AN INTRUSION DETECTION SYSTEM FOR SMART GRID NEIGHBORHOOD AREA NETWORK

by

Nasim Beigi Mohammadi

Bachelor of Engineering, Shahed University, 2008

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

An Intrusion Detection System for Smart Grid Neighborhood Area Network

Master of Science 2013

Nasim Beigi Mohammadi

Computer Science

Ryerson University

Abstract

Smart grid is expected to improve the efficiency, reliability and economics of current energy systems. Using two-way flow of electricity and information, smart grid builds an automated, highly distributed energy delivery network. In this thesis, we present the requirements for intrusion detection systems in smart grid, neighborhood area network (NAN) in particular. We propose an intrusion detection system (IDS) that considers the constraints and requirements of the NAN. It captures the communication and computation overhead constraints as well as the lack of a central point to install the IDS. The IDS is distributed on some nodes which are powerful in terms of memory, computation and the degree of connectivity. Our IDS uses an analytical approach for detecting Wormhole attack. We simulate wireless mesh NANs in OPNET Modeler and for the first time, we integrate our analytical model in Maple from MapleSoft with our OPNET simulation model.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Jelena Mišić, for guiding me along the research process and graciously supporting me throughout the course of the degree. I value her respect towards professionalism and the desire to excel higher and higher. She has made me competent in organizing and managing research activities and projects in applying them within the industry environment.

I wish to thank Dr. Vojislav B. Mišić for his useful insights and invaluable feedback. I am also thankful to Dr. Alireza Sadeghian, the head of computer science department, for his inspiring advice through this journey.

I would like to thank my dear husband, Dr. Hamzeh Khazaei, who has selflessly supported me from the first day of graduate school. Hamzeh has been a true and great supporter and has unconditionally loved me during my good and bad times. These past several years have not been an easy ride, both academically and personally . I truly thank Hamzeh for sticking by my side, even when I was depressed and desperate. I feel that what we both learned strengthened our commitment and determination to each other and to live life to the fullest.

Dedication

To my wonderful husband for his constant love and support.

Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

The utility industry is experiencing a major transformation that enhances energy systems by using advanced technologies and intelligent devices. Such a transformation is called smart grid which is expected to improve the efficiency, reliability and economics of current energy systems. According to the US department of energy (DoE) "smart grid generally refers to a class of technology that is trying to bring utility delivery systems into the 21st century".

For a century, utility companies have had to send workers out to gather much of the data needed to provide electricity. The workers read meters, look for broken equipment and measure voltage, for example. Most of the devices utilities use to deliver electricity have yet to be automated and computerized. Now, many options and products are being made available to the electricity industry to modernize it. A key feature of the smart grid is automation technology that lets the utility adjust and control each individual device or millions of devices from a central location.

Using two-way flow of electricity and information, smart grid builds an automated, highly distributed energy delivery network. It incorporates real-time information exchange with the intention to balance supply and demand [59], [47]. The main smart grid services include:

- Automatic meter reading.
- Power grid monitoring: electrical properties such as voltage and current of the power grid infrastructure are monitored.
- Demand side management: it is comprised of two parts:
  - load shifting/demand response.

  - energy conservation, e.g., using energy efficient products [10]

- Home networking between electrical devices for energy management.

- Vehicle to power grid technology: vehicles store power during off peak hours and send it back to the power grid during on peak hours.

Smart grid consists of seven major blocks namely: bulk generation, transmission, distribution, operation, market, customer and service provider [76]. A high-level smart grid framework is shown in Figure 1.1.

