LATENCY BASED DEVICE FINGERPRINTING IN A LOW-POWER INDUSTRIAL WIRELESS SENSOR NETWORK

by

Carel Phillip Kruger

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

June 2021

SUMMARY

LATENCY BASED DEVICE FINGERPRINTING IN A LOW-POWER INDUSTRIAL WIRELESS SENSOR NETWORK

by

Carel Phillip Kruger

Supervisor(s):	Prof. G.P. Hancke
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Fingerprinting, Industrial Internet of Things, Latency, Security, Wireless Sensor Network.

Security is a key challenge for any IIoT network and more so for constrained IWSN deployments. Novel methods are thus required to enhance security, taking into consideration the lossy and low power nature of the IWSN. The use of ICMP packets is proposed as a method to generate fingerprinting information for IWSN devices. The ICMP based method uses the round-trip time information in the ICMP header as a fingerprinting metric. The results showed that the effect of the physical layer can be averaged out of the measurement if enough samples are available. A linear relationship was found between hop count and round-trip time for a static network which can be used in the design phase of the IWSN network or alternatively as a method to fingerprint routing anomalies in real-time. The ICMP method was able to differentiate between devices from different vendors, but unable to fingerprint devices from the same vendor due to physical layer interference. The work shows that fingerprinting in an IWSN using the ICMP method is possible if the timing delta under investigation is an order of magnitude larger than the timing variation introduced by the physical layer while maintaining a reasonable signal-to-noise ratio.

LIST OF ABBREVIATIONS

6LowPAN	IPv6 over Low-Power Wireless Personal Area Networks
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
COTS	Commercial off-the-shelf
EUI	Extended Unique Identifier
HHT	Hilbert-Huang Transform
HOS	Higher Order Statistics
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IIOT	Industrial Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
IQ	In-phase and Quadrature
IWSN	Industrial Wireless Sensor Network
KNN	K-Nearest Neighbors
LLN	Lossy Low-Power Network
LPM	Linear Programming Method
LRM	Linear Regression Method
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection

PLC	Programmable Logic Controller
PTP	Point-To-Point
PUF	Physically Unclonable Functions
QoS	Quality of Service
QPM	Quick Piecewise Minimum Algorithm
RFF	Radio Frequency Fingerprinting
RPL	Routing Protocol for Low-power and lossy networks
RSSI	Received Signal Strength Indicator
RTT	Round-Trip Time
SDR	Software Defined Radio
SM	Statistical Methods
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Security is a key challenge for any Industrial Internet of Things (IIoT) network [1], [2] and specifically for Industrial Wireless Sensor Networks (IWSN) [3] where the use of constrained devices and broadcast based wireless communications channels can be exploited to compromise the security of the network [4]–[6]. Novel methods are thus required to ensure protection against eavesdropping, identity theft, man in the middle, and denial of service attacks [7].

Fingerprinting is the method of extracting a unique identity from a device or network using one or more device features which are uniquely and repeatedly identifiable. Latency based device fingerprinting is proposed as a novel solution to enhance the security of IIoT deployments. The method researched in this thesis focuses on exploiting fingerprinting characteristics at device firmware level, specifically investigating the use of round-trip time (RTT) information obtained via ICMP ping requests. The majority of IIoT devices respond to ICMP ping requests without modification and is thus the ideal method to provide additional security for vendor specific IIoT systems where changes to proprietary firmware is impossible.

1.1.2 Research gap

Mainstream device fingerprinting research in mostly focused on hardware based signal manipulation and AI based algorithms to produce device fingerprints. These mainstream methods require modifications to the hardware and software stack as well as computationally expensive algorithms for successful fingerprinting. The typical IWSN consists of low-power, computationally constrained devices with many legacy deployments and are thus not well suited towards the new techniques.

Research is thus required to determine the feasibility of existing fingerprinting methods in constrained IWSN deployments and how such methods can be modified to be more suitable for low-power devices. The scope will be limited to fingerprinting methods and metrics where no modification of the existing device firmware or hardware is required. Using external radio frequency (RF) capturing devices in harsh environments or computationally expensive traffic analysis algorithms [8] is not practical for constrained IWSN devices.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The aim of the research is to identify fingerprinting criteria suitable for IWSN use. Suitable fingerprinting criteria should be evaluated for performance taking into account the lossy, low-power nature of the communications channel.

The following research questions will be used to guide the research:

- What criteria can be used to determine a successful fingerprint for an IWSN device?
- Which of the identified fingerprinting criteria can be evaluated using ICMP ping packets?
- How does the effect of the Lossy, Low-Power Network influence the determinism of the results?

1.3 APPROACH

To meet the research questions and objectives the following approach was followed:

- Conduct a literature study on related work.
- Determine fingerprinting criteria and evaluate feasibility for use in IWSN.
- Identify possible fingerprinting criteria which can be exploited using the ICMP method.
- Evaluate efficacy of the ICMP method through experimentation.
- Prepare findings for publication.

1.4 RESEARCH GOALS

The broader goal of the research is to improve the security and performance of new and legacy IWSN deployments. Understanding latency, determinism and the factors which influence latency and determinism are key requirements to successfully plan, implement and operate device fingerprinting algorithms in IWSNs.

1.5 RESEARCH CONTRIBUTION

A structured analysis and discussion of existing fingerprinting methods and the applicability of such methods to IWSNs is contributed as part of this thesis. The experimental results will provide concrete statistics on the feasibility of using the ICMP method for IWSN device fingerprinting as well as quantify the effect of the wireless channel on the measurement of latency in an IWSN.

1.6 RESEARCH OUTPUTS

The following publication has resulted from the research conducted within this study:

- C. P. Kruger and G. P. Hancke, "Enhanced security in Industrial internet of Things networks using latency based fingerprinting," in *Proceedings of the 18th IEEE International Conference on Industrial Informatics (INDIN)*, 2020, pp. 100–106

1.7 OVERVIEW OF STUDY

This thesis starts with a literature study in Chapter 2. The research methodology and test bed used to obtain the experimental results is discussed in Chapter 3. The experimental results for the three main types of experiments are given in Chapter 4. Results obtained in Chapter 4 are discussed in Chapter 5 and a conclusion and future work is given in Chapter 6.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

The aim of the literature study is to identify relevant methods for obtaining fingerprinting metrics for IoT devices. The sorted papers are presented in Section 2.2 and discussed from a device fingerprinting perspective using a modified OSI stack. The lowest level of the stack is called the sensors layer (Section 2.2.1) and consists of radio frequency sensors (Section 2.2.1.1) as well as MEMS sensors (Section 2.2.1.2). The second layer of the model is the network layer, discussed in Section 2.2.2 and the final layer is the software layer as in Section 2.2.3.

2.2 DEVICE FINGERPRINTING BASED ON THE OSI STACK

A structured method is required to evaluate sources of fingerprinting information and to better understand the interaction and dependencies between the variables of interest. A review of the related work has shown that fingerprinting is highly dependent on the layers used to implement the OSI stack [10], [11]. The OSI based method of sorting the relevant literature was selected due to the UDP-IP nature of the IWSN IoT stack [12] to be used for the experimental evaluation. The related work will thus be organized based on the OSI stack as shown in Figure 2.1.

The seven layers of the OSI model can be grouped into three layers when evaluating possible fingerprinting features. The hardware layer is the lowest layer in the stack and the most complicated layer from a timing and system clock perspective [13]. The hardware layer mostly consists of sensors [14] which measure signals of interest from the environment for information gathering and communication purposes. Fingerprinting features found in the sensor layer can further be categorized as location-dependent and location-independent features [5]. RSSI and channel frequency response are examples of features that change based on the location. Fingerprinting methods closer to the hardware layer are in general harder to manipulate and spoof when compared to fingerprinting metrics higher up

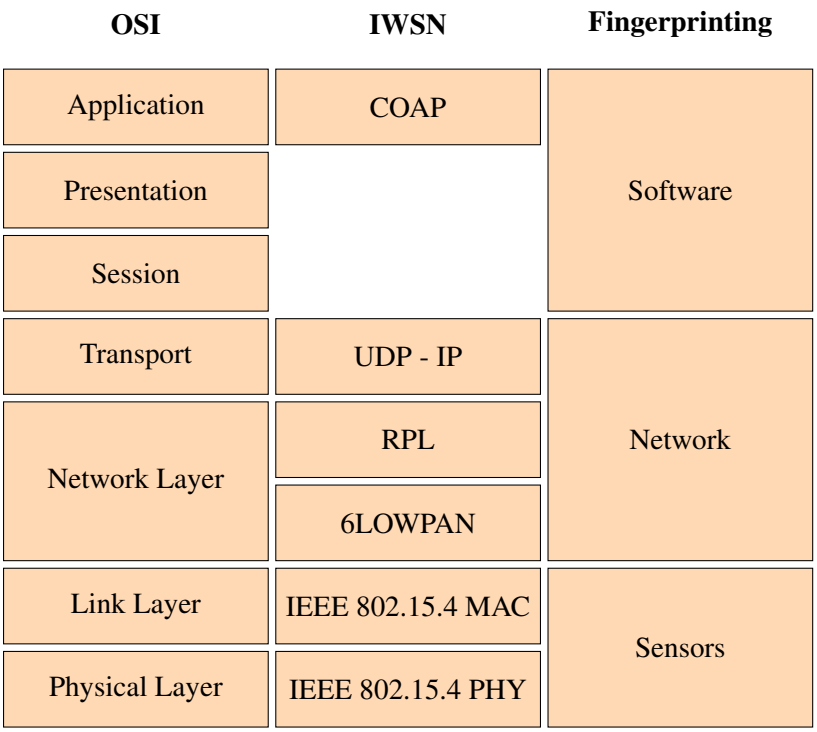

Figure 2.1. The OSI stack compared to the IIoT IWSN stack under investigation. Sources of fingerprinting features can be grouped into sensor, network and software based sources. Adapted from [9], © 2020 IEEE.

in the stack [15]. The second layer of possible fingerprinting features is the networking layer and can either be dependant on the hardware layer or produce fingerprinting results that are independent of the underlying hardware. The ICMP method is an example where the fingerprinting variable of interest, RTT is dependent on the deterministic behaviour of the hardware layer and the software specific implementation of the network stack. While traffic analysis based fingerprinting is an example of a software only fingerprint, independent of the hardware layer timing and only dependent on the application layer software implementation. The software layer is the final layer in the stack and is mostly independent from the preceding layers. Fingerprinting information is obtained by studying the behaviour of the software in question using either active or passive probing.

2.2.1 Sensor-based fingerprinting

The relevant citations identified over a four year period shows that the majority of research in the field of device fingerprinting is conducted at the sensor layer and focuses substantially on Radio Frequency Fingerprinting (RFF). The body of knowledge is mostly generated by identifying new fingerprinting metrics and applying new, known or derived classification methods. The features of

interest are digitised using one of several methods and classified using either Artificial Intelligence (AI) with Machine Learning (ML) or statistical methods (SM).

2.2.1.1 Radio frequency fingerprinting

The main motivation behind Radio frequency fingerprinting (RFF) is to find computationally inexpensive methods to provide devices with a unique identity. One possible advantage of RFF is the possibility to provide a computationally inexpensive method to secure IoT devices using the unique identity provided by the manufacturing tolerances in the radio. RFF can thus be a viable replacement for computationally expensive encryption algorithms when implementing resource constrained IWSN devices. Existing RFF based methods are however computationally intensive due to the use of machine learning for classification. One method of reducing the computational complexity is to replace machine learning methods with simpler and more computationally efficient protocols such as random forest for classification [16] or alternatively moving computationally complex algorithms to a gateway device.

RFF can be defined as the physical layer identification of radio devices by measuring the unique tolerances introduced by the physical layer component manufacturing process. Unique features are introduced by the manufacturing process of analog components in the transmission chain and can differentiate devices even if the manufacturer and model are identical. A high-end and low-end device [17], [18] can be seen as a basic measure of the extent to which the unique device tolerances are measurable. High-end devices make use of high quality commercial grade components where synchronized clocks and accurate, low tolerance components significantly reduces the ability to detect the fingerprinting metrics in a noisy environment.

Low-end devices are however mass produced with an emphasis on low-cost and quantity with noticeable manufacturing tolerances and are hence easier to detect when compared to high-end devices. Examples of low-end devices include Zigbee radio modules, Bluetooth devices and low-cost software-defined radio (SDR) devices while high-end devices consist of commercial grade GSM base stations [19], mobile handsets and high end SDR radios. The quality of the fingerprinting metric and the difficulty to detect a metric will thus be determined by the capturing device used to produce and capture the signal. The literature has shown that mainly two diverging methodologies to implement device fingerprinting has emerged. The first is manual feature extraction where the fingerprinting metrics are selected manually based on prior knowledge of the signal. The second is where ML and

specifically convolutional neural networks (CNNs) with deep learning is used to automatically extract the fingerprinting metrics or features of interest.

Manual feature selection starts by obtaining a relevant time domain sample of a signal. Fingerprinting metrics are mostly obtained in two regions of the signal. The transient signal and steady-state signal [20] which can also be identified as the static and dynamic sections of the signal from a protocol perspective [21], [22]. The transient signal requires the transmitter to be cycled and additionally requires a very high sample rate to detect in contrast to steady-state signals which are easy to identify an analyse [23]. Once a region of the signal has been selected some prior information, such as signal type, modulation method and noise level is needed to select an extraction method [24]. Common methods for RFF extraction include transient signals in the signal envelope, modulation parameters, noise characteristics, spurious characteristics and carrier frequency characteristics [24]. An extracted signal can be further transformed using several techniques [25], [26]. The list of techniques include time-domain methods [23], [27], [28], short time Fourier Transform [29], Wigner-Ville Distribution and Hilbert-Huang Transform (HHT) [30]. Instantaneous amplitude, phase and frequency are the most widely used fingerprinting characteristics [31] once a signal of interest has been identified and can be compared to other signals of the same type using deviation, variance, skewness, and kurtosis if higher order statistics (HOS) is used to classify signals [32]. An example of an HOS method is where the covariance feature is extracted as an RFF and K-Nearest Neighbor (KNN) classifier is used for classification [33]. Extracting the feature of interest before classification avoids over-training and reduces training time due to not learning irrelevant features.

