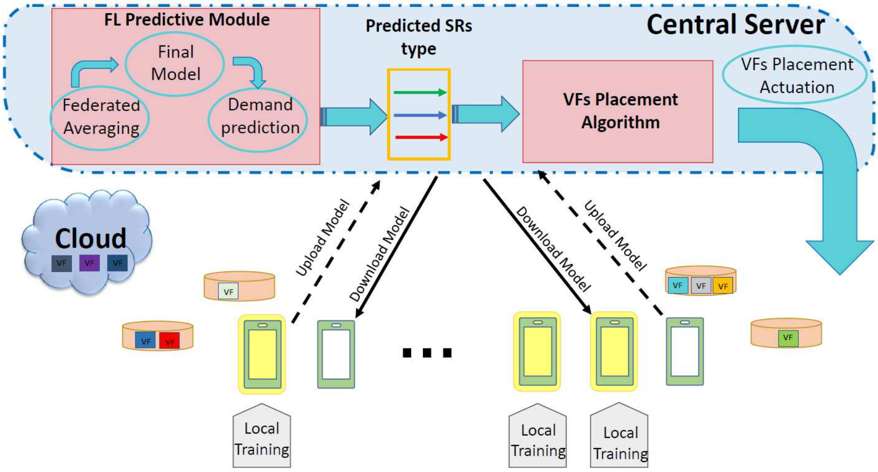


Figure 5.2: FL framework for the VFs placement

Both the SRs with high and low priority, in order to be accomplished, require the presence of a VF in set $\mathcal{V}$ which has to be preliminary loaded on at least one CN of the network or on the far cloud. The loading process requires the CN or cloud availability in terms of SRBs, since each VF $v \in \mathcal{V}$ requires a number a_v of SRBs, different for each VF. Consequently, the time required for the service accomplishment (TSA) of a generic SR r, independently by its priority, is given by

$$T_r = \sum_{v \in \mathcal{V}} \sum_{h \in \mathcal{N}} (\gamma_z + \omega_{z,h}) \rho_{r,h} \theta_{v,h} + (1 - \rho_{r,h}) \zeta_{v,C} (\gamma_C + \omega_{z,C}), \tag{5.12}$$

where γ_z and γ_C are the execution time spent by the SR z on the CPU of a CN and of the cloud, respectively. It is important to note that both the execution times γ_z and γ_C mainly depend on the size of the SR z, the CPU frequency of the node hosting its elaboration, and the time spent by the SR on that node waiting for the actual computation. Therefore, $\omega_{z,h}$ and $\omega_{z,C}$ represent the queuing time experienced by the the SR z waiting for its execution on the CN h and cloud, respectively[2]. Furthermore, $\rho_{r,h}$ is a binary value equal to 1 if the SR j is executed on the CN h, 0 otherwise. Similarly, $\theta_{v,h}$ is equal to 1 when the VF v is present on CN h, 0 otherwise. Finally, $\zeta_{v,C}$ is equal to 1 if the VF v is loaded on cloud, 0 otherwise. It is important to make evident that the TSA in (5.12) strongly depends on the queuing time experienced by the SR on the service provision site. In fact, a proper deployment of VFs on the ECN may drastically reduce the TSA time.

In formal terms, the aim of this chapter is the maximization of the SP revenue by providing decision making on the VFs placement, in order to satisfy the SRs. Therefore, the main goal of the chapter is given by

$$\min_{\mathbf{q},\mathbf{z}} \sum_{i=1,\dots,\mathcal{T}} \mathcal{X}(x_i, q_i) + \sum_{j=1,\dots,\mathcal{M}} \mathcal{Y}(y_j, z_j), \tag{5.13}$$

s.t.

$$T_i \leq \tau_i, \forall i = 1, \dots, \mathcal{T}, \tag{5.14}$$

$$\sum_{v \in \mathcal{V}} \theta_{v,h} a_v \leq S, \forall h \in \mathcal{N}, \tag{5.15}$$

$$\sum_{v \in \mathcal{V}} \zeta_{v,C} a_v \leq U. \tag{5.16}$$

In problems (5.9)-(5.15), constraint (5.14) expresses the fact that each SR with a high priority has to be served, while constraints (5.15) and (5.16) represent that the VFs allocation has to respect the storage limit of CNs and cloud, respectively.

5.7 Federated Learning Framework

In this section I describe the learning problem, federated learning framework, planning of VFs installations points and SRs allocation plannings.

[2] The CPU queue has been modeled with the first-in-first-out service policy.

5.7.1 The learning problem

The aim of ML is the exploitation of some data used for training, to learn models. In order to do that, typically, ML involves the definition of a loss function representing the error implicitly resulting from the model training [152]. The loss function depends on the data sample z and a parameter vector $\mathbf{w}$, and it is named hereafter as $f_z(\mathbf{w})$. As previously introduced, this chapter supposes the presence of L SRs, with $L = \mathcal{T} + \mathcal{M}$, deriving from an underlying level of EUs, each of which disposes of a local dataset Θ_l, $l = 1, \ldots, L$. Therefore, as assumed in [98, 152], we suppose the collective loss function equals to

$$F_l(\mathbf{w}) = \frac{1}{|\Theta_l|} \sum_{z \in \Gamma_l} f_z(\mathbf{w}), \tag{5.17}$$

where $|\Gamma_l|$ is the number of elements belonging to Γ_l, referred as the cardinality of the Γ_l set. Respectively, the global function evaluated at the central server site, the global loss function, based on the distributed local dataset Θ_l and defined as [98, 152], is expressed by the following relation

$$F(\mathbf{w}) = \frac{\sum\limits_{l=1,\ldots,L} |\Theta_l| F_l(\mathbf{w})}{\sum\limits_{l=1,\ldots,L} |\Theta_l|}. \tag{5.18}$$

Therefore, the objective here is to find $\mathbf{w}^\star$ such that [152]

$$\mathbf{w}^\star = \operatorname{argmin} F(\mathbf{w}). \tag{5.19}$$

Accordingly, with numerous contemporary papers [98,152] recently proposed in literature, the optimization of (5.19) limiting the computational complexity, is pursued by applying the gradient descent method.

5.7.2 Federated learning framework

As represented in Fig. 5.2, the proposed FL framework consists of the client level, responsible for the distributed local data training, and of a server side. The server side is typically represented by a base station or a more general central unit, set up for improving the global learning model, and to merge the locally trained EU models. The client and server sides interact with each other, throughout a series of iteration rounds u. It is important to highlight that the number of EUs involved in the training procedure are a subset of the totality of the EUs.

The FL procedure consists of the following steps

- Let $\mathcal{K}$ be the set of the EUs involved in the training process. In parallel, each EU belonging to $\mathcal{K}$, i.e. EU χ, updates its local parameter vector $\mathbf{w}_\chi(u)$, which depends on its local dataset Θ_χ, accordingly with the following rule [152]

$$\mathbf{w}_\chi(u) = \hat{\mathbf{w}}_\chi(u-1) - \xi \nabla F_\chi(\hat{\mathbf{w}}_\chi(u-1)), \tag{5.20}$$

  where ξ is the learning rate and $\hat{\mathbf{w}}_\chi(u-1)$ represents the term $\mathbf{w}_\chi(u-1)$ after global aggregation.

- As detailed in [98], the server side computes the weighted average expressed by

$$\mathbf{w}(u) = \frac{\sum_{\chi \in \mathcal{K}} |\Theta_\chi| \mathbf{w}_\chi}{\sum_{\chi \in \mathcal{K}} |\Theta_\chi|}. \tag{5.21}$$

It is important to make evident that EUs, in performing distributed data training accordingly with the FL framework, achieve numerous advantages in terms of client privacy, and limited exploitation of their computational resources. This is directly connected to the fact that training data locally on the client's site, helps users to keep their sensitive and personal information reserved, since the uploading of the EU χ parameter vector $\mathbf{w}_\chi$ does not expose the client to any sort of privacy matter. More specifically, from $\mathbf{w}_\chi$, it is not elementary to retrieve Θ_χ.

Finally, each algorithm iteration round involves just a part of the whole EUs' set, reducing the message passing between client and central server entities. Strongly connected with this aspect, the usage of the gradient descent algorithm is able to afford the learning problem without implying an excessive resource consumption, meeting the limited computational capabilities intrinsic of each mobile device.

5.7.3 VFs placement planning

Once the FL framework is applied to obtain SRs prediction on the basis of the historical EUs' information, properly aggregated by the central server, the VFs' placement planning strategy starts. The placement acts on the basis of the VFs popularity, expressed with the popularity vector $\mathbf{p}$. The popularity vector $\mathbf{p}$ has length equal to $\mathcal{V}$ and contains the type of the VFs sorted by descending order on the basis of the occurrence frequency of each VF type in the pool of the whole network requests.

In order to validate the benefits of the proposed framework to the VFs placement problem, we propose a straightforward placement strategy strictly dependent on $\mathbf{p}$. Supposing that the predicted network SRs are given in terms of the VFs' popularity and expressed with the popularity vector $\mathbf{p}$, the VFs' placement is realized through the following steps

1. Process the popularity vector $\mathbf{p}$ starting from the most popular VF in $\mathbf{p}$, i.e., $r^{\star}$, hence from the most requested VF;
2. Deploy $r^{\star}$ on the first CN with enough available SRBs to host $r^{\star}$;
3. Deploy $r^{\star}$ on the cloud if it has enough available SRBs to host $r^{\star}$;
4. If $r^{\star}$ cannot be loaded neither on the CNs nor on the cloud
   (a) if the VF $\hat{r}$ which can be hosted by a CN or cloud does not exist in $\mathbf{p}$, then terminate placement;
   (b) Otherwise repeat steps 1) – 4).

5.7.4 SRs allocation planning

The designed SRs allocation policy is based on the matching theory principles [24, 120], and consider the EUs' perspective. In order to better explain this point, it is important to highlight that the SRs allocation strategy is based on metrics which do not consider the SP revenue, but only the EUs' interests. In this regard, the two parts involved in the matching are the SRs and the computational sites, referred hereafter, for each SR r, as $\mathcal{C}_r$. The set of the computational sites may be different for diverse SRs since, given the SR r, $\mathcal{C}_r$ consists of the CNs which contain the VF requested by r and of the cloud, if this contains the desired VF. Each SR r expresses the preference in being matched, i.e., in being computed, with each element of $\mathcal{C}_r$ and vice versa. The SRs aim at minimizing their own TSA defined as in (5.12), hence they prefer to be executed on computational sites which lower (5.12). By contrast, the computational sites prefer SRs requiring VFs with stringent deadline requirements.

Therefore, the matching algorithm consisting of a modified version of the Gale-Shapley [24] algorithm can be summarized through the following steps

1. Each SR builds its preference on the elements belonging to $\mathcal{C}_r$;

2. Each SR r, proposes to be computed on its most preferred computational site;

3. Each computational site, among the received computational proposals, accepts the SR requiring the VF type with the closest deadline, and discards the other proposals;

4. Update queuing time on each CN;

5. Update preferences of the unallocated SRs;

6. repeat steps 2) $-$ 6) until all the SRs are allocated.

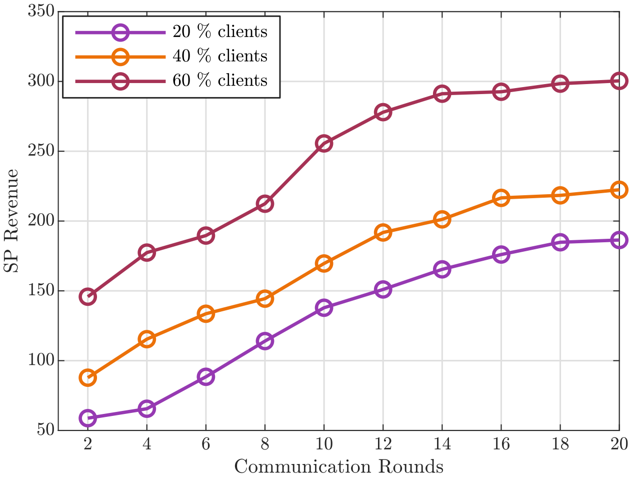


Figure 5.3: SP revenue by varying communication rounds, considering 100 SRs and 20 VFs

5.8 Numerical Results

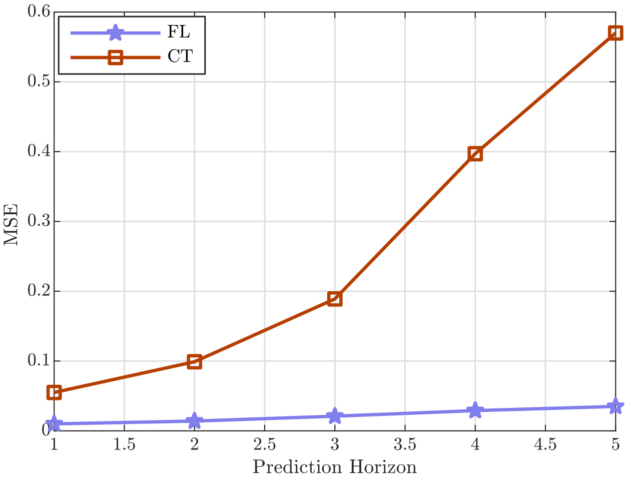


Figure 5.4: MSE by varying the time prediction horizon for type 1 SRs

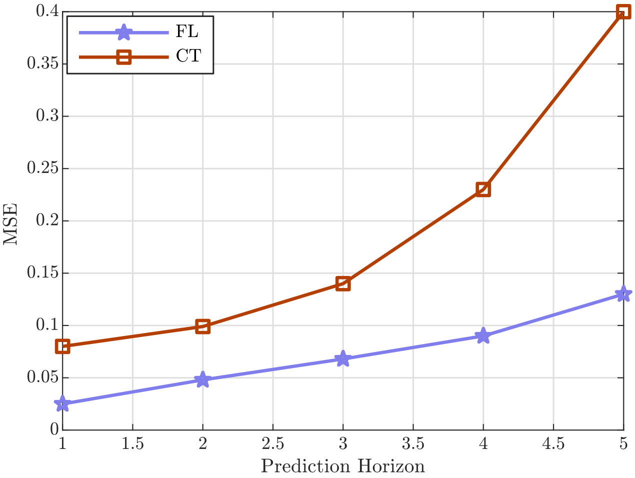


Figure 5.5: MSE by varying the time prediction horizon for type 2 SRs

The proposed FL-based framework has been tested by resorting to numerical simulations in the Tensorflow environment. We supposed an IoE scenario consisting of $\mathcal{N} = 3$ CNs, equipped with a CPU frequency equals to 2.4 GHz, while the cloud has been equipped with a CPU frequency equals to 4.6GHz. Furthermore, we set $S = 70$ and $U = 120$.