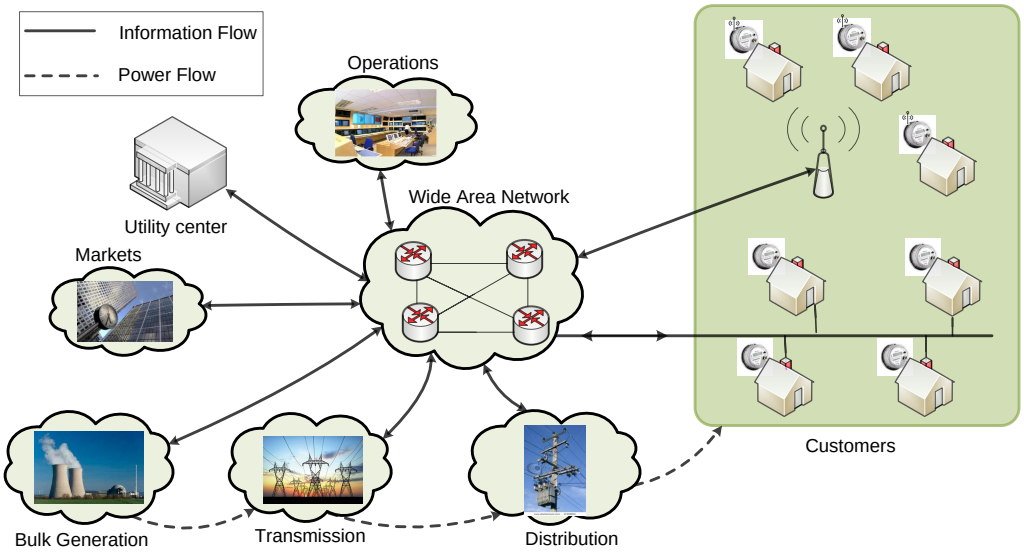

Figure 1.1: A high-level framework of smart grid.

1.1 Smart Grid Goals

There are some primary concerns when migrating form old utility delivery network to smart grid which are outlined below [3]:

– Customer participation: customers will receive price signals and adjust their electronic devices according to it. In addition, under the demand response program, some customers permit the utility companies to control their smart electric devices at home.

Hence, utilities are allowed to turn the customers devices off in case of emergency or during peak hours.

– Power quality for the 21st century: through monitoring the power factors such as current and voltage, the power work forces are able to identify the power grid problems.

– Integration of all generations and storage options: smart grid aims to integrate distributed electrical generations, e.g., micro grids and renewable energies with the power grid. Thus, managing the produced power would be easier.

– Self healing: the power grid would be able to heal itself automatically. It can decide based on the collected data and react dynamically.

– Resilience against attacks and disasters: this characteristic can be provided by increasing power grid robustness, protecting key assets from physical attacks and providing sufficient redundancy in the power grid [43]

– Asset management and operational efficiency: quality of assets and how efficient they are working in the power grid will be monitored. For example, the cable temperature is measured.

– New markets and operations: smart grid will integrate and open new businesses to the power grid. For instance, it integrates IT infrastructure to the power grid; smart devices are needed to be designed and communication infrastructures are needed to be developed.

To achieve the above-mentioned goals there is a need for communication infrastructure to provide a two-way communication between the utility side and customer side.

1.2 Smart Grid Communication Networks

The emergence of machine-to-machine (M2M) communication has also begun in developing smart power grid. such communication occurs among different components of smart grid such as sensors, smart meters, gateways and other intelligent devices [26]. A three-level hierarchy can be defined for smart grid communication network which includes the Home Area Network (HAN), Neigborhood area network and Wide Area Network (WAN). In smart grid advanced

metering infrastructure (AMI) makes use of the HAN, NAN and WAN for metering-related functions. An overview of the AMI communication scheme is shown in Fig. 1.2.

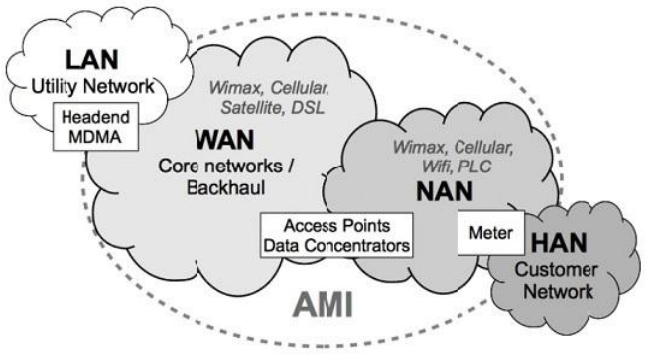

Figure 1.2: Overview of AMI networks adopted from [16].

Home Area Network (HAN)

HAN consists of intelligent the home/building devices which makes it possible to monitor and control the electrical devices at home. The intelligent appliances at customer premises communicate with customers' gateways or directly with smart meters in residential and industrial areas. They can react to the price signal they get from the utility during peak hours.

Neighborhood Area Network (NAN)

This network embraces all the data coming from smart meters and sends it to the WAN. NAN deployment provides the opportunity for utility companies to control end user devices, send real time commands, and control the distribution grid devices. As NAN is the main focus of this work, we will elaborate NAN communications and functionalities in more details in Chapter 3.