Automatic feature selection uses deep learning algorithms to extract features of interest directly from the signal source which overcomes the traditional limitations [34] of needing expert-defined fingerprinting metrics to successfully classify a feature of interest. The majority of automatic feature selection methods use the in-phase and quadrature components (IQ) of a signal as input to a suitable convolutional neural network (CNN). The CNN is responsible for the correct classification of the signal based on the training data provided. The lack of interpretation is however a standing criticism of automatic feature selection and one can argue that spoofable data such as MAC addresses and channel characteristics are being used for classification instead of the unique physical layer imperfections of the radio [35], [36]. Channel variations and correct classification thus remains a open problem and additional work is required to design CNNs invariant to the channel [37]. Another well known problem of CNNs is the inability to identify devices which do not belong to the original dataset. One strategy to

add a learning ability is to separate the feature extraction and classification process. One such example is [38] where a CNN with deep learning is used only for feature extraction while clustering is used for device identification. Splitting the identification and classification tasks avoids the issue of classifying a new device as one of the existing devices in the trained CNN and thus overcomes the issue of CNNs not being able to learn once trained [39].

2.2.1.2 MEMS sensors

MEMS sensor-based fingerprinting focuses on generating unique identification metrics from the physical layer sensing variations in micro-electromechanical system (MEMS) sensors and is an emerging and novel research focus area within device fingerprinting.

The concept of Physically Unclonable Functions (PUFs) [40], make use of variations in the manufacturing process to obtain a unique fingerprint for various types of MEMS circuits including clock oscillators and analogue-to-digital (ADC) converters found in pressure sensors, gyroscopes and accelerometers. Experimental results obtained from running the IoT-ID system [40] showed an 100 percent accuracy in repeatedly identifying individual commercial-off-the-shelf (COTS) devices in a 50 node deployment. The efficacy of exploiting variations in the manufacturing process can thus meet the uniqueness, robustness and repeatability requirements for fingerprinting metrics.

An acoustic based fingerprinting technique called MicPrint [14] exploits the PFU of embedded microphones to uniquely identify each device. A unique feature of MicPrint is the ability to be spoof-resistant since the acoustic PUF of the microphone is only accessible when the user blocks the microphone with a finger. A two factor authentication system is thus introduced to protect the PFU of the device from brute force or automated attacks. MicPrint uses a binary multi-layer perceptron (MLP) network for classification which is part of the feed forward artificial neural network (ANN) family of ML techniques.

A PUF can also be generated using the raw sensor data obtained from the motion sensors in a mobile device [41], [42]. Less than a 100 data points were required from the accelerometer and gyroscope to uniquely identify a device using the motion fingerprint PUF method. A multi-LSTM (Long Short Term Memory) neural network was used to classify the data and hence the low number of samples required to uniquely identify the device when compared to a conventional neural network. Similar results are obtained [43] where the calibration data is recovered from the raw measurements to create a

unique ID for a smart phone device.

Another novel PUF method [44] uses the physics associated with a specific sensor to implement the PUF function. The PUF is generated by measuring the time difference between actuating a device of interest and receiving a confirmation that the physical device has achieved the desired state of actuation. Timestamping and the deterministic behaviour of the network is thus critical to the success of the method.

The concept of a PUF is combined with blockchain technology [45] to create a registry of trust factors for each device. The trust registry is maintained by a gateway device which determines the trust factor based on the PUF of each wireless device it is able to detect. Blockchain is used to distribute the trust factors between gateway devices where constrained devices can query the trust database with minimal overhead.

The idea of a PUF is further extended by considering an entire cyber physical system (CPS) as a sensor with a unique PUF [46]. A data fusion approach is used to obtain a fingerprint for the sensors and process noise during the normal operation of the CPS. The difference in error between the running system and a derived model is used to detect malicious activity via unpredictable state changes. The CPS as a sensor is further developed [47] by introducing the concept of a hybrid fingerprint, which uses several features of a CPS to develop a unique identifier.

2.2.2 Network-based fingerprinting

Network based fingerprinting (Figure 2.1) makes use of fingerprinting metrics found inside data flows passing through the network stack. One of the first papers to exploit clock entropy as a fingerprinting method [48] used the TCP timestamps option as well as ICMP ping requests over a wired network to obtain a set of measurements for fingerprinting metrics. The majority of recent work uses machine learning to classify data once the fingerprinting metric is known in contrast to the simple differential time methods used by Kohno et al [48] in 2005. A review of the literature has also shown that fingerprinting metrics in the network layer are mostly obtained from either packet headers or by analysing the data inside the packets [11], with packet lengths and transmission rates being the two most distinctive traits of an IoT device [49].

The concept of a static and dynamic fingerprinting metrics is introduced by Bezawada et al [50]. A

static fingerprinting metric can be described as the protocols used by the device or the specific ports which the device responds to. The set of dynamic fingerprinting metrics can be defined as the data inside packets and how the data patterns change over consecutive transmissions. The idea aligns with selecting fingerprinting metrics from either the packet header or the data inside the packet. The idea of finding fingerprinting metrics inside a single packet can further be extended to analysing the data in a several packet sequence of transmissions, identifying the fingerprinting metric as a behavioural pattern. One such method is called inter-arrival-time and is a popular method to obtain a dynamic fingerprinting metric.

After fingerprinting metrics are obtained classification can be achieved using ML, statistical analysis or even state change analysis. The majority of ML methods used to classify fingerprinitng metrics on the network layer are either whitelist based fingerprinting or unsupervised learning [51]. Whitelist fingerprinting requires training data and can thus not classify data which is not part of the initial training set. Unsupervised learning can detect new devices without additional training and thus contributes to the popularity of k-nearest neighbors (KNN) based classification. The majority of papers in the literature review simply classifies the fingerprinting metrics of interest using several ML algorithms and then selects the algorithm with the best performance via a trial and error approach. A more analytical approach is to use genetic algorithms to compare and select the fingerprinting metrics with the most information content [52] in contrast to hand picking metrics and classifiers by trial and error.

Finding usable fingerprinting metrics for identifying devices from the same vendor is a recurring challenge especially in a corporate environment where the same operating system image is used company wide [53]. The problem of identifying devices from the same device type can be adressed by using TSMC-SVM [54]. The TSMC-SVM algorithm adds cosine similarity into the support vector machine (SVM) in an attempt to improve the identification accuracy of devices with identical hardware and software compositions. Packet-field, sequence-protocol and sequence-statistics features were used as fingerprinting metrics and obtained from network packet headers. The cosine method can also be used to implement automated device identification [55] comparing the header information from identified devices to the header information of a new packet to determine if a new device has been discovered.

In the same way the TSMC-SVM method uses packet headers to extract fingerprinting metrics the iFinger system [56] uses the data in network traffic to identify variables representing programmable

logic controller (PLC) state. The fingerprinting data is then used to build a state table of allowable states and when a state is detected outside of the allowable states in the state table, a malicious event is detected. The boolean logic in a industrial control packet can thus be seen as a form of information leakage. A passive fingerprintig attack using the 20-bit flow label field in the IPv6 protocol header [53] is another example where information leakage is used to generate fingerprinting metrics. The original idea behind the IPv6 flow label was to allow tracking of network flows for Quality of Service features (QoS). The well known tuple generation method for encoding the flow label was however vulnerable to reverse engineering and easily exploited. The vulnerability of using a device IP address as a fingerprinting metric should also not be underestimated. A flaw in IP address allocation is exploited [57] where the IP address itself is retained or reused for long periods of time. The authors conducted a study on 34,488 unique public IP addresses obtained from 2,230 users and found that IP address based fingerprinting remains a realistic method for determining the identity of an end-user. IP addresses are thus overlooked as a identity metric when compared to other techniques like cookies and traffic based fingerprinting.

Z-IoT [58] is a traffic based fingerprinting framework to identify ZigBee and Z-Wave based devices. The inter-arrival-time of consecutive packets was used to generate the device type signature and 300 bins used to digitize the data for ML evaluation. Experimental results showed device identification with average precision and recall of over 91 percent. The framework makes use of passive packet capture over a wireless interface, thus incurring no performance penalty in the network in contrast to ICMP based inter-arrival-time measurements, where obtaining and maintaining a database for fingerprinting features can easily overwhelm a network if too many scans are run at the same time. Discrete wavelet transformation can also be used on the inter-arrival delay of network packets to produce a unique pattern for identification [59]. The time difference between successive packets can also be captured on a routing device and analyzed using a CNN [60]. The Raspberry PI that was used to generate the device signatures was only able to maintain 8-10 device signatures and the recommendation is made to use cloud infrastructure to process the signatures.

A major advantage of network layer based fingerprinting is the ability to capture an analyse traffic at a central location with ample processing power and storage. Centralised packet capture on a gateway device resolves the issue of securing legacy industrial control processes due to the weak computing and storage capabilities of the industrial equipment [61]. When malicious activity is detected the compromised device can be immediately isolated from the rest of the network using the

central firewall or gateway device [52]. A prototype architecture for an automated fingerprinting and centralized packet capture system [15] incorporates both active and passive probing. Using both the uniqueness of a network device and the state of the information available from the OS to validate a nodes identity. The performance and scalability problems experienced by centralised fingerprinting is addressed by a distributed device fingerprinting technique (DEFT) [62]. A DEFT controller maintains a distributed classifiers database for fingerprinting, while gateways located at the edge implement the device classification and automatic device discovery.

2.2.3 Software-based fingerprinting

The software layer as shown in Figure 2.1 is a valuable source of hardware and network independent fingerprinting metrics. Many of these metrics are however contained in banner messages and user interfaces or returned as human readable text. An natural language processing (NLP) engine [63] is used to make sense of human readable fingerprinting metrics by automatically identifying the type, vendor, and product of IoT devices using banner data as input. The paper emphasises that useful fingerprinting information might not be hidden at all. Automatic collection of natural language based text is thus a viable method to obtain fingerprinting information where diagnostic information is transferred over the network in human readable format. NLP is further exploited to identify subtle differences between the file systems of various firmware images as a fingerprinting method [64]. Firmware images from 9,716 official websites were learned for identification and showed that thousands of devices on the internet are still using vulnerable firmware images.

The sheer number of possible metrics in the software layer is however a problem and intelligent frameworks are required to distinguish between useful metrics and those that are not. AndroPRINT [65] is a framework with the ability to automatically identify fingerprintable information in an Android device. The framework uses a collection, preparation and evaluation phase which is similar to the acquisition, sorting and learning phases used in most machine learning applications. The framework does however not use any statistical methods to determine the identity and simply relies on the unique information pattern created by querying various values from the Android API on each device in the study. The GATT profile of Bluetooth Low Energy (BLE) devices can also be used to create a device fingerprint that can be exploited to circumvent anti-tracking features of the BLE standard [66]. The GATT based method uses the same principle as in AndroPRINT [65] where information exposed by the standard feature set is collected and a fingerprint generated by combining mutually exclusive data. Exploiting Bluetooth at the software layer in contrast to using dedicated hardware to detect anomalies

in the RF domain and shows that multiple layers can be considered to obtain the same fingerprinting result. The results from [66] and [65] show that rigorous statistical methods are not always required if temporal change is not present in the data set under investigation.

CryptoFP [67] also makes use of API instructions, but adds statistical methods by measuring the execution time required for API instructions, allowing the correct identification of computers with different CPU load configurations. The research is an example of the original methodology followed by Kohno etal [48] adapted to device APIs, maintaining the idea that devices can be uniquely identified based on clock skew even at the software layer.

A cloud based clock skew authentication system [68] with multiple measurement servers is presented and evaluated. Clock skew is calculated using the Linear Programming Method (LPM), Linear Regression Method (LRM) and the Quick Piecewise Minimum (QPM) algorithm. Results showed that multiple servers when hosted in the same physical environment can be used to independently measure and produce device fingerprints. Skew values thus satisfy the relations of additive inverse and linearity when compared between servers. Clustering algorithms can be used to identify servers that are prone to share the same or similar fingerprints and to provide them with a new non-unique fingerprint [69]. The authors combined the clustering techniques with virtualization technology to generate web browsers with consistent fingerprints. A consistent fingerprint is indistinguishable and thus not prone to identification providing protection from API mining and clock skew based attacks.

2.3 CHAPTER SUMMARY

The reasoning behind using the modified OSI approach to organise the literature study was given in Section 2.2. The literature relevant to the sensor layer (Section 2.2.1) showed that the majority of the work in the sensor layer focuses on RFF (Section 2.2.1.1) which is not suitable for IWSN use due to the additional processing power and hardware modification required to implement such methods. The literature study also indicated that fingerprinting metrics based on MEMS sensors (Section 2.2.1.2) is an emerging focus area with potential for generating robust fingerprinting metrics. Network layer based fingerprinting methods was explored in Section 2.2.2 and found to be the most applicable to IWSN devices, due to the multitude of fingerprintable metrics found in the protocol information even when encrypted. Network layer fingerprinting metrics can easily be obtained through both active and passive methods without modification of the device under investigation and is thus ideal for legacy devices or systems where firmware modification is not possible. Section 2.2.3 discussed several methods in

the software layer and showed that finding relevant fingerprinting metrics can be a challenge due to the sheer number of fingerprintable metrics in the software layer. The efficacy of using information supplied by the device through an API to unknowingly identify itself was also demonstrated and should be considered as a plausible method to identify IWSN devices over the network.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

The methods chapter starts in Section 3.2 with a discussion on fingerprinting criteria obtained from the literature and adapted for IWSN use. Known evaluation criteria is required for possible fingerprinting metrics, based on prior work to avoid duplication. Known fingerprinting criteria adapted to the IWSN also provides insight into where fingerprintable metrics may be found once experimentation has started. The main methodology (Section 3.3) is centered around three experiments named after the network topology the experiments represent. The three experiments are designed to test the deterministic behaviour of the IWSN as well as the evaluation criteria given in Section 3.2. The chapter concludes with Section 3.4 where the test bed design, used to realise the proposed experiments is explained.