The VFs required by SRs have been modeled in a similar way as in [90, 107, 162], and we considered the presence of two priorities, corresponding to the set MovieLens 1M dataset [59] and MovieLens 100K dataset [59], respectively. We modeled 10 VFs, each of which needs a number of SRBs uniformly distributed in $[50, 80]$. All the FL network hyperparameters and the neural architecture have been assumed to be the same as those in [162]. Each SR has been modeled as a number of 64 bits format instructions uniformly distributed in $[250, 800]$, needing 8 CPU cycles per instruction. Furthermore, as loss function, we adopted the Mean Squared Error (MSE) which, for each data ι_ϕ in Θ_χ, is defined as

$$MSE = \frac{1}{\Phi} \sum_{\phi=1}^{\Phi} (\hat{\iota}_\phi - \iota_\phi)^2, \tag{5.22}$$

where Φ is the number of samples in the test data, and $\hat{\iota}_\phi$ is the predicted value. Then, to test the effectiveness of the proposed approach, we made comparison in terms of accuracy of our strategy, with the prediction scheme based on the application of the CT principles by performing the phase space reconstruction method as explained in [92, 139], and by using the predictive model of the k-neighbors discussed in [72]. It is important to note that the CT approach is performed on the central server site, on which all the user data is gathered without considering the preservation of their privacy.

Fig. 5.4 and Fig. 5.5, which exhibit the MSE behavior by varying the prediction horizon, confirm the greater accuracy of the proposed model in comparison to CT. As it is evident in Fig. 5.4 and Fig. 5.5, the MSE grows as the prediction horizon increases. This is a direct consequence of the natural difficulty in predicting the long-term behavior of the series. Nevertheless, both the figures show the superiority of the proposed approach in comparison with the alternative here considered.

Then, Fig. 5.3 makes clear the significant improvement obtained by increasing the number of communication rounds, i.e., information updates, between the server and the clients, for different numbers of EUs involved in the FL process. The direct implication is that higher is the number of the EUs taking part in the learning process, the greater the levels of accuracy on the acquired information on which the VFs placement strategy is based. Moreover, the SP revenue improves its trend. It is important to highlight here that the FL requires a converge time of 12.42 seconds to converge, against the 6.17 seconds required by the CT approach. Fig. 5.6 shows the SP revenue behavior by increasing the number of SRs. As it is straightforward to note, the SP revenue tends to grow by increasing the number of SRs, until the network infrastructure is not saturated and consequently it cannot accept new SRs. Such a situation is clearly a consequence of the physical resources limitation of the network. Finally, Fig. 5.7 depicts the behavior of the percentage of the SRs discarded, i.e., the percentage of the SRs which have not been served by the network infrastructure since their computation is not finished before the expiration of their deadline. In conclusion, the resulting system performance makes clear the validity of the FL application for our problem, highlighting the importance of considering the data expressing the users' preferences and daily habits.

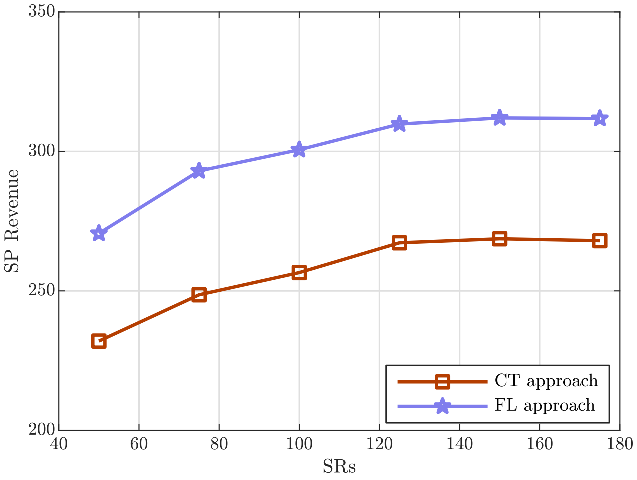


Figure 5.6: SP revenue by varying the number of SRs, considering 10 VFs

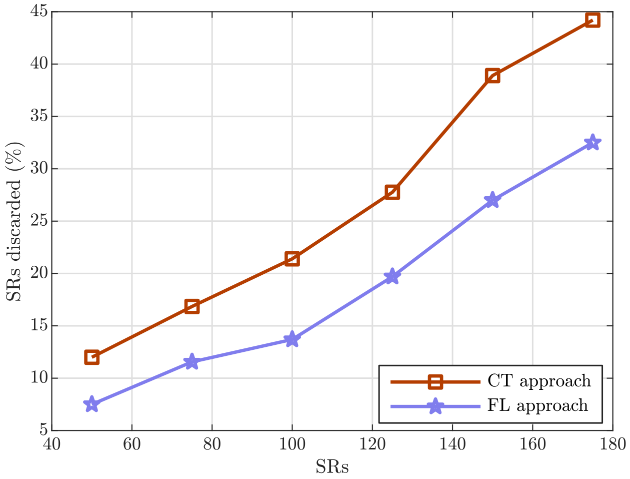


Figure 5.7: Percentage of SRs discarded, by increasing the SR number

5.9 Conclusion

The chapter has dealt with a framework based on the federated learning paradigm to maximise SP revenue, in a hybrid cloud-edge system, arranged to support IoE applications. The proposed framework resorts to the use of the FL approach to predict the SRs demand, in compliance with the users' privacy. Furthermore, a VFs placement on the basis of the obtained SRs demand prediction has been performed and, the related SRs allocation, modeled as a matching game problem, has been hence accomplished. The effectiveness of the proposed framework has been finally validated by providing performance comparisons with an alternative predictive approach based on the chaos theory. In reference to the future research directions, a very interesting topic needing further exploration may be represented by the definition of novel solutions and methodologies to allow the design of privacy-based learning and inference of deep learning and advanced signal processing in heterogeneous hardware architectures. Such a privacy-preserving approach will rely on Homomorphic Encryption that enables processing directly on encrypted data.

Chapter 6

IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option

IPv6 is the core requirement of the IoT networks and making IoTs resilient and secure. It is also mandatory for a network administrator to keep getting information about the deployed IoT network. However, PDM is the right available option present in the Destination Option header of IPv6, which helps the network administrator in this regard. In the initial release, the PDM standard was published but without security protocol, meaning that all the network important information was not encrypted. In This chapter, I present the PDMv2 first draft where HPKE based security mechanism is applied to PDM data. The reason behind choosing HPKE for PDMv2 is that it is robust and flexible for users to implement. HPKE is a framework that includes the KEM, KDF, and AEAD like mechanisms. The draft has been published and can be found on the IETF data tracker website by the following name `draft-elkins-ippm-encrypted-pdmv2`

6.1 Introduction

IETF is a vast and open international community of network designers, operators, suppliers, and academics interested in the growth of Internet architecture and its seamless operations. The IETFs research and technical work are done in different working groups, which are divided into different categories based on their topics (e.g., transport, routing, security, etc.). Area directors are in charge of the IETF regions and they are members of the Internet Engineering Steering Group (IESG). Mailing lists are used extensively in the IETF for discussions or any kind of IETF work. Three times a year, the IETF organizes meetings. IETF Hackathons stimulate the development of utilities, ideas, and solutions that demonstrate how IETF standards may be implemented in the real world. The [19] contains a detailed mission statement of the IETF and information regarding IETF working groups can be found online[1].

IP Performance Measurement (IPPM) is one of the IETF working groups that develops and maintains the standard metrics that can be used to assess the quality, performance, and reliability of the Internet data delivery services and applications that use transport layer protocols such as (TCP and UDP) over IP. Aside from that, it also develops and improves methodologies and procedures for computing these metrics. In 2017, this group published a standard called IPv6 Performance and Diagnostic Metrics (PDM) Destination Option [45]. To assess performance problems in IoTs, this document describes optional headers embedded in each packet that provide sequence numbers and timing information as a basis for measurements. Such measurements may be interpreted in real-time or after the fact. It also specifies the PDM Destination Options header. The field limits, calculations, and usage in the measurement of PDM are included in this document. As discussed in Chapter 1 that to secure the IoTs it is important for the network administrator to understand the deployed network dynamics continuously and for this reason this standard is the right choice. The problem with PDM is that all the information is transmitting in the plain-text. The data is sent in clear text, this may create an opportunity for malicious actors to get information for subsequent attacks. This Chapter defines PDMv2 which has a lightweight handshake (registration procedure) and encryption to secure the data. Additional performance metrics which may be of use are also defined. A similar body like IETF, known as IRTF supports research important to the growth of Internet protocols, applications, architecture, and technology. However, IRTF concentrates on long-term research problems and issues. As a result, research groups have the long-term stability required to foster the growth of research collaboration and teamwork in exploring the research is-

[1] https://datatracker.ietf.org/wg/

sues. Individual contributors, rather than representatives of organizations, are invited to participate. IRTF has fourteen research groups that are currently chartered or proposed for chartering[2].

The Crypto Forum Research Group (CFRG) is working on the HPKE framework, under the umbrella of IRTF. It is also opts by many other research groups (such as [22]) because of its flexibility and strength. So to deal with PDM security I choose the HPKE framework. The CFRG is a general forum for discussing and reviewing uses of cryptographic mechanisms, both for network security in general and for the IETF in particular. It serves as a bridge between theory and practice, bringing new cryptographic techniques to the Internet community and promoting an understanding of the use and applicability of these mechanisms via Informational RFCs (in the tradition of, e.g., MD5 [119] and HMAC [80]. Their aim is to provide a forum for discussing and analyzing general cryptographic aspects of security protocols and to offer guidance on the use of emerging mechanisms and new uses of existing mechanisms.

6.2 Motivation

As discussed above that PDM data can represent a serious data leakage in presence of a malicious actor. In particular, the sequence numbers included in the PDM header allows correlating the traffic flows, and the timing data can highlight the operational limits of a server to a malicious actor. Moreover, forging PDM headers can lead to unnecessary, unwanted, or dangerous operational choices, e.g., to restore an apparently degraded Quality of Service (QoS). Due to this reason, it is important that the confidentiality and integrity of the PDM headers are maintained. The PDM headers can be encrypted and authenticated using the methods discussed in Section 6.7, thus ensuring confidentiality and integrity. However, if PDM is used in a scenario where the integrity and confidentiality are already ensured by other means, they can be transmitted without encryption or authentication. This case will be highly appreciated and useful for the IoTs where underlying protocols have robust security functions like HPKE. because in this manner we can save the energy of IoT devices. However, this includes, but is not limited to, the following cases:

[2] https://irtf.org/

1. PDM is used over an already encrypted medium (For example VPN tunnels).
2. PDM is used in a link-local scenario.
3. PDM is used in a corporate network where there are security measures strong enough to consider the presence of a malicious actor a negligible risk.

6.3 Contribution

Considering the importance of the PDMv1 where there is no security mechanism, attackers can dream to launch active attacks as well as passive attacks. In active attacks, they can trigger unappropriate network management operations, while in passive attacks they free to learn the possible weak points in the entire network. For example, the attacker can launch a DoS attack. The main contribution is to apply the HPKE concepts to PDM to secure the PDM data and a lightweight handshake (registration procedure). Initially, PDMv1 security requirements are data confidentiality and data integrity. I have presented the detailed publish work in the Section 6.7. Considering the IoT networks where constrained nodes have limited power and processing resources, I reduce the HPKE functions call. For example, KEM only used during the registration phase, KDF once in a while and AEAD for every packet.

6.4 Performance and Diagnostic Metrics(PDM)

The PDM is used to assess performance problems in IoT networks. The first standard was published in 2017 [45]. Currently, the IPPM working group working on its enhancement by adding more parameters and security features [46]. The recent work is also present in Section 6.7.

6.4.1 PDMv1

As discussed in Section 2.2.8 of Chapter 2 the PDM option is used to analyze the network performance problems. The information allows the measurement of the *round-trip delay* and *server delay* metrics. It is a “network” delay, that is the delay for packet transfer from a source host to a destination host and then back to the source host [17]. Whereas the server delay is the interval between when a packet is received by a device and the first corresponding packet is sent back in response. This may be "server processing time". It's also possible that acknowledgments are causing the delay. The time it takes for the stack and application to return the answer is included in the server processing time. It's possible that the stack delay is due to network performance. More client-based measurements are required if this aggregate time is viewed as an issue and a clear difference between application processing time and stack delay, including that induced by the network [45].