Wide Area Network (WAN)

WAN is a multipurpose network that provides connectivity from data collectors to control units in the utility center. It connects multiple substations and local control points back to the main

utility center. It forms a communication backbone to connect the utility centers to the highly distributed substations or customers' endpoints.

1.3 Communication Technologies for Smart Grid

Different communication technologies are being suggested for use in smart grid. In this section we will discuss some of these technologies along with their pros and cons.

1.3.1 PLC: Power Line Carrier

Electrical power is transmitted over high voltage transmission lines, distributed over medium voltage, and used inside buildings at lower voltages. Power line communications can be applied at each stage. Most PLC technologies limit themselves to one set of wires (for example, premises wiring), but some can cross between two levels (for example, both the distribution network and premises wiring). Typically the transformer prevents propagating the signal, which requires multiple PLC technologies to be used to form very large networks.

1.3.2 Fiber Optic

The optical network is known as passive optical network (PON). The basic configuration of a PON connects the central office (CO) to businesses and residential users by using one wavelength channel in the downstream direction from optical line terminal (OLT) at the CO to optical network units (ONUs). It utilizes another wavelength channel in the upstream direction from ONUs to OLT. A PON provides much higher bandwidth for data applications than current solutions such as digital subscriber line (DSL) and cable modem (CM), as well as deeper fiber penetration. Based on current standards, a PON can cover a maximum distance of 20 km from the OLT to the ONU. While fibre-to-the-building, fibre-to-the-home (FTTH), or even fibre-to-the-PC solutions has the ultimate goal of fibre reaching all the way to end-user premises, fibre-to-the-curb may be a more realistic deployment scenario today. Wired communication network build costs vary from $500-2{,}000$ per customer depending on density of customers covered according to Torotno Hydro Electric System Limited [73]. As such, they are not typically economically viable for smart grid. However, they can still be considered because they can often be leased from wire line communications carriers where they exist.

1.3.3 Cellular Technologies: GSM/GPRS/CDMA/LTE

Global System for Mobile (GSM) is an ETSI standard for 2G pan-European digital cellular with international roaming. It provides capabilities to transmit information among user-network interfaces. Traditionally it includes data access to PSTN/ISDN and packet switched data network for asynchronous data transmission, the data rate is up to 300-9600 bps (transparent/non-transparent). And to transmit synchronous data, the data rate is 2400-9600 bps transparent. Also GSM is capable of synchronous packet data transmission. GPRS (General Packet Radio Service) is a packet switching technology, using free slots only if data packets ready to send (e.g., 50 kbit/s using 4 slots temporarily). It provides network service including PTP (point-to-point) connection oriented, PTP connectionless, and PTMP (point-to-multi-point). The advantage is one step towards UMTS, more flexible but more investment needed (new hardware).

CDMA technologies, for example, CDMA 2000 is implemented in DS-CDMA, FDD and TDD modes, Which is Backward compatibility with IS-95A and IS-95B and uses auxiliary carriers to help with Downlink channel estimation in forward link beam forming. WCDMA is also implemented in DS-CDMA, FDD and TDD modes. It is backward compatibility with GSM/DCS 1900. The data rate is up to 2.048Mbps on downlink in FDD mode. It uses pilot bits assist in downlink beam forming, as well provides High Speed Downlink Packet Access (HSDPA).

LTE has been defined by the third generation partnership project (3GPP). The first amendment of LTE (release 8) provides a transmission rate of 300 Mbit/s and operates in both time divisions duplex (TDD) and frequency division duplex (FDD) modes. It provides peak data rates of 100 Mbps in downlink and 50Mbps in uplink within a 20MHz bandwidth or, equivalently, spectral efficiency values of 5bps/Hz and 2.5bps/Hz, respectively. The LTE aims at providing a smooth evolution from earlier 3GPP and 3GPP2 cellular networks such as wide-band code division multiple access/high-speed packets access (WCDMA/HSPA) and code division multiple access (CDMA2000) [84]. In 2008, LTE-Advanced (also known as requirements for further advancements for evolved-universal terrestrial radio access) was initiated to enhance LTE radio access in terms of system performance and capabilities [73].

The cellular carriers got into the market first with their 2.xG/3G data services, but their offerings are positioned as an add-on to what is essentially a voice service. The real challenge to the cellular data services will come from the two emerging data oriented technologies, WIMAX

and Wi-Fi. With chip level components due for shipment in recent years, infrastructures for Wi-Fi and WiMAX are more feasible and mature. The 2G/3G cellular technologies are widely deployed and open standard; however they have limited capacity and limited service life.