3.2 IWSN FINGERPRINTING CRITERIA

Any new method of device fingerprinting needs to be evaluated for performance in order to determine the efficacy of the method. Existing evaluation criteria [48], [70], [71] does however not take into account the lossy and low-power nature of an IWSN and thus needs to be modified accordingly for the IWSN context. The five features of device fingerprinting is thus expanded below based on the IWSN context as a basis for performance evaluation.

3.2.1 Universality

Every device in the IWSN must have the feature of interest and all the devices in the IWSN must be able to independently measure the feature of interest with determinism. Universality can also be extended to the IWSN gateway, which can perform complex measurements on behalf of a constrained device, negating the requirement for every device in the IWSN to measure the feature of interest.

3.2.2 Uniqueness

No two devices should have the same fingerprints especially where the same hardware is used from a single vendor. IEEE 802.15.4 EUI and MAC addresses uniquely identify devices, but can be modified

in software and are thus not trustworthy sources of identification.

3.2.3 Collectability

Is it possible to capture the features of interest with unmodified hardware and software or is external equipment or hardware modification required? The constrained nature of the IWSN makes collecting large amounts of data impractical due to the strain on battery life. Collectability is thus a function of the number of data points needed and the processing power required for obtaining a result. Many specialized and proprietary devices are deployed in IIoT networks and cannot be modified or replaced due to cost implications.

3.2.4 Robustness

The feature of interest should remain stable over time and the effect of external environmental aspects that directly influence the signal propagation, clock speed and processing latency should be evaluated. IWSN systems are deployed in harsh environments where temperature stability and RF performance is not guaranteed. Temperature stability and RF performance should thus be fully quantified before any conclusions are made on the robustness of a specific method.

3.2.5 Data-dependency

How much data is required to produce the fingerprint of the device and how fast can the device be fingerprinted once the fingerprinting database has been generated? IWSN devices are mostly constrained, battery powered devices and excessive processing or RF transmissions will dramatically shorten the lifespan of the device. A gateway device can be used to store and process fingerprinting data to reduce the impact of data dependency on the network performance.

3.3 METHODOLOGY

The use of ICMP ping packets to generate a fingerprint for IIoT devices using RTT delays was successfully demonstrated in [60], [72]–[75]. The feasibility of using these methods in an IWSN are however untested and needs to be explored further through experimentation. The robustness of the ICMP ping method is of specific interest as the Lossy Low-Power Network (LLN) of which the IWSN consists may introduce non-deterministic behaviour when applying fingerprinting at the transport layer.

The effect of the wireless channel in a point-to-point configuration is of specific interest since all the software layers above the hardware layer are dependent on the physical layer for timing accuracy and determinism. The multi-hop timing and the effect of multiple hops on a ping measurement will also be

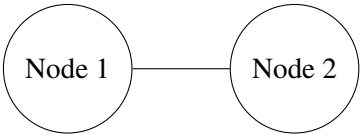

Figure 3.1. The point-to-point experiment layout. Nodes are placed at fixed distances with line-of-sight between nodes. Measurements are taken for each fixed distance.

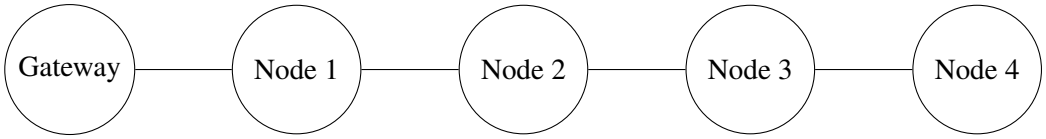

Figure 3.2. The mesh experiment layout. Nodes are placed at fixed 30 meter distances with line-of-sight between nodes. A measurement is taken from the gateway node to each of the nodes in the linear mesh network.

investigated since the majority of IWSN deployments use multi-hop mesh topologies to realize the network. The feasibility of device fingerprinting using the ICMP ping method will be evaluated in a star topology once the wireless channel and multi-hop performance is known.

3.3.1 Point-to-point

The aim of the point-to-point experiment (Figure 3.1) is to obtain RTT data where only the distance between two nodes are varied. Two nodes of the same type are placed at predetermined, linearly increasing distances from one another and a measurement taken. Each measurement consists of a ping6 ICMP probe with one thousand consecutive measurements. The predetermined intervals are increased and measurement process repeated until significant packet loss is detected on the specific measurement, indicating the experiment is finished. Care is taken to ensure line-of-sight RF conditions between the two devices and to remove objects which may cause RF interference. The experiment will be repeated twice once at minimum RF output power and once at maximum RF output power to assess the influence of the physical space on the experiment.

3.3.2 Mesh topology

The aim of the mesh topology experiment (Figure 3.2) is to obtain RTT data where multiple hops are introduced into the path taken by the ICMP packets. Five IWSN nodes of the same type are placed 30 meters apart in a straight line and configured to only have connectivity to the adjacent nodes. Data is obtained by taking one thousand consecutive ICMP measurements with the ping6 command from the first node to each of the nodes in the chain of devices. The experiment will be repeated twice once

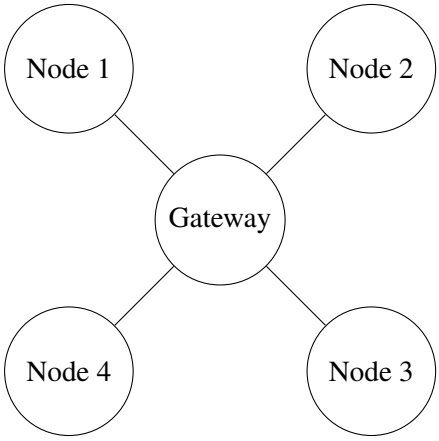

Figure 3.3. The star topology experiment. All the devices from different vendors are able to directly communicate with the gateway device as well as with each other.

on RF channel 26 and once on RF channel 18 to assess the influence of Wi-Fi interference on the experiment. Channel 26 is the only 802.15.4 channel which does not coincide with 802.11 and is thus immune to interference from Wi-Fi.

3.3.3 Star topology

The aim of the star topology experiment (Figure 3.3) is to test device fingerprinting characteristics if multiple devices from different vendors are part of the same IWSN network. A star topology network with a maximum distance of one meter between devices is created. All the devices from different vendors are able to directly communicate with the gateway device as well as with each other. Data is obtained by taking one thousand consecutive ICMP measurements with the ping6 command from the gateway node to each of the nodes in the star topology. The experiment is repeated two times. The first time with RF interference present and the second without any RF interference.

3.4 TEST BED DESIGN

A test bed is required to host experiments and provide a stable environment with measurable attributes in which to conduct experiments and collect data without bias. The test bed design will be explained in three major sections. The first is the hardware used to realize the experiments, the second the software or embedded device firmware and the third the environment in which the hardware and software will operate. The main purpose of the test bed is to provide an environment with deterministic RF noise levels. The effect of the RF noise has been identified as a potential influence on the determinism of the ICMP method and needs to be reproducible to maintain the integrity of the experiments.

3.4.1 Hardware

The effect of the physical layer and hardware on the universality and uniqueness of the ICMP method were tested by installing Riot-OS on three different devices and running ping probes in a multi-hop chain and star topology. Three development kits were used to obtain the experimental results as summarized in Table 3.1 below.

Table 3.1. Specifications of IoT devices used in experiments

Device	CPU	Speed(MHz)	FLASH(kB)	RAM(kB)
ATSAMR21G18A	ARM Cortex-M0+	48 MHz	256	32
nRF52840	ARM Cortex-M4	64MHz	1000	256
CC2538	ARM Cortex-M3	32 MHz	512	32

3.4.2 Software

Riot-OS is structured in a manner where the same operating system code is executed on each device. The hardware abstraction layer (HAL) is the only difference and thus makes the comparison between devices more realistic in contrast to devices with different operating systems. Possible flaws that may exist in software are equally influenced by all the experiments and thus does not bias the results in any way due to differences in stack implementation.

Latency measurements will be obtained using the ping6 command in Riot-OS. The command will be configured using the -c option to take one thousand measurements for each execution of the command. The large number of measurements allows for the examination of temporal changes in the latency that might coincide with temporary interference and signal loss. The implementation of the ping command varies from platform to platform and analyzing the procedure used to obtain the latency measurements is thus required. The measurement method used by the ping6 command in Riot-OS can be summarized as follows:

- The command is entered through the user interface with the address of interest and number of packets required.
- A unique timer component is initialized to maintain an independent microsecond timer.
- An ICMP packet with the ping payload identifier is allocated in memory.
- The current timer value is added to the ICMP packet.
- The stack sends ICMP packet to the network layer for processing.

- The network layer waits for a packet to return or times out if no reply is received.
- If a successful packet is returned the time payload is recovered.
- The received time payload is deducted from the current timer value and displayed.
- The process is repeated until the desired number of measurements is obtained.

The calculation of the RTT is thus highly dependent on the resolution of the timer which in this case is able to keep time accurately to microsecond level. The processing speed of the device itself does not influence the accuracy of the timer and only effects the speed at which the network stack processes packets. Riot-OS automatically prioritizes lower level processing and will thus provide minimal latency in the stack itself. ICMP packets are also not required to traverse the entire stack thus reducing the time required and possible sources of non-deterministic influences inside the network stack and operating system implementation.

3.4.3 Environmental factors

Radio frequency characteristics and temperature drift are the two main environmental factors which could influence the results of the experiment. RF noise from Wi-Fi devices needs to be taken into consideration for both a baseline and continuous operating point-of-view. The usable 802.15.4 physical layer frequencies overlaps with the assigned 802.11 or Wi-Fi frequencies and should be taken into consideration as an environmental factor. The effect of RF interference from Wi-Fi should be included in the study since many harsh environments already have Wi-Fi infrastructure which will practically interfere with the proposed methods in reality. The assumption is made that if sufficient received signal strength (RSSI) is present the effect of the RF propagation loss will be negligible. A shipping container will be used to create a Faraday cage to eliminate sources of external noise where applicable. Temperature drift will not be a problem in the test bed, but might effect the results for a real world deployment. The assumption is made that the OEM designer of the hardware used for testing will compensate for the changes in clock drift due to temperature through specific design for harsh environments. All experiments will be conducted at a room temperature of 20 degrees Celsius.

3.5 CHAPTER SUMMARY

Fingerprinting criteria adapted to the IWSN was given in Section 3.2 and used as the basis to identify the ICMP method as a plausible fingerprinting method for IWSN devices. The robustness and uniqueness of the ICMP method in a LLN is however untested and needs to be further investigated. The methodology which needs to be used to test the efficacy of the ICMP ping method through experimentation was

outlined in Section 3.3 together with an explanation of the test bed used to collect experimental data in Section 3.4.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

This chapter presents the experimental results and is organized in three sections according to the network topology of the experiment. Section 4.2 compares the point-to-point RTT over a long and short distance, between two nodes by transmitting at maximum and minimum transmission power. The point-to-point experiment is specifically designed to evaluate the effect of the wireless channel on the RTT. The mesh topology experiment in Section 4.3 compares the RTT over multiple hops with and without Wi-Fi interference and was designed to evaluate the effect of additional hops on the total RTT. The final section (Section 4.4) evaluates the RTT in a star topology with and without Wi-Fi interference of the same device at different time instances in order to evaluate if a unique signature can be obtained for devices of the same type.

4.2 POINT-TO-POINT

The effect of the wireless channel was evaluated by placing two of the same type of nodes in a point-to-point topology while maintaining line-of-sight. The distance between the two nodes were varied and RTT measurements taken for each distance and device type.

4.2.1 Long distance, maximum transmission power

The results for the long distance, maximum transmission power for the SAMR21 development kit are shown in Figure 4.1. Two of the SAMR21 nodes were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each distance. The statistical measures for the data set are given in Table 4.1. A worst case median value of around 11.6ms was obtained for the three distances with the most deviation visible at 90m. No packet loss was observed during the experiment.