6.4.2 PDMv2

As discussed in the Section 6.1, PDMv1 standard protocol is lacking the security method to secure the its data. PDMv2 is an ongoing work where I make the security procedure to ensure the confidentiality and integrity of the PDM data. The detailed work is presented in the section 6.7 .

6.5 Hybrid Public Key Encryption

The traditional way of cryptography encrypts the symmetric key with the public key, while the HPKE generates the symmetric key and its encapsulation with the public key. As discussed in Chapter 2, that HPKE is still an ongoing standard. I choose this because of its some variants are functional and implemented[3].

The Figure 6.1 shows the HPKE framework that includes multiple standards such as KEM [81,85], KDF [81], and AEAD [44,111]. An HPKE cipher suite containing a choice of algorithm for each primitive. It is more robust, flexible, and significantly more efficient than traditional cryptography.

Here I discuss four different HPKE modes. All modes take a receiver public key "`pkR`" and plain text "(`pt`)" and generate an encapsulated key "`enc`" and sequence of cipher-texts "(`ct`)". So the owner of the private key (skR) can decapsulate the key from "`enc`" and decrypt the ciphertext (`ct`). All algorithms take "`info`" as a parameter so that can be used to affect the creation of keys (e.g., to fold in identification information) and "`aad`" parameter can be used to give Additional Authenticated Data to the AEAD algorithm in use.

[3] https://github.com/sftcd/happykey

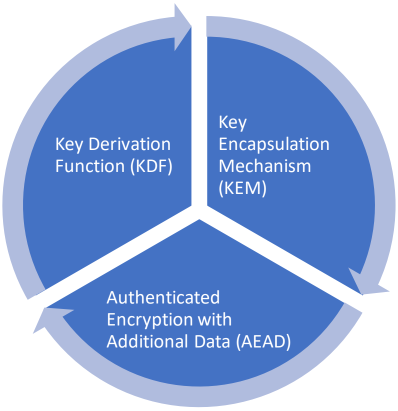

Key Derivation Function (KDF)
Key Encapsulation Mechanism (KEM)
Authenticated Encryption with Additional Data (AEAD)

Figure 6.1: Hybrid Public Key Encryption Framework)

Before going to discuss the HPKE framework and modes, I would like to explain the concept of encapsulation that HPKE is using in [23]. Actually, the HPKE is built using a Key Encapsulation Mechanism (KEM). Encapsulation is one algorithm of a KEM and KEM is like public-key encryption, but you can only encrypt uniformly random data (symmetric keys), and not arbitrary plain texts. The encapsulation algorithm takes a public key "`pkR`", and will return a fresh uniformly random symmetric key (`K`), and an "encryption" (`enc`) of that key(`K`). Because it is a KEM and not public-key encryption, called encapsulation and not encryption, just to distinguish them.

$$k, enc <= Encap(pkR) \tag{6.1}$$

The sender will keep the key(K) for himself and send the encapsulation (enc) to the receiver. The receiver will call the decapsulation algorithm, which will compute the same fresh uniformly random symmetric key.

$$k <= Decap(enc, skR) \tag{6.2}$$

Morally, they call a KEM secure, if no adversary can guess the random symmetric key(`K`) from seeing the encapsulation (`enc`) on the wire.

6.5.1 HPKE Modes

All modes follow the same basic two-step pattern. First, to set up an encryption context (i.e., `info`) that is shared between the sender and the receiver. Second, use that context to encrypt or decrypt data.

1. **Encryption to a Public Key:** This is the basic HPKE mode where a key exchange description and an explicit "`info`" argument given by the caller are coupled with the KEM shared secret through the KDF. Two parameters, "`pkR`" and "`enc`", which are public keys and KEM shared secrets, respectively. Figure 6.2 explains the both encapsulation and decapsulation methods.

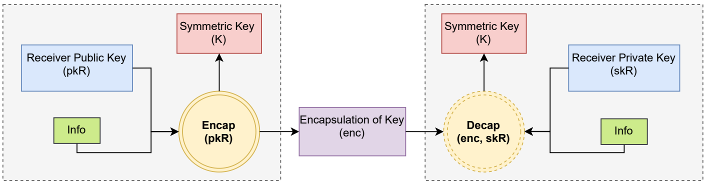

Symmetric Key (K)
Receiver Public Key (pkR)
Info
Encap (pkR)
Encapsulation of Key (enc)
Symmetric Key (K)
Receiver Private Key (skR)
Decap (enc, skR)
Info

Figure 6.2: HPKE Encryption to a Public Key (HPKE Mode-1)

2. **Authentication using a Pre-Shared Key:** It enhances the base mode, this allows the receiver to authenticate that the sender has the specified "`psk`" (Pre-shared key). Pre-shared key also improves the confidentiality, moreover, the main difference from the base mode is that the pre-shared key and "`psk_id`" are used as "`ikm`" input to the KDF. [23].

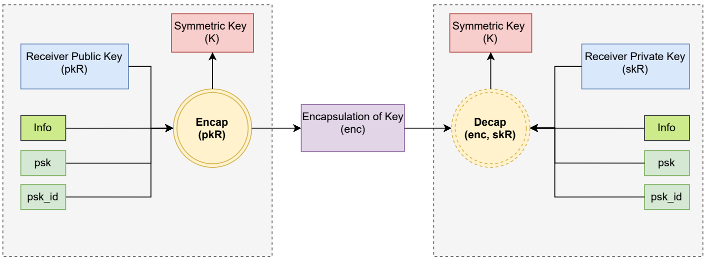


Figure 6.3: Authentication using a Pre-Shared Key (HPKE Mode-3)

3. **Authentication using an Asymmetric Key:** This mode also extends the base mode by allowing the receiver to authenticate that the sender has the KEM private key. the send use its private key "skS" during the encapsulation process, while receiver uses his public key "pkS" to decapsulate and produce the correct Symmetric key "K", as shown in the Figure 6.4. No additional identification is used to authenticate the sender using this method, simply the sender.

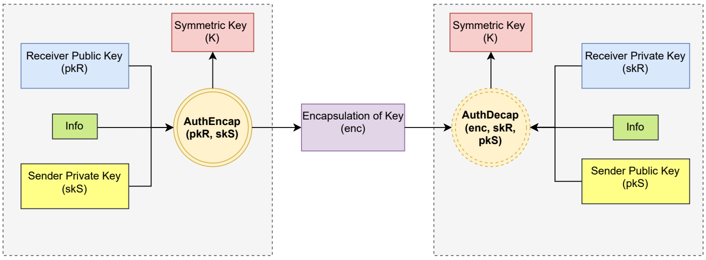


Figure 6.4: Authentication using an Asymmetric Key (HPKE Mode-3)

4. **Authentication using both a Pre-shared and an Asymmetric Key:** This mode combines the mode two and three by injecting "PSK" and "skS" into the authentication encapsulation process, as shown in the Figure 6.5

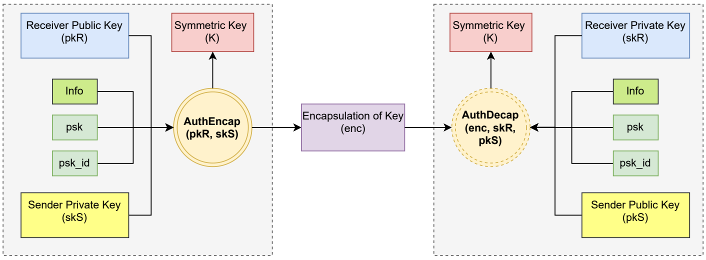


Figure 6.5: Authentication using both a PSK and an Asymmetric Key (HPKE Mode-4)

6.5.2 HPKE Framework

HPKE frame includes three kind of standards KEM [52, 81, 85], KDF [81], and AEAD [44, 111].

1. **Key Encapsulation Mechanisms:** In cryptography, KEMs are a family of encryption techniques meant to protect symmetric cryptographic key material during transmission utilizing asymmetric using public key systems to send lengthy messages is cumbersome in reality. User can use any mode as depicted in above figures 6.2, 6.3, 6.4, and 6.5.

2. **Key Derivation Functions (KDFs):** The basic aim of this type of algorithms are to generate many keys from one source key or master key. This source key is a pseudo-random key.

$$SKorPSK = K_1, K_2, K_3...K_n \tag{6.3}$$

The HPKE draft [23] specifies a simple HMAC-based KDF named HKDF. (see [81]). KDF follows the "extract-then-expand" paradigm, meaning that first it extracts the key then it expands into multiple keys. The *Extract* function where the `Input Keying Material (IKM)` and `Salt` are the inputs to produces the `pseudo-random key (PRK)`. The second function is *Expand* where extracted `PRK`, `info`, and `L` goes as inputs and creates `Output Keying Material (OKM)`. The `info` is the context that means that if both parties want the resulting key to be only used in a certain context. More precisely, to bind the key to a certain context. We want the protocol to work only if both parties actually think that they are talking to each other and for the same purpose. Where `L` denotes the length of the `OKM`. Figure 6.6 depicts the full overview of both functions.

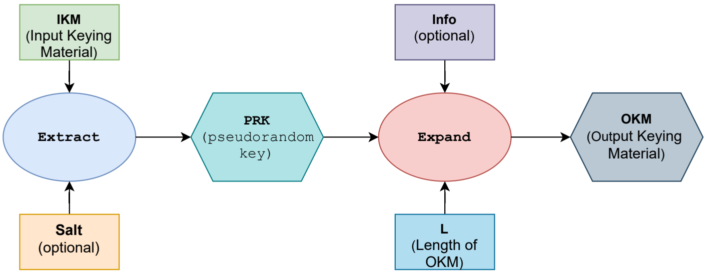


Figure 6.6: Key Derivation Function

3. **Authenticated Encryption with Associated Data (AEAD) Func-**

tions: This type of algorithm contains the data to be authenticated, but not encrypted. because the associated data is used to protect information that needs to be authenticated but does not need to be kept confidential. An AEAD can be used to secure a network protocol, for instance, by supplying inputs such as addresses, port numbers, sequence numbers, protocol version numbers, and other fields that specify how plain text or cipher-text should be handled, transmitted, or processed [97]. For encryption and decryption, it is a known string that must be supplied at both ends. As a result, the decryption will fail if the decryptor uses an incorrect string as input. The question arises that why AAD (Associated Data) is necessary, well, we often utilize the same encryption keys to encrypt several communications. If the decryptor utilizes different AADs for the two contexts, then the decryption will fail (and the attack is prevented; if the attacker just wanted the decryption to fail, he or she could just replace the cipher-text with random gibberish). As an alternative, we could merely use a different key each time, however, AAD handles the problem more efficiently. If AEAD is not used then attackers are able to do replace one cipher-text with another.

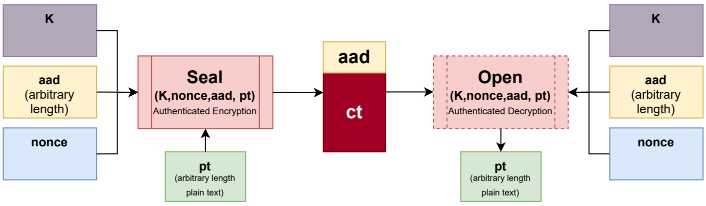


Figure 6.7: Authenticated Encryption with Associated Data (AEAD) Function

The *Nonce* is a time-varying value with a very probability of repeating, such as a random number produced once for each usage. It can be a timestamp, counter, sequence number, Linear Feedback Shift Register (LFSR), or a combination of these [111]. The AEAD algorithm uses two main functions, first is *Seal* function and second is*Open* function. In Seal function, encryption and authentication of the plain-text "`pt`" with associated data "`aad`" using symmetric key "`K`" and nonce "`nonce`", yielding cipher-text "`ct`" and tag. The Open functions is the opposite to get the plain text as depicted in the Figure 6.7.

6.6 Conclusion

To ensure the data security (such as confidentiality and integrity) of the IoT devices, it is mandatory to develop a robust and lightweight security mechanism. Considering the constrained nodes, the HPKE is very flexible and lightweight in terms of the key sizes selection and many other features [23]. To know the IoT network dynamics at different time intervals it is important to get benefits from PDM features. Moreover, a new field that PDMv2 introduces called *Global Pointer* provides a measure of the amount of traffic being processed by the PDMv2 node also helps the network administrator that which IoT node is doing malfunctioning. Securing PDM data and adding news functions make IoT networks more robust and resilient.

6.7 PDMv2 Draft 01

What follows is the transcription of the PDMv2 draft, at the moment of writing. The actual drafts can be found at https://datatracker.ietf.org/doc/draft-elkins-ippm-encrypted-pdmv2/.

Internet Engineering Task Force
Internet-Draft
Intended status: Proposed Standard
Expires: 22 April 2022

N. Elkins
Inside Products, Inc.
M. Ackermann
BCBS Michigan
A. Deshpande
NITK Surathkal
T. Pecorella
A. Rashid
University of Florence
19 October 2021

IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option
draft-elkins-ippm-encrypted-pdmv2-01.txt

Abstract

RFC8250 describes an optional Destination Option (DO) header embedded in each packet to provide sequence numbers and timing information as a basis for measurements. As this data is sent in clear- text, this may create an opportunity for malicious actors to get information for subsequent attacks. This document defines PDMv2 which has a lightweight handshake (registration procedure) and encryption to secure this data. Additional performance metrics which may be of use are also defined.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 April 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

1.1. Current Performance and Diagnostic Metrics (PDM)

The current PDM is an IPv6 Destination Options header which provides information based on the metrics like Round-trip delay and Server delay. This information helps to measure the Quality of Service (QoS) and to assist in diagnostics. However, there are potential risks involved transmitting PDM data during a diagnostics session.