1.3.4 WiFi/WiFi mesh

Wireless LANs based on the IEEE 802.11 (or WiFi) standards have been a resounding success. While Wi-Fi has virtually obliterated all other contenders in the local area, the wide area market is still up for grabs. IEEE 802.11g standard for WLAN, also known as WiFi, operates in the 2.4 GHz unlicensed band with the maximum data rate of 54 Mbit/s. WiFi uses orthogonal frequency division multiplexing (OFDM) modulation scheme for data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s. For data rates of 5.5 and 11 Mbit/s, it uses complementary code keying (CCK), and for data rates of 1 and 2 Mbit/s it uses direct-sequence spread spectrum (DSSS) modulation scheme. Moreover, Depending on application Wireless mesh can be used to provide extensibility and self healing features.
WiFi mesh network is a multi-hop network, which does not usually require sophisticated planning, and hence provides a reliable flexible network in which wireless devices assist each other in transmitting their packets throughout the network. A wireless mesh network can be easily scaled and deployed quickly. Each packet can reach its destination from different paths offering redundancy for the network. This type of network is one of the leading AMI communication solutions in North America. Each smart meter acts as a relay node and will forward messages to the next smart meter. As Wireless mesh network is of main interset of our IDS, we will elaborate it more in Chapter 2.

1.3.5 WiMAX

WiMax is a broadband wireless access that supports both fixed and mobile Internet access. It is based on IEEE 802.16 and has maximum data rate of 75Mbits/sec under optimal conditions. WiMax range covers up to several kilometers. As a result it can be used for providing wireless broadband across to cities and countries. It can be used as an alternative last mile solution to cable and DSL. WiMax uses scalable orthogonal frequency-division multiple access (SOFDMA) with 256 sub-carriers. It also supports multiple antennas for better coverage and better power consumption. Medium access control (MAC) layer of WiMax uses a schedul-

ing algorithm for the initial entry of the subscriber stations (SS) into the network. Then the base station (BS) allocates an access slot to SS and other subscribers cannot use that slot. The scheduling algorithm is also used for controlling the bandwidth efficiency and quality of service (QoS) parameters by changing the time slot duration based on the SSs application needs. WiMax uses 2.3GHz, 2.5GHz and 3.5GHz licensed bands. IMT-2000 standards are defined by the radio communication sector of the International Telecommunication Union (ITU-R). As a result any country that recognizes IMT- 2000 standards is able to use WiMax equipment. The IEEE 802.16e (known as Mobile WiMax) standard offers scalability in both radio access technology and network architecture. While the spectrum allocation is applied as a radio access technology of Mobile WiMax, its flexibility in network deployment provides various services [46].

Based on above discussion, Fig. 1.3 represents the technologies that can be used in HAN, NAN, and WAN given their specific requirements.

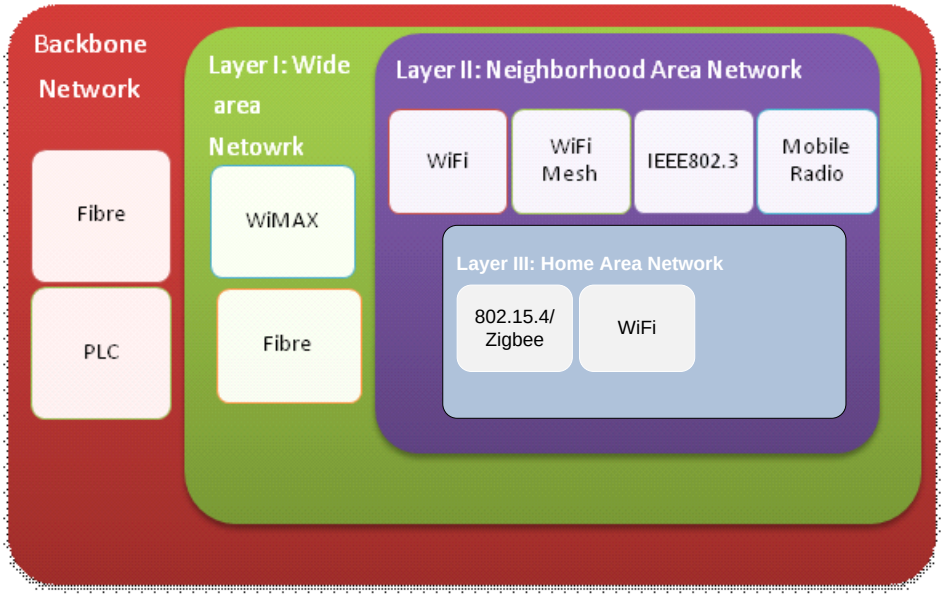

Figure 1.3: Communication technologies used for AMI.