The results for the long distance, maximum transmission power for the CC2538 development kit are

Table 4.1. RTT for the SAM21R at maximum TX power (ms)

	0m	30m	60m	90m
Median	11.331	11.636	11.346	11.350
Mode	11.649	11.652	11.338	10.379
Average	11.327	12.236	11.617	11.419
Avg Dev	0.840	1.781	1.082	0.862
Std Dev	1.032	2.778	1.716	1.097
Variance	1.065	7.720	2.946	1.204

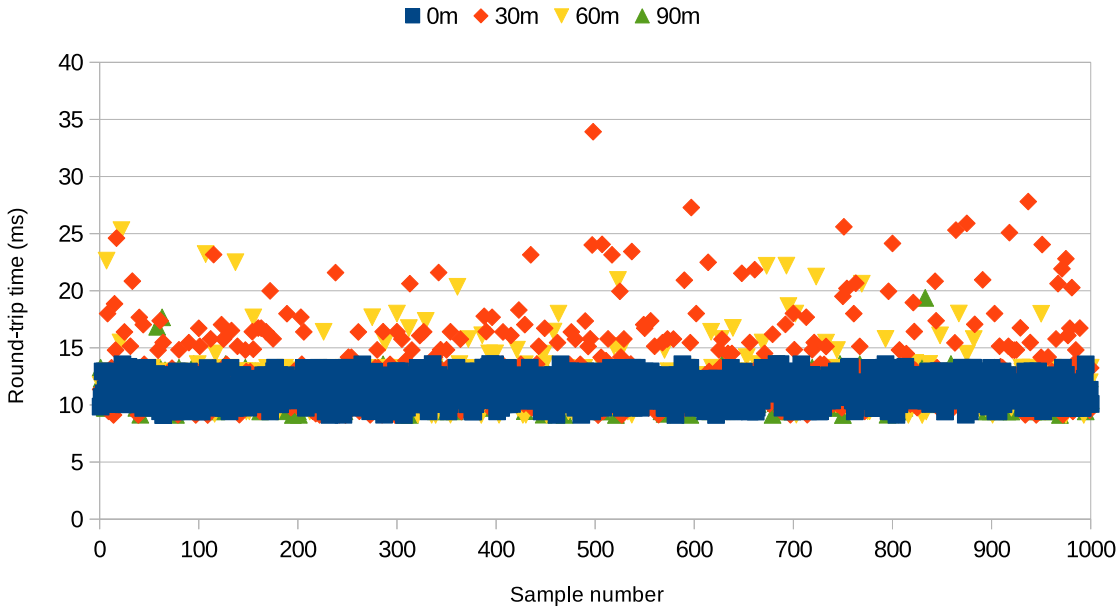

Figure 4.1. The maximum point-to-point, transmission power RTT in milliseconds is graphed as a function of the measurement number for the SAMR21 development kit. Two SAMR21 nodes were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each interval. Adapted from [9], © 2020 IEEE.

Table 4.2. RTT for the CC2538 at maximum TX power (ms)

	0m	30m	60m	90m
Median	8.594	8.594	8.595	16.758
Mode	8.594	8.594	8.595	8.595
Average	8.600	8.612	8.608	13.107
Avg Dev	0.012	0.033	0.024	4.036
Std Dev	0.023	0.244	0.193	4.060
Variance	0.001	0.059	0.037	16.484

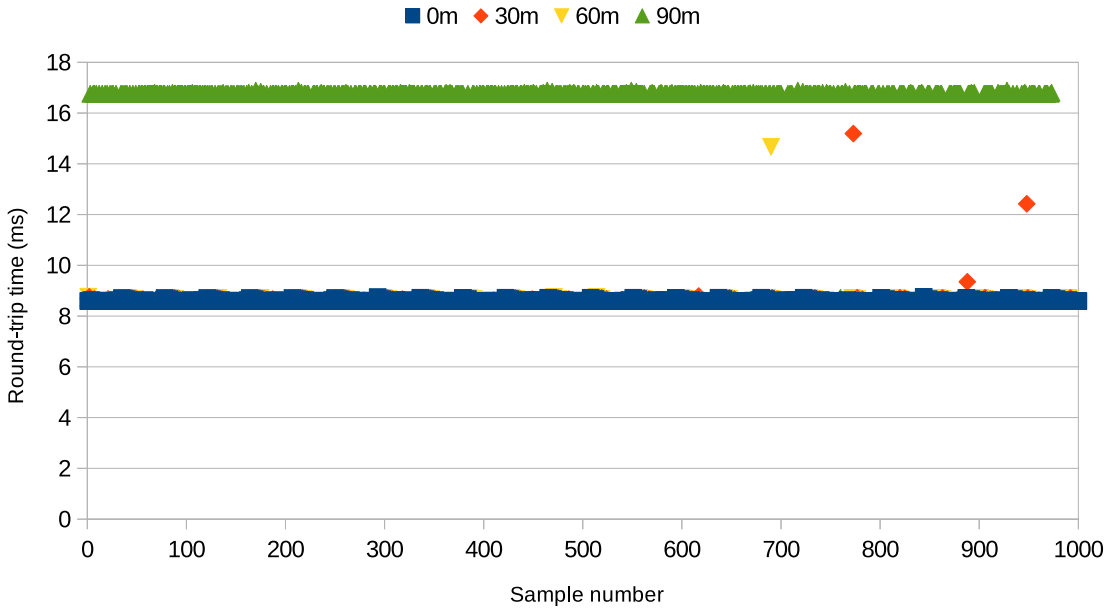

Figure 4.2. The maximum point-to-point, transmission power RTT in milliseconds is graphed as a function of the measurement number for the CC2538 development kit. Two CC2538 nodes were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each interval. Adapted from [9], © 2020 IEEE.

shown in Figure 4.2. Two of the CC2538 development kits were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each distance. The RTT in milliseconds is graphed as a function of the measurement number. The median remained steady at 8.5 ms for 0m, 30m and 60m and almost doubled at 90m indicating a retransmission due to a timeout. Some packet loss was observed at 90m and the increased RTT due to retransmission thus makes sense. The statistical measures for the data set are given in Table 4.2. The mode remains fairly constant even though the median jumps from 8.5 to 16.7 the variance is however much larger for the 90m results.

Table 4.3. RTT for the nRF52840 at maximum TX power (ms)

	0m	30m	60m	90m
Median	5.824	5.856	5.856	5.856
Mode	5.824	5.856	5.856	5.856
Average	5.826	5.857	5.861	5.862
Avg Dev	0.003	0.002	0.010	0.010
Std Dev	0.007	0.005	0.023	0.023
Variance	0.000	0.000	0.001	0.001

The results for the long distance, maximum transmission power for the nRF52840 development kit are shown in Figure 4.3. Two of the nRF52840 development kits were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each distance. The RTT in milliseconds is graphed as a function of the measurement number. The RTT for each of the four measurements can be distinguished on the graph and increases as the transmission distance became longer. The median remained constant at around 5.8ms without any packet loss. The variance and deviation remained insignificant over the average of the number of measurements. The statistical measures for the data set are given in Table 4.3.

4.2.2 Short distance, minimum transmission power

The results for the short distance, minimum transmission power for the nRF52840 development kit are shown in Figure 4.4. Two of the nRF52840 development kits were placed at intervals of 0cm, 10cm, 20cm, 30cm, 40cm, 50cm and 60cm apart and one thousand RTT measurements taken for each distance at minimum transmission power. The RTT increases as the transmission distance becomes longer in the same manner as the long distance equivalent experiment. The median remained roughly the same at 5.8ms while significant packet loss was observed at 60cm. The statistical measures for the data set are given in Table 4.4.

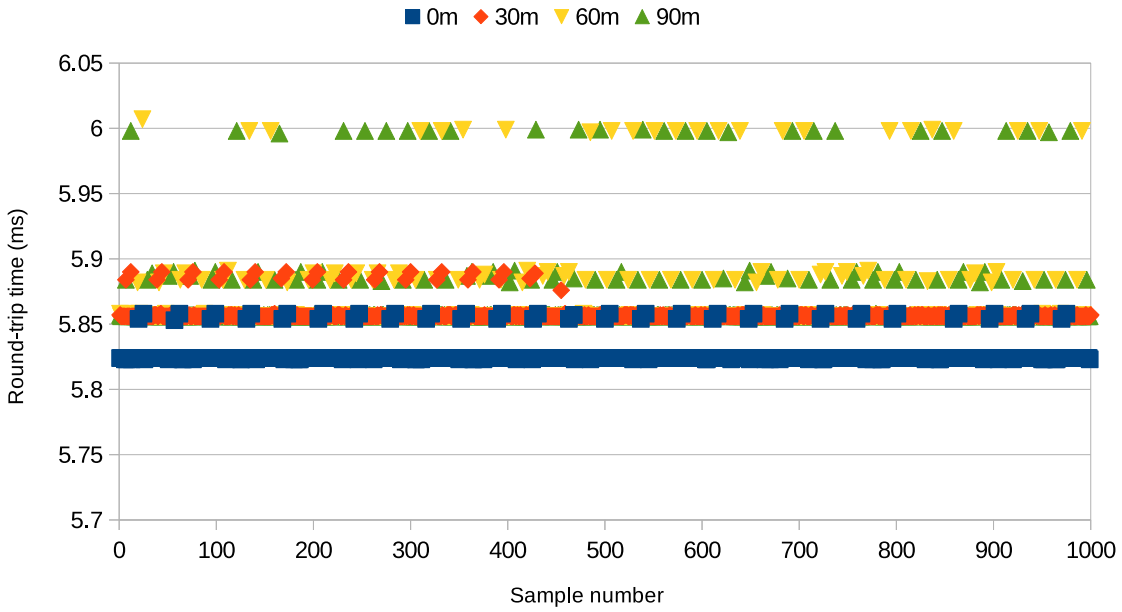

Figure 4.3. The maximum point-to-point transmission power RTT in milliseconds is graphed as a function of the measurement number for the nRF52840 development kit. Two nRF52840 nodes were placed at intervals of 0m, 30m, 60m and 90m apart and one thousand RTT measurements taken for each interval. Adapted from [9], © 2020 IEEE.

Table 4.4. RTT for the nRF52840 at minimum TX power (ms)

	0cm	10cm	20cm	30cm	40cm	50cm	60cm
Median	5.853	5.853	5.853	5.831	5.831	5.831	5.818
Mode	5.853	5.853	5.853	5.831	5.831	5.831	5.818
Average	5.876	5.854	5.853	5.833	5.848	5.828	5.830
Avg Dev	0.046	0.002	0.000	0.003	0.032	0.008	0.022
Std Dev	0.372	0.032	0.000	0.007	0.243	0.015	0.198
Variance	0.139	0.001	0.000	0.000	0.059	0.000	0.039

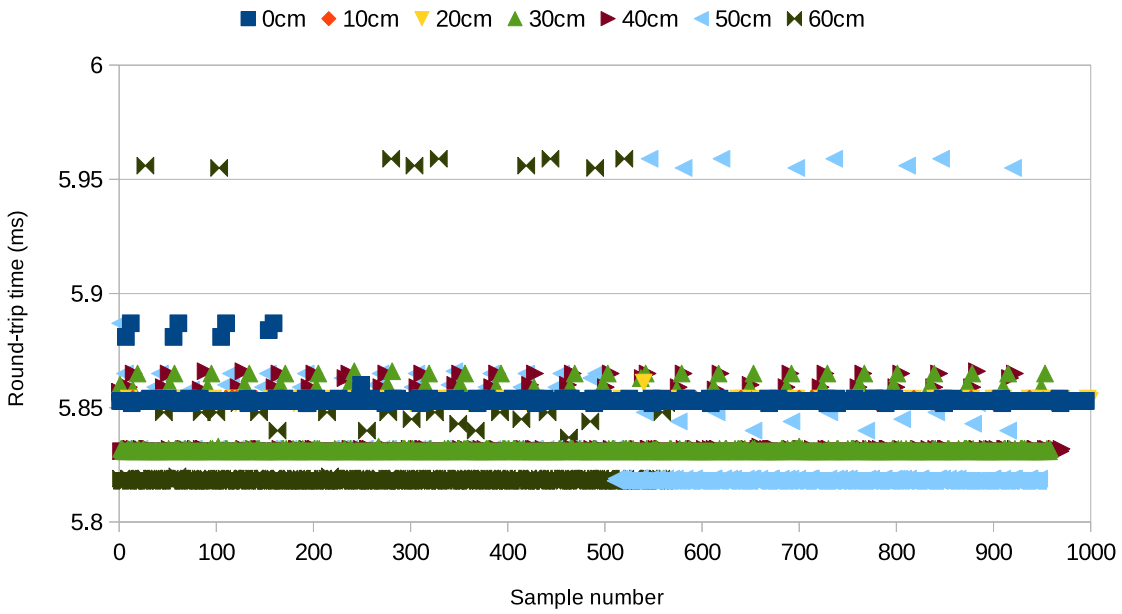

Figure 4.4. The minimum point-to-point transmission power RTT in milliseconds is graphed as a function of the measurement number for the nRF52840 development kit. Two nRF52840 nodes were placed at intervals of 0cm, 10cm, 20cm, 30cm, 40cm, 50cm and 60cm apart and one thousand RTT measurements taken for each interval. Taken from [9], © 2020 IEEE.