PDM metrics can help an attacker understand about the type of machine and its processing capabilities. Inferring from the PDM data, the attack can launch a timing attack. For example, if a cryptographic protocol is used, a timing attack may be launched against the keying material to obtain the secret.

Along with this, PDM does not provide integrity. It is possible for a Man-In-The-Middle (MITM) node to modify PDM headers leading to incorrect conclusions. For example, during the debugging process using PDM header, it can mislead the person showing there are no unusual server delays.

1.2. PDMv2 Introduction

PDMv2 introduces confidential, integrity and authentication.

TBD

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] .

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

3. Terminology

* Primary (Writer) Client (WC): An authoritative node that creates cryptographic keys for multiple reader clients.

* Primary (Writer) Server (WS): An authoritative node that creates cryptographic keys for multiple reader servers.

* Secondary (Reader) Client (RC): An endpoint node which initiates a session with a listening port and sends PDM data. Connects to the Primary (Writer) Client to get cryptographic key material.

* Secondary (Reader) Server (RS): An endpoint node which has a listening port and sends PDM data. Connects to the Primary (Writer) Server to get cryptographic key material.

Note: a client may act as a server (have listening ports).

* Symmetric Key (K): A uniformly random bitstring as an input to the encryption algorithm, known only to Secondary (Reader) Clients and Secondary (Reader) Servers, to establish a secure communication.

* Public and Private Keys: A pair of keys that is used in asymmetric cryptography. If one is used for encryption, the other is used for decryption. Private Keys are kept hidden by the source of the key pair generator, but Public Key is known to everyone. pkX (Public Key) and skX (Private Key). Where X can be, any client or any server.

* Pre-shared Key (PSK): A symmetric key. Uniformly random bitstring, shared between any client or any server or a key shared between an entity that forms client-server relationship. This could happen through an out-of band mechanism: e.g., a physical meeting or use of another protocol.

* Session Key: A temporary key which acts as a symmetric key for the whole session.

4. Protocol Flow

The protocol will proceed in 3 steps.

Step 1: Negotiation between Primary (Writer) Server and Primary (Writer) Client.

Step 2: Registration between Primary (Writer) Server / Client and Secondary (Reader) Server / Client

Step 3: PDM data flow between Secondary (Reader) Client and Secondary (Reader) Server

After-the-fact (or real-time) data analysis of PDM flow may occur by network diagnosticians or network devices. The definition of how this is done is out of scope for this document.

4.1. Registration Phase

4.1.1. Rationale of Primary (Writer) and Secondary (Reader) Roles

Enterprises have many servers and many clients. These clients and servers may be in multiple locations. It may be less overhead to have a secure location (ex. Shared database) for servers and clients to share keys. Otherwise, each client needs to keep track of the keys for each server.

Please view Appendix 1 for some sample topologies and further explanation.

4.1.2. Diagram of Registration Flow

```
           +------------+                       +------------+
           |   Writer   |<--------------------->|   Writer   |
           |   Client   |                       |   Server   |
           +------+-----+                       +------+-----+
                  |                                    |
       +----------+----------+              +----------+----------+
       |          |          |              |          |          |
   +---+---+  +---+---+  +---+---+      +---+---+  +---+---+  +---+---+
   | Reader|  | Reader|  | Reader|      | Reader|  | Reader|  | Reader|
   |   1   |  |   2   |  |   3   |      |   1   |  |   2   |  |   3   |
   +---+---+  +---+---+  +---+---+      +---+---+  +---+---+  +---+---+
       |          |          |              |          |          |
       |          |          +--------------+          |          |
       |          +------------------------------------+          |
       +----------------------------------------------------------+
```

4.2. Primary (Writer) Client - Primary (Writer) Server Negotiation Phase

The two entities exchange a set of data to ensure the respective identities.

They use HPKE KEM to negotiate a "SharedSecret".

4.3. Primary (Writer) Server / Client - Secondary (Reader) Server / Client Registration Phase

The "SharedSecret" is shared securely:

* By the Primary (Writer) Client to all the Secondary (Reader) Clients under its control. How this is achieved is beyond the scope of the present specification.

* By the Primary (Writer)Server to all the Secondary (Reader) Servers under its control. How this is achieved is beyond the scope of the present specification.

4.4. Secondary (Reader) Client - Secondary (Reader) Server communication

Each Client and Server derive a "SessionTemporaryKey" by using HPKE KDF, using the following inputs:

* The "SharedSecret".

* The 5-tuple (SrcIP, SrcPort, DstIP, DstPort, Protocol) of the communication.

* A Key Rotation Index (Kri).

The Kri is initialized to zero.

The server and client initialize (separately) a pseudo-random non-repeating sequence between 1 and 2^15-1. How to generate this sequence is beyond the scope of this document, and does not affect the rest of the specification. When the sequence is used fully, or earlier if appropriate, the sender signals the other party that a key change is necessary. This is achieved by flipping the "F bit" and resetting the PRSEQ. The receiver increments the Kri of the sender, and derives another SessionTemporaryKey to be used for decryption.

It shall be stressed that the two SessionTemporaryKeys used in the communication are never the same, as the 5-tuple is reversed for the Server and Client. Moreover, the time evolution of the respective Kri can be different. As a consequence, each entity must maintain a table with (at least) the following informations:

- Flow 5-tuple, Own Kri, Other Kri

An implementation might optimize this further by caching the OwnSessionTemporaryKey (used in Encryption) and OtherSessionTemporaryKey (used in Decryption).

5. Security Goals

As discussed in the introduction, PDM data can represent a serious data leakage in presence of a malicious actor.

In particular, the sequence numbers included in the PDM header allows correlating the traffic flows, and the timing data can highlight the operational limits of a server to a malicious actor. Moreover, forging PDM headers can lead to unnecessary, unwanted, or dangerous operational choices, e.g., to restore an apparently degraded Quality of Service (QoS).

Due to this, it is important that the confidentiality and integrity of the PDM headers is maintained. PDM headers can be encrypted and authenticated using the methods discussed in section [x], thus ensuring confidentiality and integrity. However, if PDM is used in a scenario where the integrity and confidentiality is already ensured by other means, they can be transmitted without encryption or authentication. This includes, but is not limited to, the following cases:

a) PDM is used over an already encrypted medium (For example VPN tunnels).

b) PDM is used in a link-local scenario.

c) PDM is used in a corporate network where there are security measures strong enough to consider the presence of a malicious actor a negligible risk.

5.1. Security Goals for Confidentiality

PDM data must be kept confidential between the intended parties, which includes (but is not limited to) the two entities exchanging

PDM data, and any legitimate party with the proper rights to access such data.

5.2. Security Goals for Integrity

PDM data must not be forged or modified by a malicious entity. In other terms, a malicious entity must not be able to generate a valid PDM header impersonating an endpoint, and must not be able to modify a valid PDM header.

5.3. Security Goals for Authentication

TBD

5.4. Cryptographic Algorithm

Symmetric key cryptography has performance benefits over asymmetric cryptography; asymmetric cryptography is better for key management. Encryption schemes that unite both have been specified in [RFC1421], and have been participating practically since the early days of public-key cryptography. The basic mechanism is to encrypt the symmetric key with the public key by joining both yields. Hybrid public-key encryption schemes (HPKE) [Draft-12] used a different approach that generates the symmetric key and its encapsulation with the public key of the receiver.

Our choice is to use the HPKE framework that incorporates key encapsulation mechanism (KEM), key derivation function (KDF) and authenticated encryption with associated data (AEAD). These multiple schemes are more robust and significantly efficient than the traditional schemes and thus lead to our choice of this framework.

6. PDMv2 Destination Options

6.1. Destinations Option Header

The IPv6 Destination Options extension header [RFC8200] is used to carry optional information that needs to be examined only by a packet's destination node(s). The Destination Options header is identified by a Next Header value of 60 in the immediately preceding header and is defined in RFC 8200 [RFC8200]. The IPv6 PDMv2 destination option is implemented as an IPv6 Option carried in the Destination Options header.

6.2. Metrics information in PDMv2

The IPv6 PDMv2 destination option contains the following base fields:

SCALEDTLR: Scale for Delta Time Last Received
SCALEDTLS: Scale for Delta Time Last Sent
GLOBALPTR: Global Pointer
PSNTP: Packet Sequence Number This Packet
PSNLR: Packet Sequence Number Last Received
DELTATLR: Delta Time Last Received
DELTATLS: Delta Time Last Sent

PDMv2 adds a new metric to the existing PDM [RFC8250] called the Global Pointer. The existing PDM fields are identified with respect to the identifying information called a "5-tuple".

The 5-tuple consists of:

SADDR: IP address of the sender
SPORT: Port for the sender
DADDR: IP address of the destination
DPORT: Port for the destination
PROTC: Upper-layer protocol (TCP, UDP, ICMP, etc.)

Unlike PDM fields, Global Pointer (GLOBALPTR) field in PDMv2 is defined for the SADDR type. Following are the SADDR address types considered:

a) Link-Local

b) Global Unicast

The Global Pointer is treated as a common entity over all the 5-tuples with the same SADDR type. It is initialised to the value 1 and increments for every packet sent. Global Pointer provides a measure of the amount of IPv6 traffic sent by the PDMv2 node.

When the SADDR type is Link-Local, the PDMv2 node sends Global Pointer defined for Link-Local addresses, and when the SADDR type is Global Unicast, it sends the one defined for Global Unicast addresses.

6.3. PDMv2 Layout

PDMv2 has two different header formats corresponding to whether the metric contents are encrypted or unencrypted. The difference between the two types of headers is determined from the Options Length value.

Following is the representation of the unencrypted PDMv2 header:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  | Option Length | Vrsn  |     Reserved Bits     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Random Number         |f|  ScaleDTLR    |   ScaleDTLS    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Global Pointer                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PSN This Packet          |    PSN Last Received          |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Delta Time Last Received    |     Delta Time Last Sent      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Following is the representation of the encrypted PDMv2 header:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  | Option Length | Vrsn  |     Reserved Bits     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Random Number          |f|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               :
|                     Encrypted PDM Data                        :
:                         (30 bytes)                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Type

0x0F

8-bit unsigned integer. The Option Type is adopted from RFC 8250 [RFC8250].

Option Length

0x12: Unencrypted PDM

0x22: Encrypted PDM

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. The options length is used for differentiating PDM [RFC8250], unencrypted PDMv2 and encrypted PDMv2.

Version Number

0x2

4-bit unsigned number.

Reserved Bits

12-bits.

Reserved bits for future use. They are initialised to 0 for PDMv2.

Random Number

15-bit unsigned number.

TBD

Flag Bit

1-bit field.

TBD

Scale Delta Time Last Received (SCALEDTLR)

8-bit unsigned number.

This is the scaling value for the Delta Time Last Sent (DELTATLS) field.

Scale Delta Time Last Sent (SCALEDTLS)

8-bit unsigned number.

This is the scaling value for the Delta Time Last Sent (DELTATLS) field.

Global Pointer

32-bit unsigned number.

Global Pointer is initialized to 1 for the different source address types and incremented monotonically for each packet with the corresponding source address type.

This field stores the Global Pointer type corresponding to the SADDR type of the packet.

Packet Sequence Number This Packet (PSNTP)

16-bit unsigned number.

This field is initialized at a random number and is incremented monotonically for each packet of the 5-tuple.

Packet Sequence Number Last Recieved (PSNLR)

16-bit unsigned number.

```
         This field is the PSNTP of the last received packet on the
         5-tuple.



Elkins, et al.            Expires 22 April 2022               [Page 11]

Internet-Draft    draft-elkins-ippm-encrypted-pdmv2-01      October 2021


      Delta Time Last Received (DELTATLR)

         16-bit unsigned integer.

         The value is set according to the scale in SCALEDTLR.

         Delta Time Last Received =
         (send time packet n - receive time packet (n - 1))

      Delta Time Last Sent (DELTATLS)

         16-bit unsigned integer.

         The value is set according to the scale in SCALEDTLS.

         Delta Time Last Sent =
         (receive time packet n - send time packet (n - 1))

7.  Security Considerations

   TBD

8.  Privacy Considerations

   TBD

9.  IANA Considerations

   This memo includes no request to IANA.

10.  Contributors

   TBD

11.  References
```

11.1. References

11.2. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <https://www.rfc-editor.org/info/rfc8250>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

11.3. Informative References

[Draft-12] Barnes, R. (et al), "Hybrid Public Key Encryption", draft-irtf-cfrg-hpke-12, Sep, 2021, Work In Progress, draft-irtf-cfrg-hpke-12 - Hybrid Public Key Encryption

[RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <https://www.rfc-editor.org/info/rfc1421>.

Appendix A. Rationale for Primary (Writer) Server / Primary (Writer) Client

A.1. One Client / One Server

Let’s start with one client and one server.

```
     +------------+  Derived Shared Secret  +------------+
     |   Client   |    ----------------->   |   Server   |
     +------+-----+                         +------+-----+
            |                                      |
            V                                      V
      Client Secret                         Server Secret
```

The Client and Server create public / private keys and derive a shared secret. Let’s not consider Authentication or Certificates at this point.

What is stored at the Client and Server to be able to encrypt and decrypt packets? The shared secret or private key.