1.4 Smart Grid Communication Requirements

In smart grid, communication networks plays a significant role and are required to be highly reliable. All the nodes in the network should be available under all circumstances and the network must be robust. The communication network should have a high coverage to connect the highly distributed nodes in the smart grid realm. Communication overhead is another issue in the smart grid; even though the commands and data packets are usually short, the total volume to be transferred is overwhelming.

Different smart grid applications have different QoS requirements. Some of these applications e.g., control and alarming data are delay sensitive and have some Quality-of-Service (QoS) requirements[63] and are loss-intolerant. Applications like substation and feeder SCADA (Supervisory Control and Data Acquisition) data, meter reading, longer term market pricing information, and collecting long term data such as power quality information have time latency of seconds, hours, days/weeks/months, respectively [28].
Easy deployment and maintenance is essential for any distributed network and smart grid is no exception. As National Institute of Standards and Technology (NIST) indicates the use of current technology and adaptation is one of the acceptable strategies that can be employed in smart grid communications [28].

According to the Electric Power Research Institute (EPRI), security is one of the biggest challenges for widespread deployment of smart grid [55]. The two-way communication path that monitors and controls the smart grid infrastructure is a potential opportunity for knowledgeable attackers. Physically unprotected entry points as well as wireless networks that can be easily monitored and possibly interfered pave the path for attackers. Hence, there should be security mechanisms in place intended to prevent unauthorized use of these communication paths. Based on the history of security in other types of networks, many risks exist and are yet to be discovered. Many of the technologies being deployed into smart grid such as smart meters, sensors, and advanced communication networks can increase the vulnerability of the grid to cyber attacks and the risk grows as the deployment becomes more widespread. Communication security is an absolute requirement in the smart grid; smart grid deployments will fail without proper cyber security mechanisms built in. Security mechanisms must address not only deliberate attacks but also inadvertent ones due to user errors, equipment failures, and natural disasters.

1.5 Smart grid Security

Smart grid AMI brings on new security challenges since it is composed of the devices that are placed in physically insecure locations and it makes use of wireless communication that can be possibly corrupted. These resources can be accessed by careless or malicious users [50],[49]. Cleveland in [21] discusses the security requirements and related threats of the main components of an AMI. Security concerns for AMI can be classified into :

- **Confidentiality** and in particular, privacy, which can strongly affect the customers' view of deploying smart grid. Customers might not like unauthorized people, or companies to know about their usage patterns. Usage patterns also can reveal life habits and even the presence/absence of residents that could be used by thieves. If people's concerns are not satisfied, they may refuse to cooperate in deploying smart grid, i.e., they may refuse to let the utility providers install smart meters at their places.

- **Integrity** in AMI systems means preventing any changes in the metering data received form meters and control commands sent to the meters. One of the scenarios that may happen is when a hacker sends disconnect command by breaching into a meter management system and disconnect millions of smart meters.

- **Availability** is considered as the most crucial requirement in AMI since some systems or applications are real time and they possibly deal with the availability of power.

- **Non-repudiation** is also needed since different entities are involved in financial transactions, owning data and even generating control commands. Audit logs of interactions are mainly used for non-repudiation, although these logs can be affected by integrity and availability attacks.

In AMI, availability and integrity of data take precedence over confidentiality [50], [83].

Attacks targeting AMI can be classified into three categories: network compromise, system compromise and denial of service (DoS) [16]. Traffic modification, false data injection, replay and traffic analysis attacks try to compromise the network [48] while compromised node and spoofing of metering devices, authentication violation and access to encryption keys are examples of attacks which target the systems [45]. Flaws or misuses of routing, configuration, name resolution and signal jamming are considered as DoS attacks.