Table 4.5. RSSI for the nRF52840 at minimum TX power (dBm)

	0cm	10cm	20cm	30cm	40cm	50cm	60cm
Median	-47.000	-72.000	-73.000	-77.000	-80.000	-84.000	-92.000
Mode	-47.000	-72.000	-73.000	-77.000	-80.000	-84.000	-92.000
Average	-47.241	-72.052	-73.036	-76.854	-79.980	-84.031	-92.000
Avg Dev	0.458	0.104	0.069	0.253	0.128	0.117	0.000
Std Dev	0.555	0.252	0.186	0.359	0.410	0.298	0.000
Variance	0.308	0.063	0.035	0.129	0.168	0.089	0.000

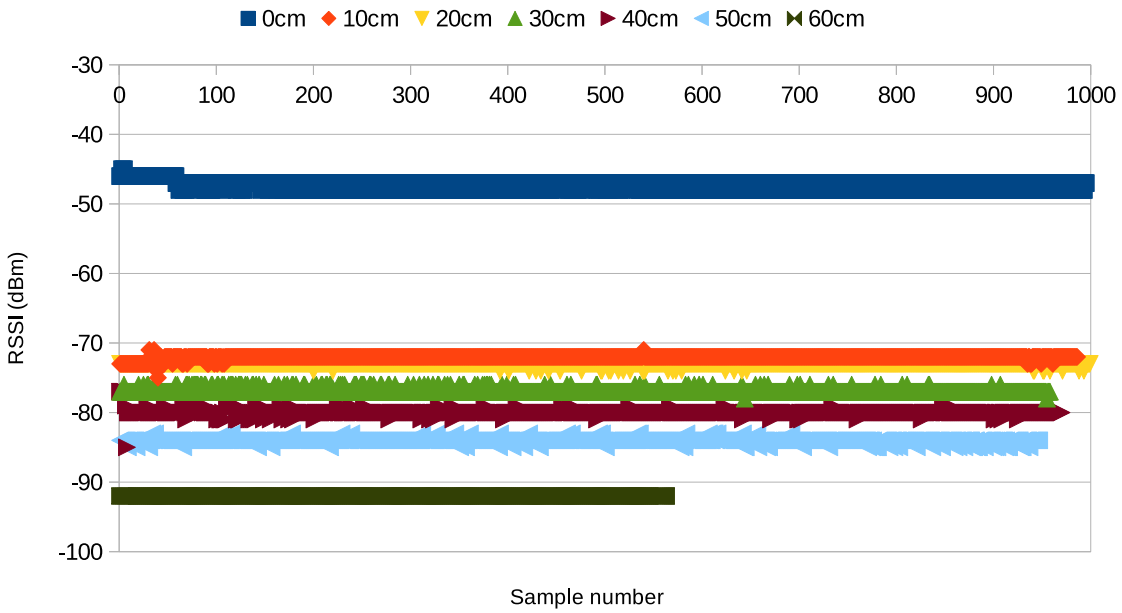

Figure 4.5. The minimum point-to-point transmission power RSSI in dBm is graphed as a function of the measurement number for the nRF52840 development kit. Two nRF52840 nodes were placed at intervals of 0cm, 10cm, 20cm, 30cm, 40cm, 50cm and 60cm apart and one thousand RTT measurements taken for each interval with the corresponding dBm value for the measurement. Taken from [9], © 2020 IEEE.

The short distance, minimum transmission power RSSI for the nRF52840 development kit are shown in Figure 4.5. Two of the nRF52840 development kits were placed at intervals of 0cm, 10cm, 20cm, 30cm, 40cm, 50cm and 60cm apart and one thousand RTT measurements taken for each distance. The RSSI in dBm is graphed as a function of the measurement number. The RSSI increased as the distance increased with significant packet loss at 60cm as indicated by the missing measurements. The data set has almost no deviation or variance. The statistical measures for the data set are given in Table 4.5.

4.2.3 Stack processing time

The localhost RTT time are shown in Figure 4.6. The ping6 with the localhost address was used to determine the stack latency on each device. The emulator and nRF52840 had the smallest RTT, but the nRF52840 had significantly less variation and deviation. The CC2538 was the slowest device and has more variance and deviation than the SAMR21XPRO which is faster. The statistical measures for the data set are given in Table 4.6.

Table 4.6. Stack processing time (ms)

	Emulator	NRF52840	SAMR21XPRO	CC2538DK
Median	0.352	0.331	0.523	1.369
Mode	0.353	0.331	0.523	1.369
Average	0.317	0.331	0.523	1.370
Avg Dev	0.064	0.000	0.000	0.002
Std Dev	0.086	0.001	0.000	0.009
Variance	0.007	0.000	0.000	0.000

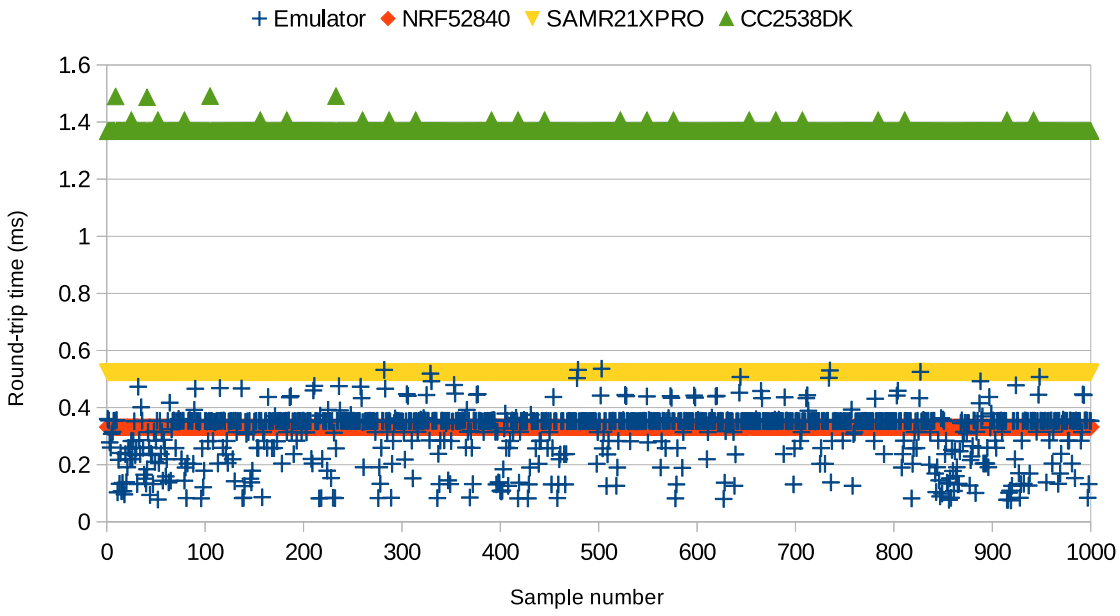

Figure 4.6. The RTT in milliseconds for the localhost ping6 command is graphed as a function of the measurement number. The ping6 command with the localhost address was used to determine the stack latency on each device.

4.3 MESH TOPOLOGY

The multi-hop timing was evaluated by placing five nRF52840DK nodes in a point-to-point topology with each node only able to communicate with the next node in the network. Routing was statically configured to ensure that the results would not be biased by routing topology changes. A distance of thirty meters was selected between devices to ensure optimal RSSI levels as per the results of the previous section. Data was obtained by running a ping6 command on the first device in the topology to each of the nodes in the topology in a closest node first sequence.

4.3.1 No Wi-Fi interference in measurement

Table 4.7. RTT for each hop in the mesh topology without Wi-Fi interference (ms)

	NODE_1	NODE_2	NODE_3	NODE_4
Median	5.830	11.585	17.333	23.064
Mode	5.830	11.585	17.333	23.064
Average	5.830	11.585	17.333	23.064
Avg Dev	0.000	0.000	0.000	0.000
Std Dev	0.000	0.000	0.000	0.000
Variance	0.000	0.000	0.000	0.000

The results from the mesh topology experiment without Wi-Fi interference are shown in Figure 4.7. IEEE 802.15.4 channel number 26 was used to avoid Wi-Fi interference. The ping6 command was used to obtain one thousand RTT measurements for each node. The RTT in milliseconds is graphed as a function of the node number. Three distinct lines representing each node is visible with no deviation, variance or any packet loss. The statistical measures for the data set are given in Table 4.7.

The results from the mesh topology experiment (Figure 4.7) without Wi-Fi interference is averaged for each node and shown in Figure 4.8. The RTT in milliseconds is graphed as a function of the node number. A curve fitted equation is given for the averaged results.

4.3.2 Wi-Fi interference included in measurement

The results from the mesh topology experiment with Wi-Fi interference are shown in Figure 4.9. IEEE 802.15.4 channel number 18 was used to include Wi-Fi interference. The ping6 command was used to obtain one thousand RTT measurements for each node. Three distinct lines representing each node is visible with negligible deviation, no variance and significant packet loss as the number of hops increased. The statistical measures for the data set are given in Table 4.8.

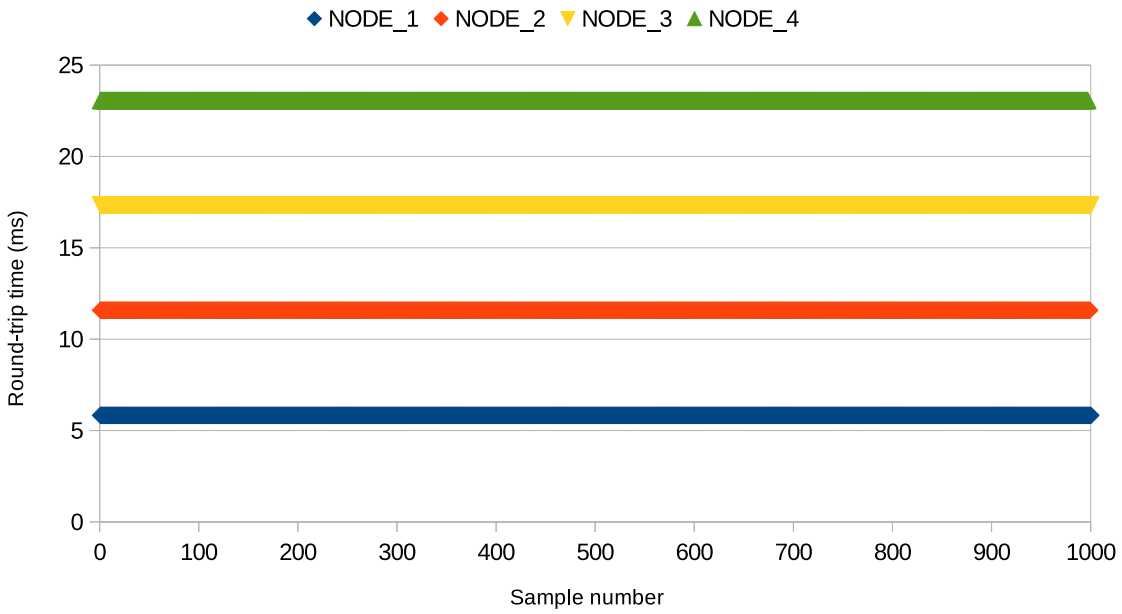

Figure 4.7. The results for the mesh topology experiment without Wi-Fi interference are shown. The RTT in milliseconds is graphed as a function of the node number. The ping6 command was used to obtain one thousand RTT measurements for each node and IEEE 802.15.4 channel 26 used to avoid Wi-Fi interference. Taken from [9], © 2020 IEEE.

Table 4.8. RTT for each hop in the mesh topology with Wi-Fi interference (ms)

	Node_1	Node_2	Node_3	Node_4
Median	5.830	11.585	17.333	23.064
Mode	5.830	11.585	17.333	23.064
Average	5.830	11.585	17.333	23.064
Avg Dev	0.000	0.000	0.001	0.000
Std Dev	0.000	0.000	0.001	0.000
Variance	0.000	0.000	0.000	0.000

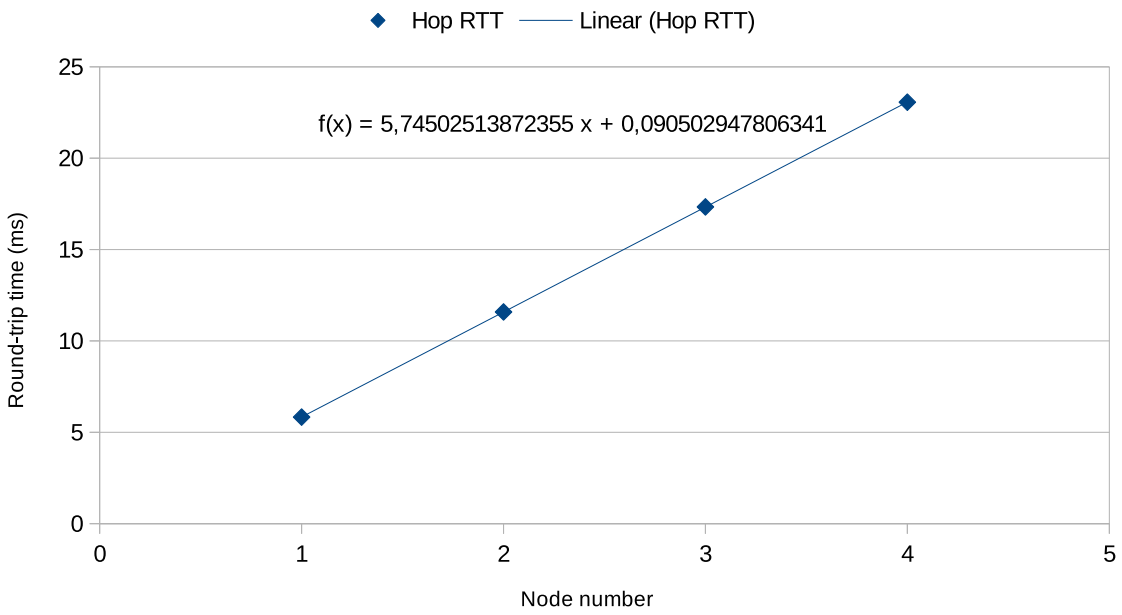

Figure 4.8. Results from the mesh topology experiment without Wi-Fi interference is averaged for each node and shown in the graph. A curve fitted equation is given for the averaged results. Taken from [9], © 2020 IEEE.

The results from the mesh topology experiment (Figure 4.9) without Wi-Fi interference is averaged for each node and shown in Figure 4.10. The RTT in milliseconds is graphed as a function of the node number. A curve fitted equation is given for the averaged results.

4.4 STAR TOPOLOGY

The feasibility of device fingerprinting using the ICMP ping method was evaluated by deploying devices in a star topology. The border router consisted of a nRF52840DK with direct line of sight connectivity to three nRF52840DK, two SAMR21XPRO, and two CC2538DK development boards. Data was obtained by running a ping6 command on the border-router to each of the nodes in the star topology. A distance of one meter was used between devices to try and evaluate the inter-device interference. An identical data collection application was flashed on each device to make the fingerprinting evaluation as realistic as possible.