Since we only have one Server and one Client, then we don’t need to have any kind of identifier for which private key to use for which Server or Client because there is only one of each.

Of course, this is a ludicrous scenario since no real organization of interest has only one server and one client.

A.2. Multiple Clients / One Server

So, let’s try with multiple clients and one Primary (Writer) server

```
     +------------+
     |  Client 1  |  --------+
     +------------+          |
     +------------+          +--->
     |  Client 2  |    ---------->    +------------+
     +------+-----+         :         |   Server   |
            :               :         +------+-----+
            :               +---->
     +------------+         |
     |  Client n  |  -------+
   +------+-----+
```

The Clients and Server create public / private keys and derive a shared secret. Each Client has a unique private key.

What is stored at the Client and Server to be able to encrypt and decrypt packets?

Clients each store a private key. Server stores: Client Identifier and Private Key.

Since we only have one Server and multiple Clients, then the Clients don't need to have any kind of identifier for which private key to use for which Server but the Server needs to know which private key to use for which Client. So, the Server has to store an identifier as well as the Key.

But, this also is a ludicrous scenario since no real organization of interest has only one server.

A.3. Multiple Clients / Multiple Servers

When we have multiple clients and multiple servers, then each not only does the Server need to know which key to use for which Client, but the Client needs to know which private key to use for which Server.

A.4. Primary (Writer) Client / Primary (Writer) Server

Based on this rationale, we have chosen a Primary (Writer) Server / Primary (Writer) Client topology.

Appendix B. Change Log

Note to RFC Editor: if this document does not obsolete an existing RFC, please remove this appendix before publication as an RFC.

Appendix C. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Nalini Elkins
Inside Products, Inc.
36A Upper Circle
Carmel Valley, CA, 93924
United States of America

Phone: +1 831 234 4232
Email: nalini.elkins@insidethestack.com

Michael Ackermann
BCBS Michigan
P.O. Box 2888
Detroit, Michigan, 48231
United States of America

Phone: +1 248 703 3600
Email: mackermann@bcbsm.com
URI: http://www.bcbsm.com

Ameya Deshpande
NITK Surathkal
Pashan-Baner Link Road, Pashan
Pune, Maharashtra, 411021
India

Phone: +91 96893 26060
Email: ameyanrd@gmail.com
URI: https://www.nitk.ac.in/

Tommaso Pecorella
University of Florence
Dept. of Information Engineering, Via di Santa Marta, 3, 50139
Firenze
Italy

Phone: +39 055 2758540
Email: tommaso.pecorella@unifi.it
URI: https://www.unifi.it/

Adnan Rashid
University of Florence
Dept. of Information Engineering, Via di Santa Marta, 3, 50139
Firenze
Italy

Phone: +39 347 9821 467
Email: adnan.rashid@unifi.it
URI: https://www.unifi.it/

Chapter 7

HYDROCONTROLLER and IMPRESAR & S4.0 project

This chapter is about the two research projects that I work during my Ph.d studies. First is known as HYDROCONTROLLER and second is known as IMPRESAR & S4.0.

7.1 HYDROCONTROLLER Project

This project is about to design of a hydrological basin monitoring system based on the IoTs approaches with application to the HYDROCONTROLLER Project scenarios. An automated IT platform for the monitoring and prediction of water resources on hydrologic basins that allow us to control the Hydro-meteorological conditions of a region of interest on the progress in real-time and possible developments in the future. The platform is made up of heterogeneous networks as well as heterogeneous input data. For example, satellite observation data, forecasting data, data coming from sensors, or Ad Hoc networks. The aim behind this project is, to date we have different systems available in the market and working to help mankind to be aware of destructions or by informing us to use the water resource timely. But those systems are not broadly addressing the issues which are discussed in the proposal. Because some systems are dedicated and specialized in certain conditions. For example, systems for monitoring a river basin don't take into account the weather conditions. Similarly, systems to prevent environmental destruction doesn't consider the soil structure. Moreover, they are not scalable and reliable. The operational objective that involved me was mainly OO3. It proposed the design, development, and integration of radio equipment (Wireless Gateway) to interface the sensors specifically developed within O2, and interconnect actuator remote controls, safety, and remote-control devices in the center of monitoring local and/or remote control also through public IP networks.

7.2 IMPRESAR & S4.0

The idea arises from the need felt by companies, especially Small and medium-sized enterprises (SMEs), to want to become "Enterprise 4.0", wanting to bring together in this definition companies that carry out research and development projects when they digitize the company to get to industry 4.0. (COMPANY 4.0 = R & D + INDUSTIRA 4.0). All the companies involved in the project are making investments in plants and machinery suitable for industry 4.0 but they wish nevertheless to combine this effort with a project that sees companies engaged in research and development activities for new processes and / or new products, thanks to the skills of the research groups involved, which enhance industry 4.0 for the transition to enterprise 4.0. Cross-cutting activity in the various sectors of the companies involved in the project is an added value and finds a common thread in the approach to industry 4.0, here in fact the knowledge of different sectors are poured into a common project.

The development of the project is allowing each company to draw on knowledge typical of other sectors, useful for enhancing the Tuscan products in a globalized world, and to take advantage of university knowledge for the definition of a correct R&D activity in enterprise 4.0 perspective on the basis of specific needs. The companies will develop the identified projects and highlight the problems in periodic meetings aimed at evaluating the common difficulties as well as the exchange of information and skills necessary for the emergence of positive synergies between companies belonging to different sectors.

The project is therefore mainly focus on defining the activities to be implemented for the integration of investments for industry 4.0 with the R&D activities and their implementation in the contexts of the individual companies involved. A fundamental aspect is therefore not only the collaboration between companies belonging to different sectors, but also the interrelationship between these and the Universities of Siena and Florence which, in addition to playing an important role in the definition of the strategy, help companies in the development of research and in predicting and monitoring expected results.

7.3 Software-Defined Networking

Decoupling of the control plane from the data plane is the foundation of the SDN. The concept of the SDN made for stable and fixed networks instead of intermittent networking. The interface between the data plane and control plane is made by different organizations and research groups, but only well-known transport security is provided. SDN-WISE [53] is one prominent framework made for the constrained devices but lack of security, devices only rely on IEEE 802.15.4 MAC layer security. Where SDN gives privileges to network administrators to solve top-level security problems it also opens two types of threats. For example, threats to three layers such as *Application Plane*, *Control Plane*, and *Data Plane* and threats to interfaces between layers i.e. *North, South, East*, and *Westbound* interfaces.

Computer and network security protocols, technologies, and policies have developed and matured over the past decades, tailored to the needs of enterprises, governments, and other users. Although there is an ongoing arms race between attackers and defenders, it is possible to build a powerful security facility for traditional networks and for SDN/NFV networks. The sudden explosion of IoT networks with millions to billions of devices poses an unprecedented security challenge. Different models and frameworks produced by different standard organizations can serve as a foundation for the design and implementation of an IoT security facility.

7.3.1 SDN Security: Motivation

The innovative LLN design that I proposed in the project is the same as discussed in the Chapter 4. To increase the resilience of LLN, I opt the well-known technologies such SDN and NFV. In this section, I highlight traditional network problem and the motivation of the SDN security in the below illustration:

Figure 7.1 is a mesh topology and assuming that the OSPF protocol is running in each router. Keep in mind that OSPF has a high convergence

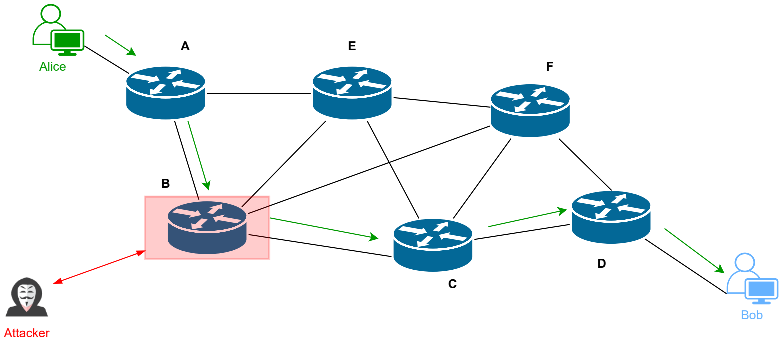


Figure 7.1: Motivation to Software Defined Networks

rate but on the other hand, due to its flooding mechanism, it congests the entire network. Alice and Bob are sender and receiver respectively, and their shortest path is established via router A, B, C, and D. If attacker attacks on router B, as a man-in-middle attack or generate a DoSs attack then there is no solution for this entire network to change the path, if somehow IDSs detect the attack then there will be a huge loss of the important information shared between sender and receiver. During the attack detection and then mitigation a time involves clearing the path. There is no entity to update the other routers to accommodate the current communication between sender and receiver. Hence, there is a need for a centralized system called a controller which acts promptly on such malicious activities. Where I can install the IDSs, Intrusion Prevention Systems (IPSs) even routing and some other required function by using the NFV technology. The decoupling of the control plane and data plane of the router gives us to make the network robust, available, and more secure than the traditional environment. This idea is known as SDN. The routing functionality and other forecasting can easily do if I separate the control plane from the data plane. As illustrated in the below diagram.

7.3.2 SDN Security

SDN represents a significant departure from traditional network architecture and may not mesh well with existing network security approaches. It involves a three-layer architecture and new techniques for network control. All of this introduces the potential for new targets for attack. This section will explain

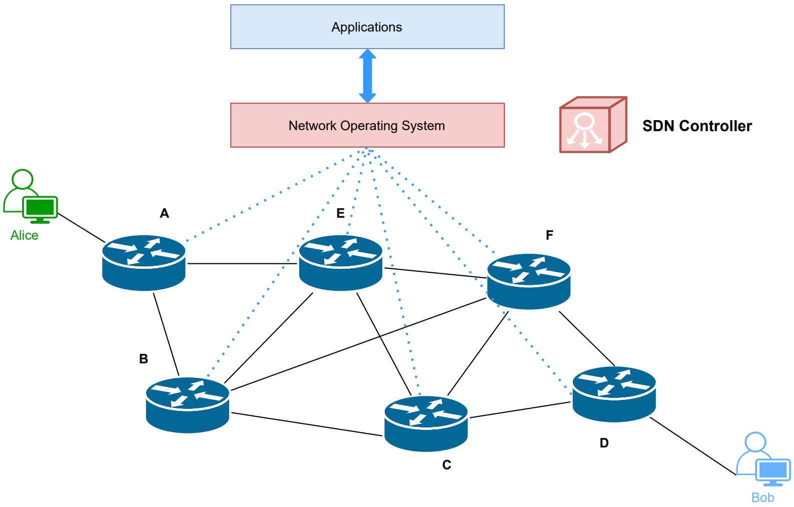


Figure 7.2: Concept of decoupling of control plane and data plane

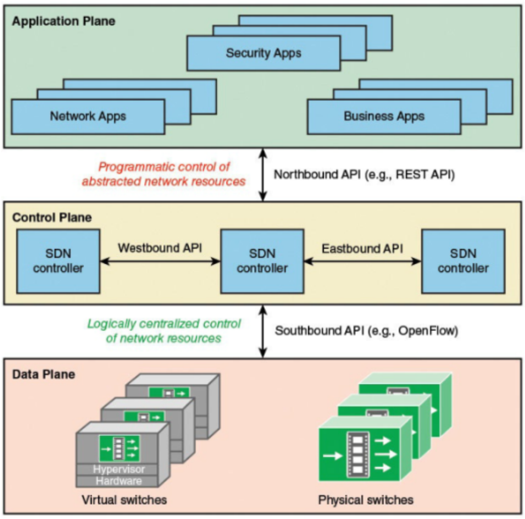


Figure 7.3: Software Defined Architecture [136]

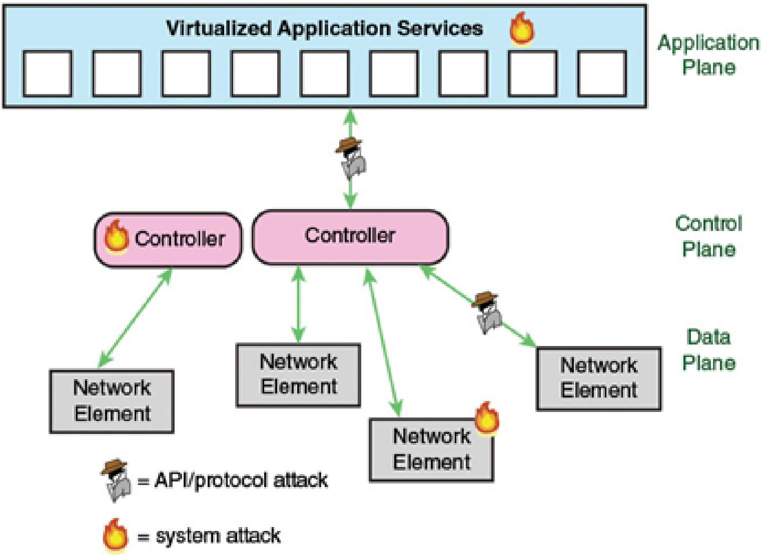


Figure 7.4: SDN Security Attack Surfaces paradigm. [136]

the SDN security from two points of view, First, the security threats to SDN and Second is the use of SDN to enhance network security.

Threats to SDN

Basically, there two attack surface areas in SDN architecture. First, at the Layers level, where hardware/software platforms at any layer are potential targets for malware or intruder attacks. Second, Interfaces, where the protocols and application programming interfaces (APIs) related to SDN provide a new target for security attacks. As demonstrated in the Figure 7.4.