1.6 Intrusion Detection System (IDS)

AMI requires a reliable monitoring solution so that in case of any security breaches, the grid can detect or deter the violation. Intrusion detection system (IDS) acts as a second wall of defense and is necessary for protecting AMI if security mechanisms such as encryption/decryption, authentication and etc. are broken.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices [64]. Generally, techniques for intrusion detection are classified into three main categories which are explained below:

- Signature-based: which relies on a pre-defined set of patterns to identify attacks. It compares known threat signatures to observed events to identify incidents. This is very effective at detecting known threats but largely ineffective at detecting unknown threats and many variants on known threats. Signature-based detection cannot track and understand the state of complex communications, so it cannot detect most attacks that comprise multiple events.

- Anomaly-based which relies on particular models of nodes behaviors and mark nodes that deviate from these models as malicious; It compares definitions of what activity is considered normal against observed events to identify significant deviations. This method uses profiles that are developed by monitoring the characteristics of typical activity over a period of time. The IDS then compares the characteristics of current activity to thresholds related to the profile.

- Specification-based which relies on a set of constraints and monitor the execution of programs/protocols with respect to these constraints [17].

The performance of IDS is evaluated based on three main measures including:

- False positive (FP): An event signaling an IDS to produce an alarm when there is no attack taken place. The formula by which FP is calculated is :

$$FP = \frac{\text{Number of normal patterns detected as attack}}{\text{Number of all normal patterns in the network}}$$

- False negative (FN): A failure of the IDS to detect an actual attack. FN is calculated using below formula:

$$FN = \frac{\text{Number of attacks not detected by IDS}}{\text{Number of attacks in the network}}$$

- Detection Rate: The ability of IDS to detect all the existing attacks and is calculated by

$$DR = \frac{\text{Number of detected attacks}}{\text{Totall number of attacks targeting the network}}$$

1.7 Related Work

In [48], authors have evaluated the security threats on the communication network in the smart gird. They compare the smart grid with Internet and highlight the critical differences such as performance metric, traffic model and so forth. Since TCP/IP is widely, if not all, used in smart grid, the studies of DoS attacks against TCP/IP such as [65], [82] can be used as a starting point for the analysis of the smart grid as well.

Authors in [51] argue for use of Public Key Infrastructure (PKI) as the best overall security solution for smart grid. They believe that "in very large systems, PKI could be significantly more efficient than shared keys in terms of setting up and maintaining operational credential". In the same category, [69] proposes the use of an identity-based signature and encryption scheme in order to off-load the senders of messages as much as possible. The machine identity number (ID) of the device connected in a smart grid is used to generate unique keys to encrypt and sign each individual data packet sent among devices in the grid.

Authors in [8] present an authentication and encryption/decryption scheme for HAN. The scheme relies on the transceivers that are attached to power outlets which communicate with HAN's meter. The scheme uses PKI for securing the communication between appliances and HAN's meter. Various attacks against HAN security such as jamming, appliance impersonation, replay and non-repudiation attack have also been studied.

A technique for evaluating the security of devices being deployed into the AMI is investigated in [50]. Authors develop an attack tree approach to guide penetration testing of multi-vendor technology classes and design a penetration testing process. This work provides a comprehensive but rather high-level classification of three types of attacks targeting AMI including

denial of service, energy fraud and targeted disconnect.

While efforts have been made to investigate the security of AMI, there are a few works that focus on proposing and designing a reliable and efficient IDS for AMI. Authors in [16] discuss the requirements and practical needs for monitoring and intrusion detection in AMI. The research done in the area of smart gird IDS and the key functional requirements of an IDS for smart grid environment have been surveyed in [42]. In [37], authors present a layered combined signature and anomaly-based IDS for HAN. Their IDS is designed for a ZigBee-based HAN which works at the physical and medium access control (MAC) layers.

In [15] a specification-based IDS for AMI is proposed. While the solution in [15] relies on protocol specifications, security requirements and security policies to detect security violations, it would be expensive to deploy such an IDS since it uses a separate sensor network to monitor the AMI.

In [60] authors propose a model-based IDS working on top of the WirelessHART protocol, which is an open wireless communication standard designed to address the industrial plant application, to monitor and protect wireless process control systems. The hybrid architecture consists of a central component that collects information periodically from distributed field sensors. Their IDS monitors physical, data link and network layer to detect malicious behavior. Although a detailed explanation of [60] has been provided, it is protocol specific and might not be applied to AMI. Authors in [80] investigate the use of wireless mesh network (WMN) and the security framework for distribution network in smart grid. A response mechanism for meter network has also been proposed.