The SAMR21XPRO kit did not produce any viable fingerprinting data. Similar results were obtained for the SAMR21XPRO kit as in Figure 4.1 showing a rather large deviation in readings with no recognizable pattern between devices. As a result the SAMR21XPRO devices were excluded from

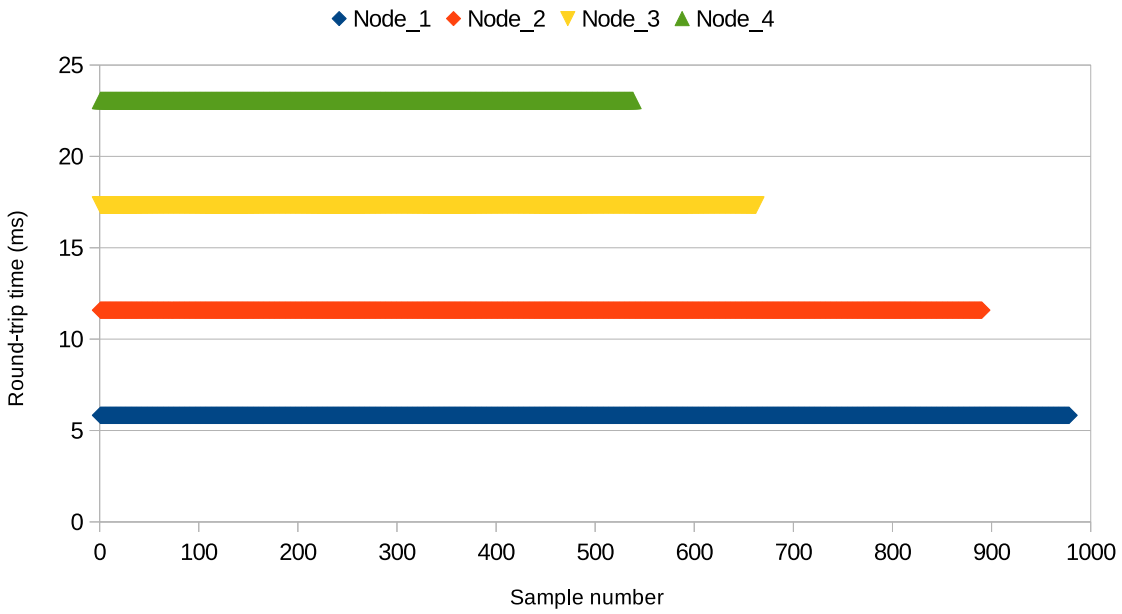

Figure 4.9. The results for the mesh topology experiment with Wi-Fi interference are shown. The RTT in milliseconds is graphed as a function of the node number. The ping6 command was used to obtain one thousand RTT measurements for each node and IEEE 802.15.4 channel 18 used to include Wi-Fi interference

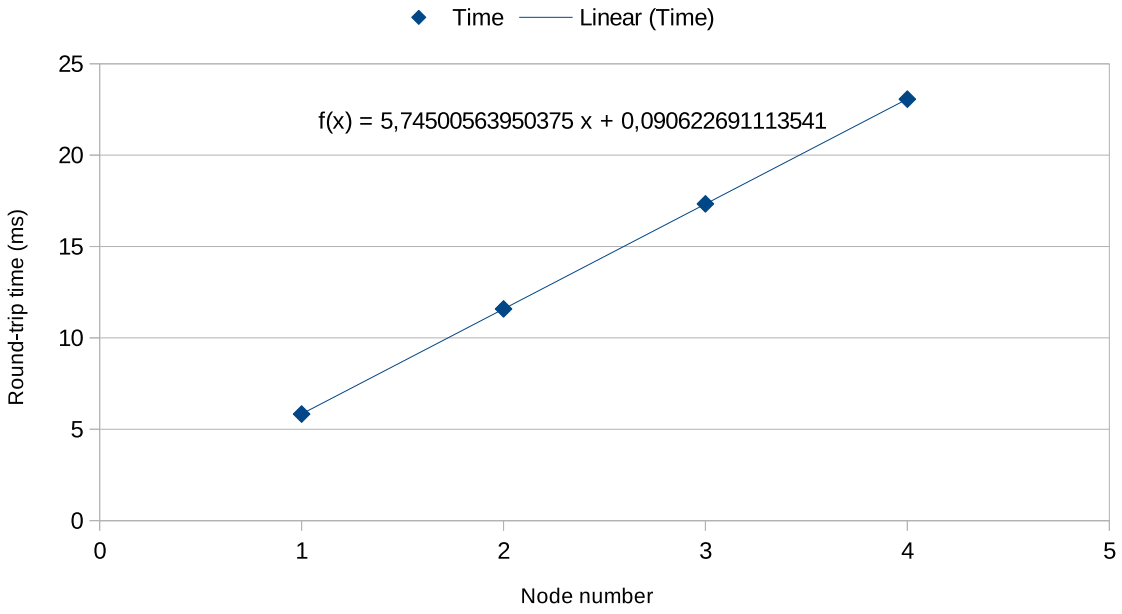

Figure 4.10. Results from the mesh topology experiment with Wi-Fi interference is averaged for each node and shown in the graph. A curve fitted equation is given for the averaged results.

further experiments.

4.4.1 No Wi-Fi interference in measurement

Table 4.9. Consecutive RTT measurement data for CC2538 without Wi-Fi interference (ms)

	CC 1A	CC 2A	CC 1B	CC 2B
Median	5.060	5.060	5.060	5.059
Mode	5.060	5.060	5.060	5.059
Average	5.068	5.066	5.065	5.065
Avg Dev	0.014	0.012	0.010	0.011
Std Dev	0.024	0.021	0.019	0.020
Variance	0.001	0.000	0.000	0.000

The experimental results for the star topology with the CC2538 development kit are shown in Figure 4.11. Two of the CC2538 nodes were placed one meter apart in a star topology and one thousand RTT measurements taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B. The experiment was conducted in a noise free environment using a Faraday cage and thus isolated from any Wi-Fi interference. The RTT in milliseconds is graphed as a function of the measurement number. No noticeable difference in mean or average can be used to identify the devices. The statistical measures for the data set are given in Table 4.9.

Table 4.10. Consecutive RTT measurement data for nRF52840 without Wi-Fi interference (ms)

	NRF_1A	NRF_2A	NRF_3A	NRF_1B	NRF_2B	NRF_2B
Median	3.788	3.786	3.785	3.789	3.791	3.790
Mode	3.788	3.786	3.785	3.789	3.791	3.790
Average	3.793	3.788	3.792	3.789	3.791	3.790
Avg Dev	0.010	0.004	0.015	0.000	0.000	0.000
Std Dev	0.024	0.021	0.223	0.000	0.001	0.001
Variance	0.001	0.000	0.050	0.000	0.000	0.000

The experimental results for the star topology with the nRF52840 development kit are shown in Figure 4.12. Three of the nRF52840 nodes were placed one meter apart in a star topology and one thousand RTT measurements taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B. The experiment was conducted in a noise

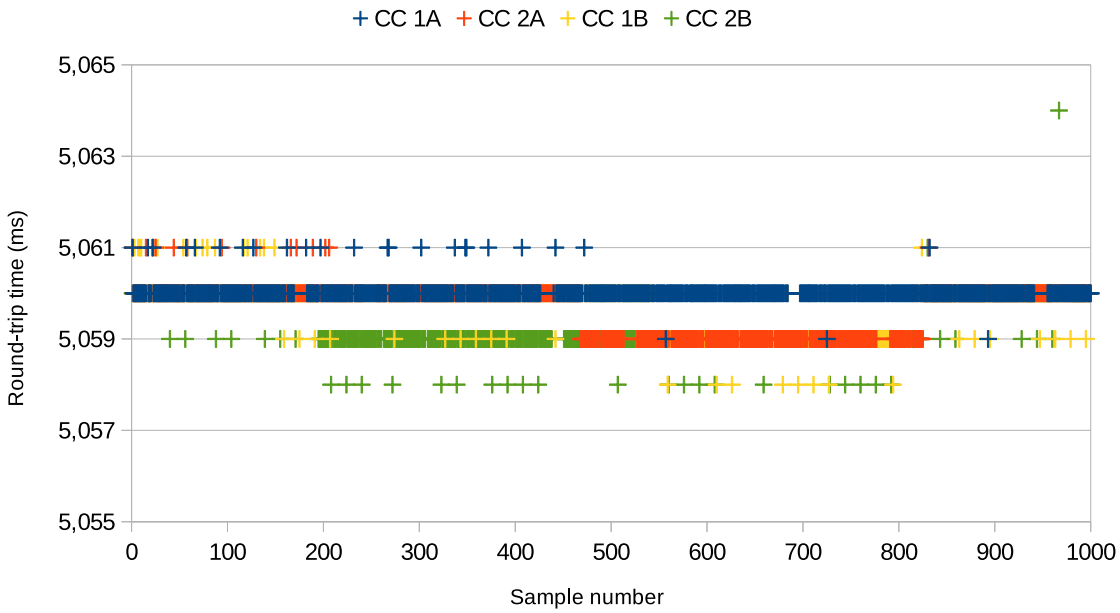

Figure 4.11. Experimental results for the star topology experiment without noise for the CC2538 development kit are given. The RTT in milliseconds is graphed as a function of the measurement number. One thousand RTT measurements were taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B.

free environment using a Faraday cage and thus isolated from any Wi-Fi interference. The RTT in milliseconds is graphed as a function of the measurement number. A noticeable difference in mean and average is visible, but not deterministic between measurements. The statistical measures for the data set are given in Table 4.10.

4.4.2 Wi-Fi interference included in measurement

The experimental results for the star topology with Wi-Fi interference for the CC2538 development kit are shown in Figure 4.13. Two of the CC2538 nodes were placed one meter apart in a star topology and one thousand RTT measurements taken for each distance, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B. The experiment was conducted in an environment with known Wi-Fi interference. The RTT in milliseconds is graphed as a function of the measurement number. A noticeable difference in mean and average is visible, but not deterministic between measurements. Packet loss is present and retransmissions visible on the graph. The statistical measures for the data set are given in Table 4.11.

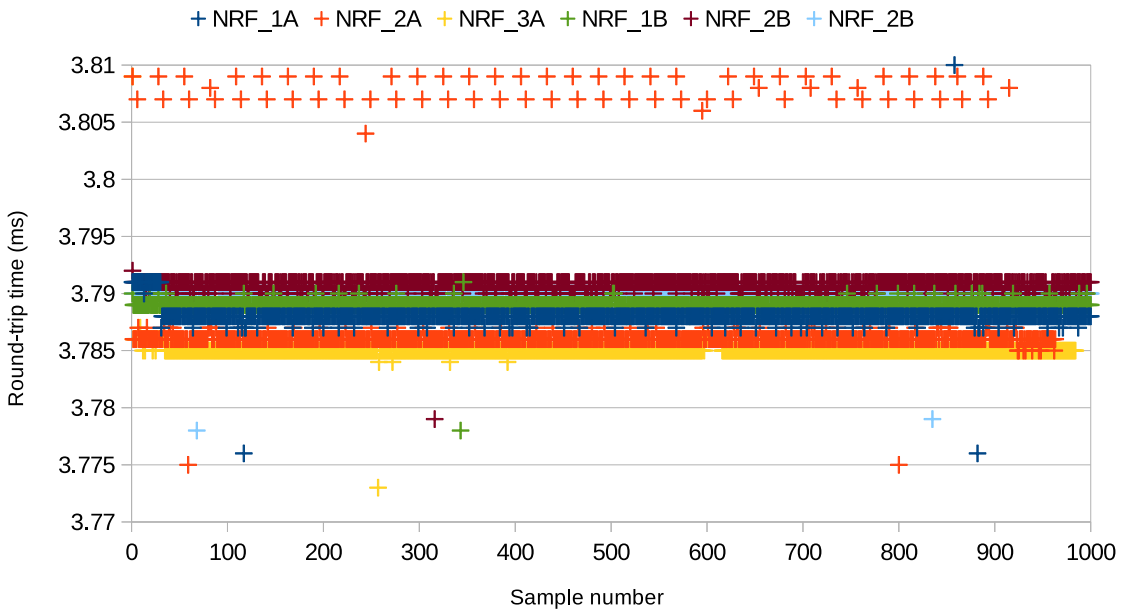

Figure 4.12. Experimental results for the star topology experiment without noise for the nRF52840 development kit are given. The RTT in milliseconds is graphed as a function of the measurement number. One thousand RTT measurements were taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B.