In SDN architecture, there are two attack surface regions. First, at the Layers level, where any hardware/software platform might be a target for malware or intruder assaults at any layer. Second, the protocols and application programming interfaces (APIs) associated with SDN create a new target for security threats. As seen in Figurer 7.4.

Data Plane

1. **Data Plane Security Threats:** The southbound API, such as OpenFlow and Open vSwitch Database Management Protocol, is a significant area of risk on the data plane (OVSDB). Because security is no longer restricted to the network equipment supplier, this API is a useful tool for controlling data plane network elements. It also expands the attack surface of the network infrastructure significantly. If the southbound protocol is implemented in an insecure manner, the network's security is jeopardized. As a result, attackers may be able to insert their own flows into the flow table and impersonate traffic that would otherwise be blocked on the network. The following are potential threats:

   - Attackers can add their own flows into the flow table by deploying the DoS attacks, for example by SYN flooding.
   - Attacker can jam/block the bandwidth between switch and controller.
   - Attackers can control the network elements.
   - Overload switch's flow table and memory and consume the Central Processing Unit (CPU) and memory of the controller by flooding table-miss packets.

2. **Data Plane Security:**

   The usage of TLS, which developed from the previous Secure Sockets Layer (SSL), allows SDN to improve security. With TLS enabled, an application receives an TLS socket address and connects with the remote application's TLS socket. As a result, the application and the TCP connection are completely unaware of the end-to-end security features supplied by TLS. As a result, neither TCP nor the application must be changed in order to use the security capabilities of TLS. Not only does TLS support Hypertext Transfer Protocol (HTTP), but it also supports any other application that uses TCP.

   Confidentiality, Message Integrity, and Authentication are the three security categories provided by TLS. TLS is divided into two phases: handshake and data transmission, in which the two parties execute authentication and establish an encryption key for data transport. Second, the encryption key is used by both parties to encrypt all transferred data during data transfer. The interface between the data route and the OpenFlow channel is implementation-specific, but all OpenFlow channel messages must follow the OpenFlow switch protocol.

TLS provides three categories of security, Confidentiality, Message integrity and Authentication. TLS consists of two phases, First, handshake and data transfer, where, the two sides perform an authentication function and establish an encryption key to be used for data transfer. Second, during data transfer, the two sides use the encryption key to encrypt all transmitted data. Between the data path and the OpenFlow channel, the interface is implementation-specific, however all OpenFlow channel messages must be formatted according to the OpenFlow switch protocol. The OpenFlow channel is often secured with TLS, although it can also be performed through TCP. Nonetheless, TLS or a similar capability is required since it is difficult to protect the data plane without also securing the southbound communication channel (between the control plane and the data plane).

Control Plane

1. **Control Plane Security Threats:**

   A single controller or a few distributed controllers handle overall administration, orchestration, routing, and other elements of network traffic flow. If an attacker can breach a controller, he or she will have a significant amount of influence over the whole network. As a result, the SDN controller is a high-value target that requires extra security.

2. **Control Plane Security** The standard repertory of computer security approaches, for example, can be used to protect the controller.

   - Prevention or protection against DDoS attacks a high-availability controller architecture could go some way to mitigating a DDoSs attack by using redundant controllers to make up for the loss of other controllers.
   - For the access control several standard access control technologies can be employed, including Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC).
   - Antivirus or antiworm techniques.

- Firewalls, IDSs, and IPSs)

Application Plane

1. **Application Plane Security Threats:** An attacker might acquire control of the networking infrastructure if this assault is successful. As a result, in this area, SDN security is focused on preventing unauthorized users and programs from abusing the controller. The applications themselves represent a weak spot in the system. The amount of harm that may be done if an attacker gains control of an application and that application is then authenticated to the control plane is significant. An authorized application with a broad set of rights can exert a lot of control over the network's setup and operation.

2. **Application Plane Security:** There are two options for dealing with these threats: The control plane access of an application must first be authenticated. All communication between applications and the controller must be encrypted using TLS or a comparable protocol to prevent against dangers during the authentication process. Second, check to see whether this approved application has been hacked. To be secured, applications must be coded securely and the application platform must be secure against hackers.

7.3.3 Software Defined Security

Now I'll go on to our second topic, which is how SDN can improve traditional networks or how SDN can improve network security. True, SDN offers additional security issues for network designers and administrators, but it also provides a platform for establishing network security policies and processes that are uniform and centrally managed. SDN enables the creation of security controllers and apps that can supply and coordinate security services and procedures.

Security controllers must provide a secure API for relevant apps in order to manage security. When an application builds a virtual machine (VM) and configures traffic pathways, for example, it must be able to associate the virtual components with security capabilities such as intrusion detection, intrusion prevention, and security information and event management (SIEM).

In fact, security demands may turn out to be one of the key motivating factors for deploying SDN. On the one hand, key networking trends place an increasing burden on system and networking administrators, including the following:

- The increase in network traffic volume
- The use of VMs for servers, storage, and networking devices
- Cloud computing
- The growth in the size and complexity of data centers
- The growth of IoT applications

Malware, on the other side, is becoming more agile and sophisticated. As a result, IT personnel becomes a significant security bottleneck. Security managers are finding it difficult to keep up with the growing number of events and warnings, as well as the need to fine-tune security measures in response. Through intelligent incident detection and automated response, SDN enables security administrators to bridge this response resource gap.

The capacity to respond on a granular level, such as per flow, per application, or per user, is an advantage in and of itself when using SDN-enabled automated tools.

7.4 Conclusion

Both projects are very helping to gain in-depth knowledge of devices and simulators. The research paper that I produce is presented in the Chapter 4 that is the outcome of both projects. The details of the experiments and results are omitted in this chapter for the sake of brevity and can be found in the Chapter 4. Furthermore, the platform developed during the HYDROCONTROLLER project is also considered useful in the IMPRESAR & S4.0 project and in particular in the final part of the project in relation to the evaluation of the benefits and the results obtained in order to evaluate any deviation from the planned strategies. Main results obtained during project is to evaluate the effectiveness of the propose architecture (Figure 3.4). As part of the IMPRESAR & S4.0 research project, the purpose of the activities in which I involve is to complete the big-data collection and management system through the different tools and devices used in the research and development projects.

Chapter 8

Conclusion and Future Work

This Chapter summarizes the main contribution of this dissertation and enlist the possible future research work.

The application of IoT devices in our daily life is increasing rapidly. They are playing a key role in industrial, transportation, energy, agriculture, etc. Their need became more critical in the last two years because of the Covid-19 pandemic where human interaction has become very limited. This rapid need also brings different types of challenges. These challenges are required to be addressed in a way to fulfill the market demand. Because of their tiny physical structure requirement and low price demand by the market (customer), vendors are designing them in a way to just fulfill the basic functional requirements. Due to their limited resources (low memory, low power, low processing, etc) vendors are not investing money on their security. Security experts state that we are at a crisis point with regard to the security of embedded systems, including IoT devices [136]. Because manufacturers are creating IoT chips with low features and on the other hand end-user buy the chip and have less or no information that when and how to update it (patch). That's why IoT devices are vulnerable to attack. Besides this problem, network administrators also face other IoT challenges such as QoS, reliability, availability, scalability, connectivity, interoperability, mobility, etc.

In this dissertation, I worked to improve the security in terms of availability with the application of the Fog layer. In the Chapter 3, I explained the three basic standards designed for constrained devices. One is developed by the IEEE, known as IEEE 802.15.4. The remaining two (RPL and 6LoWPAN-ND) are designed by the IETF. All these standard protocols are based on tree topology, where the root or head node is responsible for the configuration and management. Technically this root node can run all three protocols but the problem is that if any malfunctioning, technical failure, or security attack to this important node happens then the whole network will become compromised. I proposed an architecture where we can decouple the root node functionalities with respect to the standard, and place them at the Fog layer. At the Fog layer ISPs can get befits from emerging technologies such as NFV, where they can run the PAN coordinator which act as root node in the IEEE 802.15.4 protocol [67], 6LBR as root node in 6LoWPAN-ND [143], and VDR in RPL as a root node [158]. Keeping in mind this problem, my architecture also provides significant scalability and increased availability. The root nodes in all these three protocols installed at the Fog layer allow the network administrators to configure and manage any time anywhere. The architecture shifted the complexity of root nodes from the gateway position to Fog and leave EPs just as forwarding nodes. The Fog layer also gives the opportunity to fulfill the network and security management. My proposed architecture enables the concept of availability as discussed in the Chapter 3. The simulation results fully confirm the validity of the approach and by choosing this architecture I can further do optimization parameters with EP number and position, and a further security measure, e.g., by using “sleeping” EPs which are activated when an attack is detected. I presented EPs as the first-hop set of nodes connected to the LLN root functions, and for this, I proposed an 802.15.4 virtual interface and considered a reliable and secure link between root function and EP. This link can be a direct Ethernet or an encrypted virtual link such VPN tunnel. During this research work, I found some interesting future research topics and they are still open points that should be studied to further improve the efficiency and resilience of IoT systems. For example, the RPL standard [158] has not described the detailed functionality of the VDR. There is a need to define its proper functionalities so that it can harmonize all attached DODAG roots and their designated LLN. Second, the RPL protocol convergence time is high due to the inherent features of the distance vector and source routing (non-storing mode); in contrast, the single DODAG root has the ability to manage thousands of constrained nodes. However, the theoretical limit on a DODAG root has not been defined yet. Similarly, there is no limit on VDR to maintain DODAG roots. Thus, there are imperative limits necessary on both vital devices to obtain the optimal performance of the

RPL protocol. The third is about Mobility, The 6LoWPAN-ND [143] does not support mobility for inter LoWPAN en route over networks. Although standards [141, 143] support the mobility requirements for a device moving from one LLN to the next, proper network management is required among 6BBRs for the registration of lifetimes against acquired addresses. Fourth is regarding the anycast addresses because their utilization in 6LoWPAN-ND and RPL is challenging and requires the synchronization between DODAG roots and 6BBRs. Lastly, traffic management: the primary 6BBR is responsible for all services at a given time and can become a bottleneck for the entire 6LoWPAN, because secondary 6BBRs work as backup support in the case of primary 6BBR failure [141]. Thus, there is a need for a proper traffic management mechanism for all 6BBRs.

Furthermore, I did also work on the implementation of RFC8505 "*Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery*" in the ns-3 (`www.nsnam.org`) network simulator in order to explore and verify the reliability and robustness of the LLNs. As a matter of fact, most LLNs operating systems do not implement RFC8505 or they do implement its earlier version (RFC6775). As discussed in the Chapter 4, the results of the work did show how the current systems are subject to several issues that would be fixed by proper implementation of the latest standards, both from a security and reliability point of view. Moreover, the results (that will be submitted for publication in the next future) shows that, contrary to common beliefs, RFC8505 is mandatory for LLNs.

The proposed architecture in the Chapter 3 was designed by considering the problems and limitations of the IoT devices and their respective protocols. By considering the same architecture, I worked at the application level where I provided ease to the IoT devices to access the Virtual Functions (VF) as per their demand with the help of Federated Learning (FL). In addition to this, I also provided a solution for Service Providers (SPs) to increase their revenue during handling the Service Requests (SRs) from end-users (EUs) with the provision of proper QoS. The main contributions of my work are FL application to forecast the network VFs demand and with consideration of the user's privacy. Second, contextualization of SRs with different priorities and formulation to solve the SP maximum revenue problem. Third, I proposed the proper VFs placement strategy and a suitable matching-based SRs allocation algorithm based on the FL and the previously provided VFs forecasting scheme. Finally, I did the performance evaluation of the proposed scheme with a centralized Chaos Theory (CT) based prediction scheme.

The above-mentioned topics did bring up the problem of measuring with sufficient precision and timeliness the performance of the network, both at the network level and application level. Toward this end, it is possible to use RFC8250 "*IPv6 Performance and Diagnostic Metrics (PDM) Destination Option*" [45]. However, RFC8250 can not be deployed as it is outside of strictly controlled environments due to the lack of security options. In other terms, using it would lead to both security issues (i.e., possible attacks on the software architecture) and possible tampering with the sensed data.

Actual systems either have limited granularity (e.g., IPFIX) or suffer from a lack of privacy and security (e.g., PDM). However, the services offered through IoT systems, much like any system on the Internet, must not only be studied and improved but must also be continuously monitored. Toward this end, despite its limitations, PDM represents the best choice considering its granularity and low overhead.

The PDM problem is that performance assessment data is provided in clear-text, so malicious actors may be able to gather information for future assaults. In order to overcome these limitations, I had the opportunity to collaborate on the definition of the security of PDMv2. The standard proposal, which is still being worked on, uses a lightweight handshake (registration procedure) and encryption to safeguard data. It also includes a list of additional performance measures that might be useful for further performance evaluation. My proposal used the IRTF HPKE framework [23] standard to provide confidentiality and integrity of PDM data. The PDMv2 aims at fixing the current approaches limitations and providing a secure and low-complexity solution to provide performance measures of networked applications. This is a fundamental step toward more robust and intelligent network solutions. I am participating in the standardization of the new version of PDM under the IETF umbrella and the first two drafts of the new standard (PDMv2) have been published successfully.