In this thesis, we design and implement an IDS for NAN part of AMI. The related works discussed above either are not specifically designed for the NAN or they require a separate network for detecting intrusions in the network. In our solution, however, we rely on the NAN own characteristics and propose an IDS which does not require extra nodes as monitoring agents. Depending on the type of attacks to be detected, we employ IDS on some nodes in the NAN which are powerful in terms of computation and communication capabilities. This IDS is specifically designed for detecting Wormhole attack. We have developed our solution in OPNET modeler 17.1 [56] by integrating an analytical model implemented in Maple 16[33] with the simulation model.

The organization of this thesis is as follows: in Chapter 2 we will discuss wireless mesh networks that are used as the NAN communication network. Chapter 3 elaborates on the NAN and its communication characteristics along with its security concerns. In Chapter 4,

we explain our IDS design steps in OPNET. Simulation scenarios and the analytical model are presented in Chapter 5. The performance of Our IDS is illustrated at the end of Chapter 5. We conclude the thesis and state the future work in Chapter 6.

Chapter 2

Wireless Mesh Networks: Architecture and Security Challenges

A mesh network is configuration of peer wireless access nodes that allow for continuous connections to a network infrastructure, including reconfiguration around blocked paths, by "hopping" from node to node [27]. The term "mesh network is often used synonymously with wireless ad-hoc network. Ad-hoc networks have low commercial penetration since they deal with specialized applications. In order to turn ad-hoc networks into commodity, multi-hop ad-hoc networks should not exist as isolated self-configured networks. Instead they should be low-cost extension of wired infrastructure and coexist with them in order to enable low cost Internet access.

Wireless mesh networks (WMNs) are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad-hoc network and maintaining the mesh connectivity. They consist of mesh routers and mesh clients. Each router is responsible for setting up the ad-hoc network and maintains mesh connections with other routers within its transmission range. Therefore, through multi-hop communications, the same coverage as mobile ad-hoc network's (MANET's) can be achieved by mesh routers with much lower transmission power. A mesh router is usually equipped with multiple wireless interfaces built on either the same or different wireless access technologies. Fig. 2.1 presents the position of WMNs in wireless networks [85].

There are a variety of applications for WMNs including : intelligent transportation systems, public safety, public broadband wireless Internet access for urban, sub-urban and rural areas