Table 4.11. Consecutive RTT measurement data for CC2538 with Wi-Fi interference (ms)

	CC_1A	CC_2A	CC_1B	CC_2B
Median	5.067	5.076	5.179	5.096
Mode	5.067	5.076	5.179	5.096
Average	5.070	5.081	5.181	5.098
Avg Dev	0.006	0.008	0.004	0.003
Std Dev	0.013	0.017	0.009	0.008
Variance	0.000	0.000	0.000	0.000

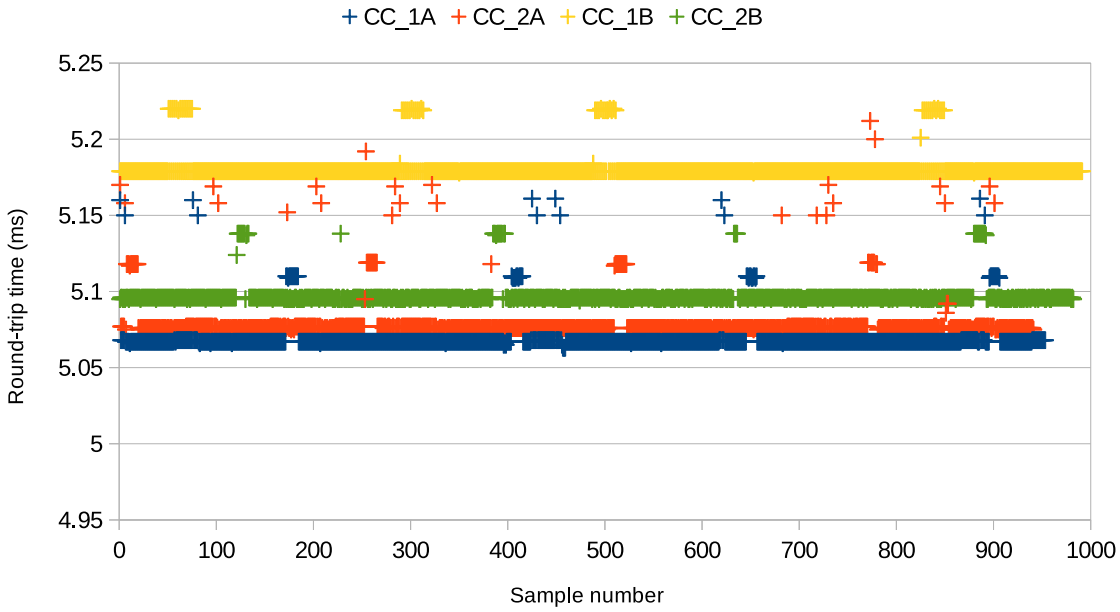

Figure 4.13. Experimental results for the star topology experiment with noise for the CC2538 development kit are given. The RTT in milliseconds is graphed as a function of the measurement number. One thousand RTT measurements were taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labelled as B.

Table 4.12. Consecutive RTT measurement data for nRF52840 without Wi-Fi interference (ms)

	NRF_1A	NRF_2A	NRF_3A	NRF_1B	NRF_2B	NRF_3B
Median	3.780	3.782	3.781	3.775	3.782	3.776
Mode	3.780	3.782	3.781	3.775	3.782	3.776
Average	3.780	3.782	3.781	3.784	3.782	3.783
Avg Dev	0.000	0.000	0.000	0.016	0.000	0.014
Std Dev	0.000	0.000	0.000	0.126	0.000	0.168
Variance	0.000	0.000	0.000	0.016	0.000	0.028

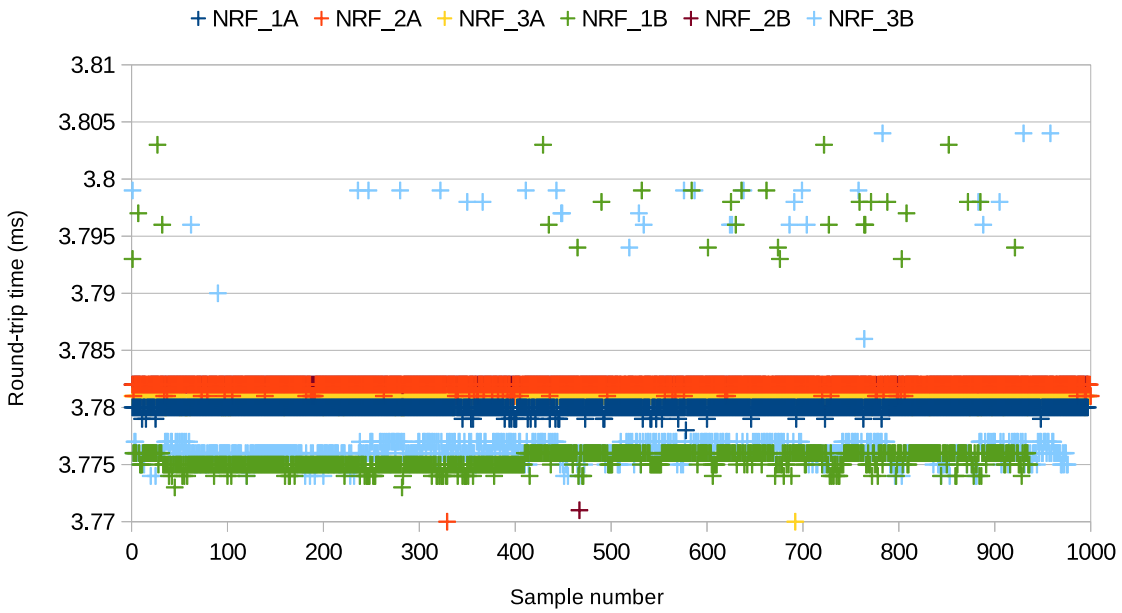

Figure 4.14. Experimental results for the star topology experiment with noise for the nRF52840 development kit are given. The RTT in milliseconds is graphed as a function of the measurement number. One thousand RTT measurements were taken for each node, labeled as A. The experiment was repeated a second time at a different time interval and labeled as B.

The experimental results for the star topology with Wi-Fi interference for the nRF52840 development kit are shown in Figure 4.13. Three of the nRF52840 nodes were placed one meter apart in a star topology and one thousand RTT measurements taken for each distance, labelled as A. The experiment was repeated a second time at a different time interval and labelled as B. The experiment was conducted in an environment with known Wi-Fi interference. The RTT in milliseconds is graphed as a function of the measurement number. A noticeable difference in mean and average is visible, but not deterministic between measurements. packet loss is present with non-deterministic scattering of measurements in some instances. The statistical measures for the data set are given in Table 4.11.

4.5 CHAPTER SUMMARY

Experimental data was collected and organized according to the network topology of the experiment. Data from the point-to-point experiments (Section 4.2.) showed minimal packet loss and variation in RTT when a signal strength of at least 70dBm was maintained. Devices of the same vendor had distinct fingerprintable RTT in each measurement. The mesh topology experiment in Section 4.3 compares the RTT over multiple hops with and without Wi-Fi interference and was designed to evaluate the effect of

additional hops on the total RTT. Wi-Fi interference was found to contribute significantly to packet loss which could be averaged out if enough measurements were taken. Section 4.4 evaluated the RTT in a star topology at different time instances. The RTT fluctuated as time passed and no fingerprintable metrics could be found for devices of the same make and model.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

The experimental results obtained from Chapter 4 is organised and discussed to address the research objectives and questions posed in Chapter 1. The chapter starts by discussing the effect of the wireless channel in Section 5.2 and continues with a discussion on the multi-hop timing in Section 5.3. The chapter is concluded with Section 5.4 where the efficacy of the ICMP method is discussed taking into consideration the results obtained from the point-to-point, mesh and star topology experiments.

5.2 EFFECT OF THE WIRELESS CHANNEL

The results from the three point-to-point experiments suggest that the effect of the wireless channel can be observed as a random deviation in the round-trip time and only varies significantly once the distance between the nodes increase beyond a maximum distance. The transmission range of a wireless signal is directly proportional to the transmission power of the wireless signal and can thus be substituted for received signal strength. The timing variations will thus remain within a bounded interval as the received signal strength remains above a predetermined level. The selection of an interference free RF channel is a key requirement for the random deviations to remain bounded.

The effect of the wireless channel can thus be quantified as a random variation in the round-trip time measurement. The variation remains statistically bounded in the microsecond range of which the baseline values are dependent on the type of device and RF environment.

5.3 MULTI-HOP TIMING

The data obtained from the multi-hop timing experiments showed distinct and identifiable timing characteristics. Round-trip times in the millisecond range were observed and are an order of magnitude larger than the effect of the physical channel and are thus reliable. Four distinct lines can be seen for each of the four measurements without packet loss or jitter corresponding to the number of hops

in the network. The experiment was repeated several times during different times in a location with high levels of Wi-Fi interference. Wi-Fi interference caused ICMP packets to drop and thus did not influence the average of the measurements as the packet is automatically discarded and not included in the averaging process.

Averaging the results obtained from the multi-hop experiments and graphing according to node number produces a set of data to which a straight line can be fitted. The resulting equation is shown below.

$$f(x) = 5.745x + 0.0905 \tag{5.1}$$

Where f(x) is the expected round-trip time and x is the hop-count. Equation (5.1) can be further simplified by forcing an intercept at zero resulting in the equation :

$$f(x) = 5.775x \tag{5.2}$$

A rule of thumb can thus be deduced from (5.2) for the NRF52840DK devices running Riot-OS where every additional hop will add 5.775 ms in latency.

5.4 EFFICACY OF THE ICMP METHOD

The ability to introduce device fingerprinting based on the ICMP method is discussed according to the fingerprinting criteria identified and modified for IWSNs in Chapter 3 . Universality, uniqueness, collectability robustness and data-dependency will be discussed from a semi-active perspective as the ping6 command can connect to a device from a remote location, but not make changes to the device itself.

5.4.1 Universality

The universality of the ICMP method was the most attractive feature of the ICMP method. The majority of IoT devices supports the measurement of RTT using ICMP packets and all the experimental data was collected using an unmodified version of the stock ping6 command provided by Riot-OS. The ICMP method thus fully meets the universality requirement for device fingerprinting.

5.4.2 Uniqueness

Data obtained from the star topology experiments shows that RF interference either adds or subtracts from the round-trip time in a random manner which is dependent on the environment and inter-device interference. What might seem as a pattern or ordered deviation in the point-to-point experiment is actually just noise. Extracting fingerprinting information at the microsecond level is thus not possible due to the randomizing effect of the noise floor and even if the noise floor could be reduced the ICMP processing of the networking stack in Riot-OS has no obvious fingerprinting characteristics in the microsecond timing ranges as shown by the localhost RTT experiment. The ICMP method thus fails to achieve the uniqueness requirement for devices of the same type.

5.4.3 Collectability

Collecting RTT measurements using the ping6 command is a trivial process and does not require modification to the hardware and software of a WSN device if the ping6 command is supported. Obtaining RTT data uses no more processing power or battery life than a normal sensor measurement would and can even be collected in parallel to the sensor measurements when raw packet data is used to determine the delta in arrival time. Automating data collection or implementing routing decisions based on RTT data is however more involved and will require supporting infrastructure for making intelligent decisions.

5.4.4 Robustness

The averaged RTT for each device did not remain stable over time as shown by the data obtained from measuring the RTT to the same device at different time intervals in the star topology experiment. The mean value shifted either adding or subtracting to the initial value as time progressed, violating the robustness requirement for generating device fingerprints.

5.4.5 Data-dependency

The amount of noise and Wi-Fi interference will influence the number of measurements required to successfully converge on an accurate mean. A thousand RTT measurements were used to confirm that experimental results are reliable, but in reality far less measurements are required. Most of the experimental data can be obtained by fifty or less measurements, but is highly dependable on the interference which effects the bounded behaviour of the RTT. The ICMP method is thus viable if low levels of interference or high levels of signal strength can be maintained in the network from a data-dependency point of view.

5.5 CHAPTER SUMMARY

The effect of the wireless channel was discussed in Section 5.2 and can be observed as a random deviation in the round-trip time. The deviation remains bounded when sufficient signal strength is maintained. The multi-hop timing results in Section 5.3 produced a linear equation for estimating RTT and a rule of thumb that can be used in the design stage of an IWSN. The discussion concluded in Section 5.4 where the efficacy of the ICMP method is discussed taking into consideration the results obtained from the point-to-point, mesh and star topology experiments. The ICMP method failed to meet the uniqueness and robustness requirement for device fingerprinting of devices from the same vendor.

CHAPTER 6 CONCLUSION

6.1 SUMMARY

The feasibility of using the ICMP method to generate fingerprinting information in an IWSN was evaluated. A linear relationship was found between hop count and round-trip time for a static network with reasonable signal strength. The physical layer effect of the Lossy, Low-Power Network can be averaged out if several measurements are averaged. The multi-hop round-trip time estimation can thus be used in the design phase of any IWSN network to estimate latency or alternatively used as a method to detect routing anomalies for enhanced security or fault diagnosis.

The ICMP method was able to differentiate between devices from different vendors using millisecond timing deltas. The timing results for identifying unique nodes from the same vendor drifted between consecutive experiments due to physical layer noise. The ICMP ping method is thus not suitable for fingerprinting devices with a response time in the microsecond range.

The experiments showed that microsecond and below delta measurements are influenced by the randomness of the RF channel and thus randomizing the delta measurements according to the noise and multi-path components experienced by the physical layer. Phenomenon in the millisecond ranges are easy to fingerprint, if they exist because the timing deltas are an order of magnitude larger than the randomness introduced by the physical layer. A device with a microsecond or faster response time is difficult to fingerprint, if not impossible due to the small timing offsets caused by the effects of the physical layer. Devices with sub-millisecond responses are thus more secure against fingerprinting attacks in IWSNs.

6.2 RESEARCH CONTRIBUTION

This thesis makes the following contributions:

- Contributes an OSI stack based approach to identifying, sorting and evaluating related work and the inter-stack dependencies of related fingerprinting methods.
- Provides evaluation criteria for IWSN fingerprinting methods taking into consideration the unique design criteria of an IWSN.
- Quantifies the expected timing requirements and related design parameters for IWSN round-trip-time measurements.
- Presents a new empirical method to determine the latency of a static routed multi-hop IWSN deployment.
- Provides baseline experimental results for the use of round trip based device fingerprinting methods in an IWSN.

6.3 FUTURE WORK

A lack of formal fingerprinting tools for 802.15.4 networks was highlighted in the related work section and should be further explored. There are several fingerprinting methods in related work which have not yet been adapted and evaluated in IWSN networks.