My future research plan can be divided into three main branches: 1) contributions to the standardization groups, 2) further exploration of the research topics, and 3) technological transfer. The first point will mainly be focused on the continuation of the contribution toward the standardization of the PDMv2 RFC. This will require further analysis of the protocol security, and the definition of proper authentication and option negotiation methods. The standardization is currently ongoing, with weekly meetings with the standardization group. Moreover, I do plan to submit my findings with respect to RFC 8505 to a journal, and to the attention of the 6lo IETF Working Group, with which I am already in contact. Furthermore, I plan to discuss with the authors of the 6LoWPAN-ND about how to deploy 6LBR in a virtualized environment. With regards to the scientific research topics, I plan to consider the use of SDN and NFV in the Fog layer, there are some open research points that are worth exploring, e.g., how to optimize the placement and load of virtual functions in nodes that, although more powerful than sensor devices, are not usually implemented in a data-center-like structure. Therefore, even if we can consider the Fog layer as more secure and reliable than the WSN, the Fog itself can be optimized for performance, security, and reliability in order to ensure the system goals (and to avoid over-provisioning). Last but not least, I plan to get in contact with the LLNs operating systems developers to foster the adoption of the relevant standards, in particular RFC 8505, and to leverage the connections with the industries to promote the early adoption of PDMv2. This last point is already ongoing at the international level (in the standardization there are also partners coming from industries), but I think that it would be beneficial to foster the standard adoption also at a national level. Hence, I plan to disseminate the ongoing work by leveraging the connections with the national industrial associations (e.g., Italian Association of Electrotechnics, Electronics, Automation, Information Technology and Telecommunications (AEIT).

Appendix A

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.

International Journals

1. **Adnan Rashid**, Tommaso Pecorella, and Francesco Chiti, . "Toward Resilient Wireless Sensor Networks: A Virtualized Perspective", *Sensors*, vol. in press, 2020. (Special Issue: Energy-Efficient Resource Allocation for beyond 5G and IoT Systems) [DOI:10.3390/s20143902]

Accepted

1. Benedetta Picano, Romano Fantacci, Tommaso Pecorella, and **Adnan Rashid**. "Federated learning for IoE environments: A service provider revenue maximization framework". In: ITU Journal 2.14 (2021), p. 3902.

In Progress

1. **Adnan Rashid**, Tommaso Pecorella, and Romano Fantacci. "Is 6LoWPAN-ND necessary for LLNs? (Spoiler alert: yes)".

International Standards

1. Nalini Elkins, M. Ackermann, Ameya Deshpande, Tommaso Pecorella, and **Adnan Rashid**. Encrypted IPv6 Performance and Diagnostic Metrics Version 2(EPDMv2) Destination Option. Internet-Draft draft-elkins-ippm-encrypted-pdmv2-00. Work in Progress. Internet Engineering Task Force,

June 2021. 16 pp. URL: https://datatracker.ietf.org/doc/html/draft-elkins-ippm-encrypted-pdmv2-00.

2. Nalini Elkins, M. Ackermann, Ameya Deshpande, Tommaso Pecorella, and **Adnan Rashid**, “IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option,” Internet Engineering Task Force, Internet-Draft draft-elkins-ippm-encrypted-pdmv2-01, Oct. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-elkins-ippm-encrypted-pdmv2-01

Open-source Projects

1. https://gitlab.com/tommypec/ns-3-dev/-/tree/6LoWPAN-ND-redo/src

Bibliography

[1] Contiki-ng. [Online]. Available: https://github.com/contiki-os/contiki/blob/master/doc/sicslowpan-doc.txt

[2] Freertos. [Online]. Available: https://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_TCP/IPv6/index.html?_ga=2.171110816.1439955271.1609622036-1945043171.1609620665

[3] Mbed-os. [Online]. Available: https://os.mbed.com/docs/mbed-os/v6.6/introduction/index.html

[4] Netsim. [Online]. Available: https://www.tetcos.com/index.html

[5] ns-3. [Online]. Available: http://www.nsnam.org

[6] omnetpp++. [Online]. Available: https://doc.omnetpp.org/inet/api-current/neddoc/index.html

[7] Openthread. [Online]. Available: https://openthread.io/

[8] Openwsn. [Online]. Available: http://openwsn-berkeley.github.io/firmware/group___lo_w_p_a_n.htmlhttps://openwsn.atlassian.net/wiki/spaces/OW/pages/688149/6LoWPAN

[9] Qualnet. [Online]. Available: https://www.scalable-networks.com/

[10] Riot os. [Online]. Available: https://doc.riot-os.org/group__net__sixlowpan__nd.html

[11] Tiny os. [Online]. Available: https://github.com/tp-freeforall/prod/blob/tp-master/tools/tinyos/c/blip/lib6lowpan/6lowpan.hhttps://github.com/tinyos/tinyos-main/blob/master/doc/txt/tep136.txt

[12] Zephyr-os. [Online]. Available: https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/zephyr/guides/networking/net-stack-architecture.html?highlight=rfc%206282

[13] A. Ahmad and L. Atzori, "Mno-ott collaborative video streaming in 5g: The zero-rated qoe approach for quality and resource management," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 361–374, 2020.

[14] A. Ahmad, A. Floris, and L. Atzori, “Qoe-centric service delivery: A collaborative approach among otts and isps,” *Computer Networks*, vol. 110, pp. 168–179, 2016.

[15] B. R. Al-Kaseem, Y. Al-Dunainawi, and H. S. Al-Raweshidy, “End-to-end delay enhancement in 6lowpan testbed using programmable network concepts,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3070–3086, April 2019.

[16] R. AL MOGBIL, M. AL ASQAH, and S. EL KHEDIRI, “Iot: Security challenges and issues of smart homes/cities,” in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 2020, pp. 1–6.

[17] G. Almes, S. Kalidindi, and M. Zekauskas, “A Round-trip Delay Metric for IPPM,” RFC 2681 (Proposed Standard), RFC Editor, Fremont, CA, USA, Sep. 1999. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2681.txt

[18] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[19] H. Alvestrand, “A Mission Statement for the IETF,” RFC 3935 (Best Current Practice), RFC Editor, Fremont, CA, USA, Oct. 2004. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3935.txt

[20] S. Athmaja, M. Hanumanthappa, and V. Kavitha, “A survey of machine learning algorithms for big data analytics,” in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 3 2017, pp. 1–4.

[21] B. Bajic, A. Rikalovic, N. Suzic, and V. Piuri, “Industry 4.0 implementation challenges and opportunities: A managerial perspective,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 546–559, 2021.

[22] R. Barnes, B. Beurdouche, J. Millican, E. Omara, K. Cohn-Gordon, and R. Robert, “The Messaging Layer Security (MLS) Protocol,” Internet Engineering Task Force, Internet-Draft draft-ietf-mls-protocol-11, Dec. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-mls-protocol-11

[23] R. Barnes, K. Bhargavan, B. Lipp, and C. A. Wood, “Hybrid Public Key Encryption,” Internet Engineering Task Force, Internet-Draft draft-irtf-cfrg-hpke-12, Sep. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hpke-12

[24] S. Bayat, Y. Li, L. Song, and Z. Han, “Matching theory: Applications in wireless communications,” *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 103–122, 11 2016.

[25] M. Benammar, A. Abdaoui, S. H. Ahmad, F. Touati, and A. Kadri, "A modular iot platform for real-time indoor air quality monitoring," *Sensors*, vol. 18, no. 2, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/2/581

[26] Bingqing Luo, Suning Tang, and Zhixin Sun, "Research of neighbor discovery for ipv6 over low-power wireless personal area networks," in *2015 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, Aug 2015, pp. 233–238.

[27] C. Bormann, "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," RFC 7400 (Proposed Standard), RFC Editor, Fremont, CA, USA, Nov. 2014. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7400.txt

[28] U. Brandes and C. Pich, "Centrality estimation in large networks," in *INTL. JOURNAL OF BIFURCATION AND CHAOS, SPECIAL ISSUE ON COMPLEX NETWORKS' STRUCTURE AND DYNAMICS*, 2007.

[29] R. Callon, "The Twelve Networking Truths," RFC 1925 (Informational), RFC Editor, Fremont, CA, USA, Apr. 1996. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1925.txt

[30] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An iot-aware architecture for smart healthcare systems," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 515–526, Dec 2015.

[31] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, "An architecture to support the collection of big data in the internet of things," in *2014 IEEE World Congress on Services*, June 2014, pp. 442–449.

[32] S. Chakrabarti, G. Montenegro, R. Droms, and J. Woodyatt, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) ESC Dispatch Code Points and Guidelines," RFC 8066 (Proposed Standard), RFC Editor, Fremont, CA, USA, Feb. 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8066.txt

[33] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, *Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era*, 6 2018, vol. 25, no. 3.

[34] H. Chaouchi, *Introduction to the Internet of Things*. John Wiley & Sons, Ltd, 2013, ch. 1, pp. 1–33. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118600146.ch1

[35] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," vol. 3, no. 6, 12 2016, pp. 854–864.

[36] F. Chiti, R. Fantacci, and B. Picano, “A matching game for tasks offloading in integrated edge-fog computing systems,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3718, 2020, e3718 ett.3718. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3718

[37] Y. Choi, Y.-G. Hong, J.-S. Youn, D. Kim, and J. Choi, “Transmission of IPv6 Packets over Near Field Communication,” Internet Engineering Task Force, Internet-Draft draft-ietf-6lo-nfc-17, Aug. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-nfc-17

[38] A. Conta, S. Deering, and M. Gupta (Ed.), “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification,” RFC 4443 (Internet Standard), RFC Editor, Fremont, CA, USA, Mar. 2006, updated by RFC 4884. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4443.txt

[39] F. Conti, S. Colonnese, F. Cuomo, L. Chiaraviglio, and I. Rubin, “Quality of experience meets operators revenue: Dash aware management for mobile streaming,” in *2019 8th European Workshop on Visual Information Processing (EUVIP)*, 2019, pp. 64–69.

[40] P. Corcoran and S. K. Datta, “Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network,” *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, 10 2016.

[41] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460 (Draft Standard), RFC Editor, Fremont, CA, USA, Dec. 1998, obsoleted by RFC 8200, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2460.txt

[42] ——, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 8200 (Internet Standard), RFC Editor, Fremont, CA, USA, Jul. 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8200.txt

[43] L. Deru, S. Dawans, M. Ocaña, B. Quoitin, and O. Bonaventure, “Redundant border routers for mission-critical 6lowpan networks,” in *Real-world wireless sensor networks*. Springer, 2014, pp. 195–203.

[44] M. J. Dworkin, *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac*. National Institute of Standards & Technology, 2007.

[45] N. Elkins, R. Hamilton, and M. Ackermann, “IPv6 Performance and Diagnostic Metrics (PDM) Destination Option,” RFC 8250 (Proposed

Standard), RFC Editor, Fremont, CA, USA, Sep. 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8250.txt

[46] N. Elkins, M. Ackermann, A. Deshpande, T. Pecorella, and A. Rashid, "IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option," Internet Engineering Task Force, Internet-Draft draft-elkins-ippm-encrypted-pdmv2-01, Oct. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-elkins-ippm-encrypted-pdmv2-01

[47] R. Fantacci and B. Picano, "A matching game with discard policy for virtual machines placement in hybrid cloud-edge architecture for industrial iot systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7046–7055, 2020.

[48] ——, "When network slicing meets prospect theory: A service provider revenue maximization framework," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3179–3189, 2020.

[49] A. Floris, A. Ahmad, and L. Atzori, "Qoe-aware ott-isp collaboration in service management: Architecture and approaches," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 2s, Apr. 2018.

[50] A. A. Fröhlich, R. M. Scheffel, D. Kozhaya, and P. E. Veríssimo, "Byzantine resilient protocol for the iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2506–2517, April 2019.

[51] I. Froiz-Míguez, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "Design, implementation and practical evaluation of an iot home automation system for fog computing applications based on mqtt and zigbee-wifi sensor nodes," *Sensors*, vol. 18, no. 8, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/8/2660

[52] P. Gallagher, "Digital signature standard (dss)," *Federal Information Processing Standards Publications, volume FIPS*, vol. 186, 2013.

[53] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 513–521.

[54] T. N. Gia, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fault tolerant and scalable iot-based architecture for health monitoring," in *2015 IEEE Sensors Applications Symposium (SAS)*, April 2015, pp. 1–6.

[55] C. Gomez, S. M. Darroudi, T. Savolainen, and M. Spoerk, "IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP," Internet Engineering Task Force, Internet-Draft draft-ietf-6lo-blemesh-10, Apr. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-blemesh-10

[56] C. Gonzalez, S. M. Charfadine, O. Flauzac, and F. Nolot, "Sdn-based security framework for the iot in distributed grid," in *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, July 2016, pp. 1–5.

[57] N. Gupta, P. K. Juneja, S. Sharma, and U. Garg, "Future aspect of 5g-iot architecture in smart healthcare system," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 406–411.

[58] M. Ha, K. Kwon, D. Kim, and P. Kong, "Dynamic and distributed load balancing scheme in multi-gateway based 6lowpan," in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, Sep. 2014, pp. 87–94.

[59] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, 12 2015. [Online]. Available: http://doi.acm.org/10.1145/2827872

[60] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016, pp. 3928–3937.