The effects of time slotted communications (802.15.4e) on the ICMP ping method was not evaluated in this paper as part of a scope restriction and should be further investigated on a suitable emulator or test bed. Maintaining a large 6LowPAN mesh network with static routing is impractical and the experiments should be expanded based on common deployment scenarios in an IWSN.

Neighbor discovery is a key attribute of timely communication in the RPL based IWSN [76] and should be an area of focus for mobility, deterministic behaviour and device fingerprinting. The speed at which neighbours are discovered, updated and added to the routing tables is a key attribute of timely communication and has a much greater impact on latency and determinism than the microsecond delays introduced on the physical layer.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] L. Zhou, K.-H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the Industrial Internet of Things: An overview of approaches to safeguarding endpoints," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, 2018.

[3] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in Industrial Wireless Sensor Networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, 2018, article no. 2491.

[4] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, "A critical analysis of research potential, challenges, and future directives in Industrial Wireless Sensor Networks," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 1, pp. 39–95, 2018.

[5] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: challenges and opportunities," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[6] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019, article no. 13.

[7] B. Cheng, J. Zhang, G. P. Hancke, S. Karnouskos, and A. W. Colombo, "Industrial Cyber-Physical Systems: Realizing cloud-based big data infrastructures," *IEEE Industrial Electronics Magazine*, vol. 12, no. 1, pp. 25–35, 2018.

[8] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of Radio Frequency Fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[9] C. P. Kruger and G. P. Hancke, "Enhanced security in Industrial internet of Things networks using latency based fingerprinting," in *Proceedings of the 18th IEEE International Conference on Industrial Informatics (INDIN)*, 2020, pp. 100–106.

[10] K. Yang, Q. Li, and L. Sun, "Towards automatic fingerprinting of IoT devices in the cyberspace," *Computer Networks*, vol. 148, pp. 318–327, 2019.

[11] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal, "IoT device identification via network-flow based fingerprinting and learning," in *Proceedings of the 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, 2019, pp. 103–111.

[12] C. Kruger and G. P. Hancke, "Benchmarking internet of things devices," in *Proceedings of the 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 611–616.

[13] L. P. Ledwaba, G. P. Hancke, H. S. Venter, and S. J. Isaac, "Performance costs of software cryptography in securing new-generation internet of energy endpoint devices," *IEEE Access*, vol. 6, pp. 9303–9323, 2018.

[14] Y. Lee, J. Li, and Y. Kim, "MicPrint: Acoustic sensor fingerprinting for spoof-resistant mobile device authentication," in *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2019, pp. 248–257.

[15] J. Hemmes and J. Dressler, "Work-In-Progress: IoT device signature validation," in *Proceedings of the 10th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 43–48.

[16] A. Ashtari, A. Shabani, and B. Alizadeh, "A new RF-PUF based authentication of Internet of Things using Random Forest Classification," in *Proceedings of the 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, 2019, pp. 21–26.

[17] E. Uzundurukan, A. Ali, Y. Dalveren, and A. Kara, "Performance analysis of modular RF front-end for RF fingerprinting of Bluetooth devices," *Wireless Personal Communications*, vol. 112, pp. 2519–2531, 2020.

[18] E. Uzundurukan, Y. Dalveren, and A. Kara, "A database for the Radio Frequency Fingerprinting of Bluetooth devices," *Data*, vol. 5, 2020, article no. 55.

[19] A. Ali and G. Fischer, "Symbol based statistical RF fingerprinting for fake base station identification," in *Proceedings of the 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2019, pp. 1–5.

[20] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in Cognitive Communication Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 160–167, 2018.

[21] S. Balakrishnan, S. Gupta, A. Bhuyan, P. Wang, D. Koutsonikolas, and Z. Sun, "Physical layer identification based on spatial-temporal beam features for millimeter-wave wireless networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1831–1845, 2020.

[22] J. Yu, A. Hu, F. Zhou, Y. Xing, Y. Yu, G. Li, and L. Peng, "Radio Frequency Fingerprint Identification based on denoising autoencoders," in *Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 1–6.

[23] F. Zhao and Y. Jin, "An optimized Radio Frequency Fingerprint extraction method applied to low-end receivers," in *Proceedings of the 11th IEEE International Conference on Communication Software and Networks (ICCSN)*, 2019, pp. 753–757.

[24] L. Zong, C. Xu, and H. Yuan, "A RF fingerprint recognition method based on Deeply Convolu-

tional Neural Network," in *Proceedings of the 5th IEEE Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020, pp. 1778–1781.

[25] Y. Tu, Z. Zhang, Y. Li, C. Wang, and Y. Xiao, "Research on the Internet of Things device recognition based on RF-fingerprinting," *IEEE Access*, vol. 7, pp. 37 426–37 431, 2019.

[26] Z. Pan, C.-N. Yang, V. Sheng, N. Xiong, and W. Meng, "Machine learning for wireless multimedia data security," *Security and Communication Networks*, vol. 112, pp. 2519–2531, 2019.

[27] Q. Wu, Y. Li, and Z. Zhang, "Research on the identification of IoT devices based on higher-order spectra," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1–6.

[28] A. Aghnaiya, Y. Dalveren, and A. Kara, "On the performance of variational mode decomposition-based Radio Frequency Fingerprinting of Bluetooth devices," *Sensors*, vol. 20, 2020, article no. 1704.

[29] Y. Lin and J. Chang, "Improving wireless network security based on Radio Fingerprinting," in *Proceedings of the 19th IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2019, pp. 375–379.

[30] A. M. Ali, E. Uzundurukan, and A. Kara, "Assessment of features and classifiers for Bluetooth RF fingerprinting," *IEEE Access*, vol. 7, pp. 50 524–50 535, 2019.

[31] M. Kose, S. Tascioglu, and Z. Telatar, "RF fingerprinting of IoT devices based on transient energy spectrum," *IEEE Access*, vol. 7, pp. 18 715–18 726, 2019.

[32] T. Bihl, T. Paciencia, K. Bauer, and M. Temple, "Cyber-physical security with RF fingerprint classification through distance measure extensions of Generalized Relevance Learning Vector Quantization," *Security and Communication Networks*, vol. 2020, pp. 1–12, 2020.

[33] Y. Li, Y. Lin, Z. Dou, and Y. Chen, "Research on RF fingerprint feature selection method," in *Proceedings of the 91st IEEE Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp.

1–5.

[34] J. M. McGinthy, L. J. Wong, and A. J. Michaels, "Groundwork for neural network-based specific emitter identification authentication for IoT," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6429–6440, 2019.

[35] T. Jian, B. C. Rendon, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "MAC ID spoofing-resistant radio fingerprinting," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[36] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust wireless fingerprinting via complex-valued neural networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[37] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 165–178, 2020.

[38] J. Bassey, D. Adesina, X. Li, L. Qian, A. Aved, and T. Kroecker, "Intrusion detection for IoT devices based on RF fingerprinting using deep learning," in *Proceedings of the Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019, pp. 98–104.

[39] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, "Finding a new needle in the haystack: Unseen radio detection in large populations using deep learning," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–10.

[40] G. Vaidya, A. Nambi, T. V. Prabhakar, V. Kumar T, and S. Sudhakara, "IoT-ID: a novel device-specific identifier based on unique hardware fingerprints," in *Proceedings of the IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2020, pp. 189–202.

[41] H. Liu, X. Li, L. Zhang, Y. Xie, Z. Wu, Q. Dai, G. Chen, and C. Wan, "Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint," in *Proceedings of the IEEE Conference on Computer Communications*, 2018, pp. 126–134.

[42] X. Li, H. Liu, L. Zhang, Z. Wu, Y. Xie, G. Chen, C. Wan, and Z. Liang, "Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1945–1958, 2019.

[43] J. Zhang, A. R. Beresford, and I. Sheret, "SensorID: Sensor calibration fingerprinting for smartphones," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 638–655.

[44] Q. Gu, D. Formby, S. Ji, H. Cam, and R. Beyah, "Fingerprinting for Cyber-Physical System security: Device physics matters too," *IEEE Security and Privacy*, vol. 16, no. 5, pp. 49–59, 2018.

[45] F. Kandah, J. Cancelleri, D. Reising, A. Altarawneh, and A. Skjellum, "A hardware-software codesign approach to identity, trust, and resilience for IoT/CPS at scale," in *Proceedings of the International Conference on Internet of Things (iThings)*, 2019, pp. 1125–1134.

[46] C. M. Ahmed, M. Ochoa, J. Zhou, A. P. Mathur, R. Qadeer, C. Murguia, and J. Ruths, "NoisePrint: Attack detection using sensor and process noise fingerprint in cyber physical systems," in *Proceedings of the Asia Conference on Computer and Communications Security*, 2018, pp. 483–497.

[47] D. Stock and D. Schel, "Cyber-physical production system fingerprinting," in *Proceedings of the 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia*, 2019, pp. 393–398.

[48] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[49] M. Skowron, A. Janicki, and W. Mazurczyk, "Traffic fingerprinting attacks on Internet of Things using machine learning," *IEEE Access*, vol. 8, pp. 20 386–20 400, 2020.

[50] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of IoT devices," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, 2018, pp. 41–50.

[51] Y. Luo, H. Hu, Y. Wen, and D. Tao, "Transforming device fingerprinting for wireless security via online Multitask Metric Learning," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 208–219, 2020.

[52] A. Aksoy and M. H. Gunes, "Automated IoT device identification using network traffic," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[53] J. Berger, A. Klein, and B. Pinkas, "Flaw Label: Exploiting IPv6 flow label," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1259–1276.

[54] Y. Song, Q. Huang, J. Yang, M. Fan, A. Hu, and Y. Jiang, "IoT device fingerprinting for relieving pressure in the access control," in *Proceedings of the ACM Turing Celebration Conference*, 2019, article no. 143.

[55] W. Cheng, Z. Ding, C. Xu, X. Wu, Y. Xia, and J. Mao, "RAFM: A real-time auto detecting and fingerprinting method for IoT devices," *Journal of Physics: Conference Series*, vol. 1518, 2020, article no. 12043.

[56] K. Yang, Q. Li, X. Lin, X. Chen, and L. Sun, "iFinger: Intrusion detection in Industrial Control Systems via register-based fingerprinting," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 955–967, 2020.

[57] V. Mishra, P. Laperdrix, A. Vastel, W. Rudametkin, R. Rouvoy, and M. Lopatka, "Don't count me out: On the relevance of IP address in the tracking ecosystem," in *WWW '20: Proceedings of The Web Conference*, 2020, pp. 808–815.

[58] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, and A. S. Uluagac, "Z-IoT: Passive device-class fingerprinting of ZigBee and Z-Wave IoT devices," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[59] A. K. Dalai, A. Jena, S. Sharma, A. Mohapatra, B. Sahoo, M. S. Obaidat, B. Sadoun, and D. Puthal, "A fingerprinting technique for identification of wireless devices," in *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2018, pp. 1–5.

[60] S. Aneja, N. Aneja, and M. S. Islam, "IoT device fingerprint using deep learning," in *Proceedings of the IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 174–179.

[61] C. Shen, C. Liu, H. Tan, Z. Wang, D. Xu, and X. Su, "Hybrid-augmented device fingerprinting for intrusion detection in Industrial Control System Networks," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 26–31, 2018.

[62] V. Thangavelu, D. M. Divakaran, R. Sairam, S. S. Bhunia, and M. Gurusamy, "DEFT: A distributed IoT fingerprinting technique," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 940–952, 2019.

[63] T. Javed, M. Haseeb, M. Abdullah, and M. Javed, "Using application layer banner data to automatically identify IoT devices," *ACM SIGCOMM Computer Communication Review*, no. 3, pp. 23–29, 2020.

[64] Q. Li, X. Feng, R. Wang, Z. Li, and L. Sun, "Towards fine-grained fingerprinting of firmware in online embedded devices," in *Proceedings of the IEEE Conference on Computer Communications*, 2018, pp. 2537–2545.

[65] G. Palfinger and B. Prünster, "AndroPRINT: analysing the fingerprintability of the Android API," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, article no. 94.

[66] G. Celosia and M. Cunche, "Fingerprinting Bluetooth-Low-Energy devices based on the Generic Attribute Profile," in *Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things*, 2019, pp. 24–31.

[67] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Clock around the clock: Time-based device fingerprinting," in *Proceedings of the Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1502–1514.

[68] D. Huang, K. Yang, W. Teng, and G. Chiu, "Design of client device identification by clock skew in clouds," in *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 1133–1138.

[69] A. Gomez-Boix, D. Frey, Y.-D. Bromberg, and B. Baudry, "A collaborative strategy for mitigating tracking through browser fingerprinting," in *Proceedings of the 6th ACM Workshop on Moving Target Defense*, 2019, pp. 67–78.

[70] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–29, 2012.

[71] K. Ren, Z. Qin, and Z. Ba, "Toward hardware-rooted smartphone authentication," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 114–119, 2019.

[72] L. , E. C. Ezin, and R. Sadre, "Proceedings of the efficient probing of heterogeneous IoT networks," in *Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 1052–1058.

[73] G. Lontorfos, K. D. Fairbanks, L. Watkins, and W. H. Robinson, "Remotely inferring device manipulation of Industrial Control Systems via network behavior," in *Proceedings of the 40th IEEE Local Computer Networks Conference Workshops (LCN Workshops)*, 2015, pp. 603–610.

[74] L. Watkins, W. H. Robinson, and R. Beyah, "A passive solution to the CPU resource discovery problem in Cluster Grid Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2000–2007, 2011.

[75] L. Watkins, W. H. Robinson, and R. Beyah, "Using network traffic to infer hardware state: A kernel-level investigation," *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 3, pp. 1–22, 2015.

[76] H. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for Low-Power and Lossy Networks (RPL): A survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.