[61] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291 (Draft Standard), RFC Editor, Fremont, CA, USA, Feb. 2006, updated by RFCs 5952, 6052, 7136, 7346, 7371, 8064. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4291.txt

[62] Y.-G. Hong, C. Gomez, A. R. Sangi, and S. Chakrabarti, "IPv6 over Constrained Node Networks (6lo) Applicability & Use cases," Internet Engineering Task Force, Internet-Draft draft-ietf-6lo-use-cases-11, Jul. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-use-cases-11

[63] J. Hou, B. R. Liu, Y.-G. Hong, X. Tang, and C. E. Perkins, "Transmission of IPv6 Packets over PLC Networks," Internet Engineering Task Force, Internet-Draft draft-ietf-6lo-plc-06, Apr. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-plc-06

[64] J. Hui (Ed.) and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282 (Proposed Standard), RFC Editor, Fremont, CA, USA, Sep. 2011, updated by RFC 8066. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6282.txt

[65] IEEE, "Ieee standard for telecommunications and information exchange between systems - lan/man specific requirements - part 15: Wireless medium

access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpan)," *IEEE Std 802.15.4-2003*, pp. 1–680, 2003.

[66] ——, "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, 2006.

[67] ——, "Ieee standard for low-rate wireless networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, 2016.

[68] ——, "Ieee standard for low-rate wireless networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.

[69] Y. Jiao, P. Wang, D. Niyato, M. Abu Alsheikh, and S. Feng, "Profit maximization auction and data management in big data markets," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 3 2017, pp. 1–6.

[70] K. Kalkan and S. Zeadally, "Securing internet of things with software defined networking," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 186–192, Sep. 2018.

[71] P. O. Kamgueu, E. Nataf, and T. Djotio, "Architecture for an efficient integration of wireless sensor networks to the internet through internet of things gateways," *International Journal of Distributed Sensor Networks*, vol. 13, no. 11, p. 1550147717744735, 2017.

[72] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge: Cambridge University Press, 2003.

[73] M. R. M. Kassim, "Iot applications in smart agriculture: Issues and challenges," in *2020 IEEE Conference on Open Systems (ICOS)*, 2020, pp. 19–24.

[74] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303 (Proposed Standard), RFC Editor, Fremont, CA, USA, Dec. 2005. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4303.txt

[75] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401 (Proposed Standard), RFC Editor, Fremont, CA, USA, Nov. 1998, obsoleted by RFC 4301, updated by RFC 3168. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2401.txt

[76] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: early architecture and research perspectives," *IEEE Network*, vol. 29, no. 3, pp. 104–112, May 2015.

[77] H. Kim, J. Park, M. Bennis, and S. Kim, “Blockchained on-device federated learning,” 2019, pp. 1–1.

[78] M. Kirsche and J. Hartwig, “A 6lowpan model for omnet++: Poster abstract,” in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, ser. SimuTools ’13. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, p. 333.

[79] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, “A survey of machine learning techniques applied to self-organizing cellular networks,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

[80] H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” RFC 2104 (Informational), RFC Editor, Fremont, CA, USA, Feb. 1997, updated by RFC 6151. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2104.txt

[81] H. Krawczyk and P. Eronen, “HMAC-based Extract-and-Expand Key Derivation Function (HKDF),” RFC 5869 (Informational), RFC Editor, Fremont, CA, USA, May 2010. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5869.txt

[82] T. Kudo and T. Ohtsuki, “Cell selection using distributed q-learning in heterogeneous networks,” in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 10 2013, pp. 1–6.

[83] Y. Kuo, C. Li, J. Jhang, and S. Lin, “Design of a wireless sensor network-based iot platform for wide area and heterogeneous applications,” *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5187–5197, June 2018.

[84] N. Kushalnagar, G. Montenegro, and C. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” RFC 4919 (Informational), RFC Editor, Fremont, CA, USA, Aug. 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4919.txt

[85] A. Langley, M. Hamburg, and S. Turner, “Elliptic Curves for Security,” RFC 7748 (Informational), RFC Editor, Fremont, CA, USA, Jan. 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7748.txt

[86] H. Lee, S. D. Min, M.-H. Choi, and D. Lee, “Multi-agent system for fault tolerance in wireless sensor networks.” *Ksii Transactions on Internet & Information Systems*, vol. 10, no. 3, 2016.

[87] L. Li, K. Ota, and M. Dong, “Human in the loop: Distributed deep model for mobile crowdsensing,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4957–4964, 12 2018.

[88] M. Li, Y. Sun, H. Huang, J. Yuan, Y. Du, Y. Bao, and Y. Luo, "Profit maximization resource allocation in cloud computing with performance guarantee," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, 12 2017, pp. 1–2.

[89] S. Li, J. Huang, and S. R. Li, "Dynamic profit maximization of cognitive mobile virtual network operator," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 526–540, 3 2014.

[90] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-driven content caching," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 4 2016, pp. 1–9.

[91] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," vol. 12, no. 3, 9 2018, pp. 2495–2508.

[92] Z. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, 2 2010.

[93] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Healing on the cloud: Secure cloud architecture for medical wireless sensor networks," *Future Generation Computer Systems*, vol. 55, pp. 266 – 277, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X15000266

[94] B. Luo and Z. Sun, "Enabling end-to-end communication between wireless sensor networks and the internet based on 6lowpan," *Chinese Journal of Electronics*, vol. 24, no. 3, pp. 633–638, 2015.

[95] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," pp. 1–6, 5 2018.

[96] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[97] D. McGrew, "An Interface and Algorithms for Authenticated Encryption," RFC 5116 (Proposed Standard), RFC Editor, Fremont, CA, USA, Jan. 2008. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5116.txt

[98] H. B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas, "Federated learning of deep networks using model averaging," vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[99] ——, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research,

vol. 54. Fort Lauderdale, FL, USA: PMLR, 4 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[100] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2017. [Online]. Available: https://arxiv.org/abs/1602.05629

[101] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

[102] I. Miladinovic and S. Schefer-Wenzl, "Nfv enabled iot architecture for an operating room environment," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb 2018, pp. 98–102.

[103] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[104] B. Molina, C. E. Palau, G. Fortino, A. Guerrieri, and C. Savaglio, "Empowering smart cities through interoperable sensor network enablers," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2014, pp. 7–12.

[105] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944 (Proposed Standard), RFC Editor, Fremont, CA, USA, Sep. 2007, updated by RFCs 6282, 6775, 8025, 8066. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4944.txt

[106] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, M. Richardson, S. Jiang, T. Lemon, and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 8415 (Proposed Standard), RFC Editor, Fremont, CA, USA, Nov. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8415.txt

[107] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. PP, 06 2016.

[108] E. Municio, S. Latre, and J. Marquez-Barja, "Extending network programmability to the things overlay using distributed industrial iot protocols," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[109] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Standard), RFC Editor, Fremont, CA, USA, Sep. 2007, updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4861.txt

[110] B. Negash, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Lisa: Lightweight internet of things service bus architecture," *Procedia Computer Science*, vol. 52, pp. 436 – 443, 2015, the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915008108

[111] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC 8439 (Informational), RFC Editor, Fremont, CA, USA, Jun. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8439.txt

[112] F. Nizzi, T. Pecorella, F. Esposito, L. Pierucci, and R. Fantacci, "Iot security via address shuffling: the easy way," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3764–3774, Apr 2019.

[113] J. Postel, "Internet Control Message Protocol," RFC 792 (Internet Standard), RFC Editor, Fremont, CA, USA, Sep. 1981, updated by RFCs 950, 4884, 6633, 6918. [Online]. Available: https://www.rfc-editor.org/rfc/rfc792.txt

[114] Ran Xua, S. Yang, Ping Li, and J. Cao, "Iot architecture design for 6lowpan enabled federated sensor network," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, June 2014, pp. 2997–3002.

[115] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," vol. 7, 2019, pp. 69 194–69 201.

[116] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446 (Proposed Standard), RFC Editor, Fremont, CA, USA, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8446.txt

[117] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347 (Proposed Standard), RFC Editor, Fremont, CA, USA, Jan. 2012, updated by RFCs 7507, 7905. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6347.txt

[118] E. Rescorla, H. Tschofenig, and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-dtls13-42, Apr. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls13-42

[119] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321 (Informational), RFC Editor, Fremont, CA, USA, Apr. 1992, updated by RFC 6151. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1321.txt

[120] A. E. Roth and M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis.* Cambridge University Press, UK, 1990.

[121] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[122] N. Saeed, R. Djechaiche, and R. A. Khalil, "5g-iot in the battle against covid-19: Prospects and challenges," *IEEE IoT Newsletter - July 2021*, 2021.

[123] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[124] D. Scazzoli, A. Mola, B. Silverajan, M. Magarini, and G. Verticale, "A redundant gateway prototype for wireless avionic sensor networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–7.

[125] S. Schneider, "The industrial internet of things (iiot) applications and taxonomy," *Internet of Things and Data Analytics Handbook*, pp. 41–81, 2017.

[126] B. Schneier, "The internet of things is wildly insecure-and often unpatchable," *Schneier on Security*, vol. 6, 2014.

[127] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[128] M. A. M. Seliem, K. M. F. Elsayed, and A. Khattab, "Performance evaluation and optimization of neighbor discovery implementation over contiki os," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 119–123.

[129] X. Shan, H. Zhi, P. Li, and Z. Han, "A survey on computation offloading for mobile edge computing information," in *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, 5 2018, pp. 248–251.

[130] Z. Shelby (Ed.), S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," RFC 6775 (Proposed Standard), RFC Editor, Fremont, CA, USA, Nov. 2012, updated by RFC 8505. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6775.txt

[131] N. Shukla and K. Fricklas, *Machine learning with TensorFlow.* Manning Shelter Island, Ny, 2018.

[132] A. K. Singh, M. Raj, and V. Sharma, "Architecture, issues and challenges in monitoring based on iot for smarter environment," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 142–146.

[133] D. Singh, G. Tripathi, and A. Jara, "Secure layers based architecture for internet of things," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 321–326.

[134] S. Singh, "Optimize cloud computations using edge computing," in *2017 International Conference on Big Data, IoT and Data Science (BID)*, 12 2017, pp. 49–53.

[135] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," vol. abs/1705.10467, 2017. [Online]. Available: http://arxiv.org/abs/1705.10467

[136] W. Stallings, *Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud.* Addison-Wesley Professional, 2015.

[137] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: Challenges, approaches, and open issues," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[138] P. Subramaniam and M. J. Kaur, "Review of security in mobile edge computing with deep learning," pp. 1–5, 3 2019.

[139] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, D. Rand and L.-S. Young, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381.

[140] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Standard), RFC Editor, Fremont, CA, USA, Sep. 2007, updated by RFC 7527. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4862.txt

[141] P. Thubert, C. E. Perkins, and E. Levy-Abegnoli, "IPv6 Backbone Router," RFC 8929, Nov. 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc8929.txt

[142] ——, "IPv6 Backbone Router," Internet Engineering Task Force, Internet-Draft draft-ietf-6lo-backbone-router-20, Mar. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-backbone-router-20

[143] P. Thubert (Ed.), E. Nordmark, S. Chakrabarti, and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery," RFC 8505 (Proposed Standard), RFC Editor, Fremont, CA, USA, Nov. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8505.txt

[144] H. Tianfield, "Towards edge-cloud computing," in *2018 IEEE International Conference on Big Data (Big Data)*, 12 2018, pp. 4883–4885.

[145] R. J. Tom, S. Sankaranarayanan, V. H. C. de Albuquerque, and J. J. P. C. Rodrigues, "Aggregator based rpl for an iot-fog based power distribution system with 6lowpan," *China Communications*, vol. 17, no. 1, pp. 104–117, Jan 2020.

[146] N. H. Tran, W. Bao, A. Zomaya, N. Minh N.H., and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 4 2019, pp. 1387–1395.

[147] H. Tschofenig (Ed.) and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things," RFC 7925 (Proposed Standard), RFC Editor, Fremont, CA, USA, Jul. 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7925.txt

[148] C. Tselios, I. Politis, M. Tsagkaropoulos, and T. Dagiuklas, "Valuing quality of experience: A brave new era of user satisfaction and revenue possibilities," in *2011 50th FITCE Congress - "ICT: Bridging an Ever Shifting Digital Divide"*, 2011.

[149] T. Tuor, S. Wang, T. Salonidis, B. J. Ko, and K. K. Leung, *Demo abstract: Distributed machine learning at resource-limited edge nodes*, 4 2018, pp. 1–2.

[150] L. Valerio, A. Passarella, and M. Conti, "Optimal trade-off between accuracy and network cost of distributed learning in mobile edge computing: An analytical approach," pp. 1–9, 6 2017.

[151] J. Vasseur (Ed.), M. Kim (Ed.), K. Pister, N. Dejean, and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," RFC 6551 (Proposed Standard), RFC Editor, Fremont, CA, USA, Mar. 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6551.txt

[152] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. S. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," 2018, pp. 63–71.

[153] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. S. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," *CoRR*, vol. abs/1506.05261, 2015. [Online]. Available: http://arxiv.org/abs/1506.05261

[154] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 63–71.

[155] ——, "Adaptive federated learning in resource constrained edge computing systems," 2019. [Online]. Available: http://arxiv.org/abs/1804.05271

[156] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 4 2019, pp. 2512–2520.

[157] T. Watteyne and P. Thubert, "Efficient 6lowpan neighbor discovery applied to multilink iot subnets," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 642–647.

[158] T. Winter (Ed.), P. Thubert (Ed.), A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), RFC Editor, Fremont, CA, USA, Mar. 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6550.txt

[159] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," 12 2018.

[160] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," vol. abs/1902.04885, 2019. [Online]. Available: http://arxiv.org/abs/1902.04885

[161] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 10 2017, pp. 1–6.

[162] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[163] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," vol. abs/1803.04311, 2018. [Online]. Available: http://arxiv.org/abs/1803.04311

[164] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," vol. 32, no. 4, 7 2018, pp. 54–60.