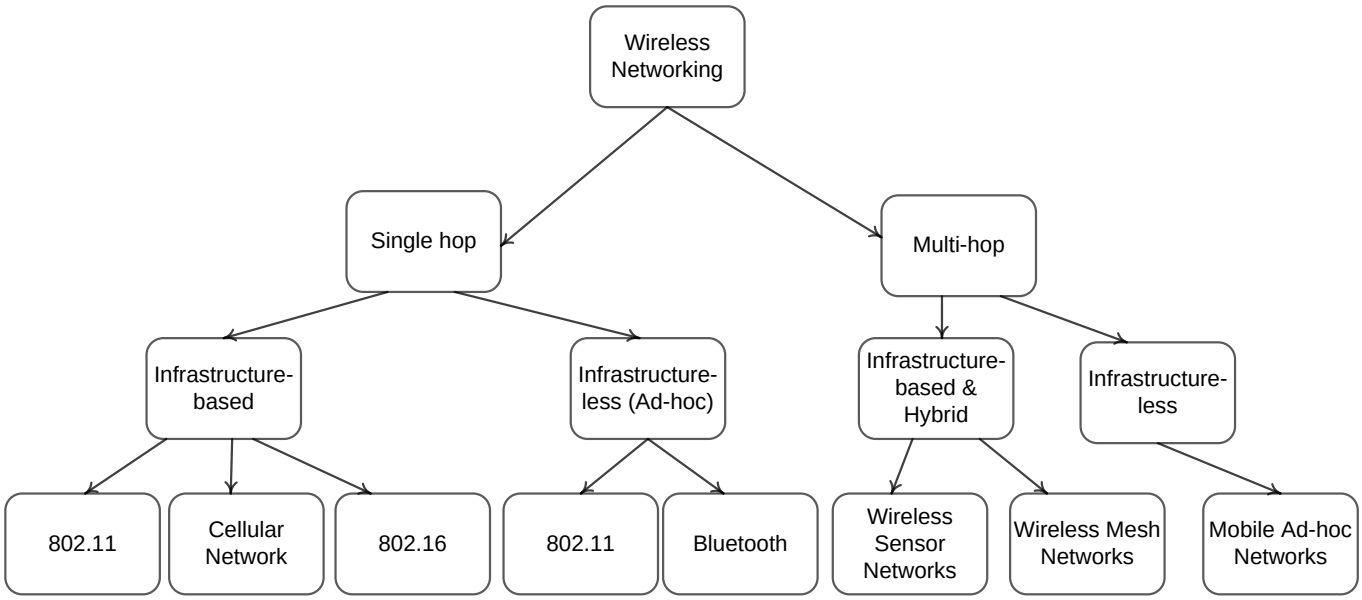

Figure 2.1: Wireless networking

and etc. WMNs should support following characteristics:

- Ad-hoc routing protocol is implemented in mesh routers to work together with 802.11 MAC. Certain radio aware functions may be included in the routing protocol.
- MAC driver is enhanced in mesh routers to improve multi-hop performance. Typical examples include fine tuning CSMA/CA parameters, developing algorithms for multi-radio or directional antennas, etc [61].
- Scalability shoud be supported in WMNs; without scalability the network performance desegregate significantly as the network size increases [62].
- Security is a vital issue in the design of WMN due to the shared medium and multi hop communications. Thus, WMN communications should be secure against internal and external attacks [77].

Architecture

WMN architecture can be in the forms of infrastructure-based, client-based and hybrid-based. In an infrastructure WMN, routers form an infrastructure for clients as shown in Fig 2.2. Client

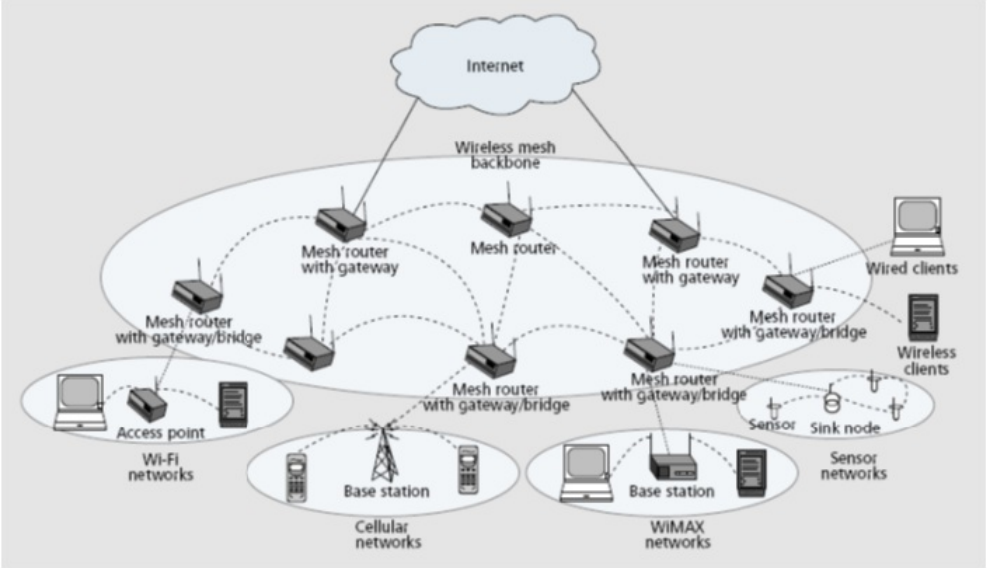

Figure 2.2: Infrastructured WMN [6]

meshing provides peer-to-peer networks among client devices very similar to ad-hoc networks. Client nodes constitute the actual network to perform routing and configuration functionalities as well as providing end-user applications to customers. The hybrid WMN is a combination of infrastructure and client meshing in which both clients and routers are capable of routing. Fig. 2.3 depicts a typical hybrid WMN.

Topology Discovery and Link State

Mesh Points (MPs) that are not yet members of the mesh must first perform neighbor discovery to connect to the network. A node scans neighboring nodes for beacons which contain at least one matching profile, where a profile consists of a mesh ID, path selection protocol identifier, and link metric identifier [19]. If the beacon contains a mesh capacity element that contains a nonzero peer link value then the link can be established through a secure protocol [36].

Routing and path selection

As WMNs are an extension of ad-hoc networks, therefore they should benefit from routing algorithms developed for MANETs. Ad-hoc routing protocols are mainly categorized into:

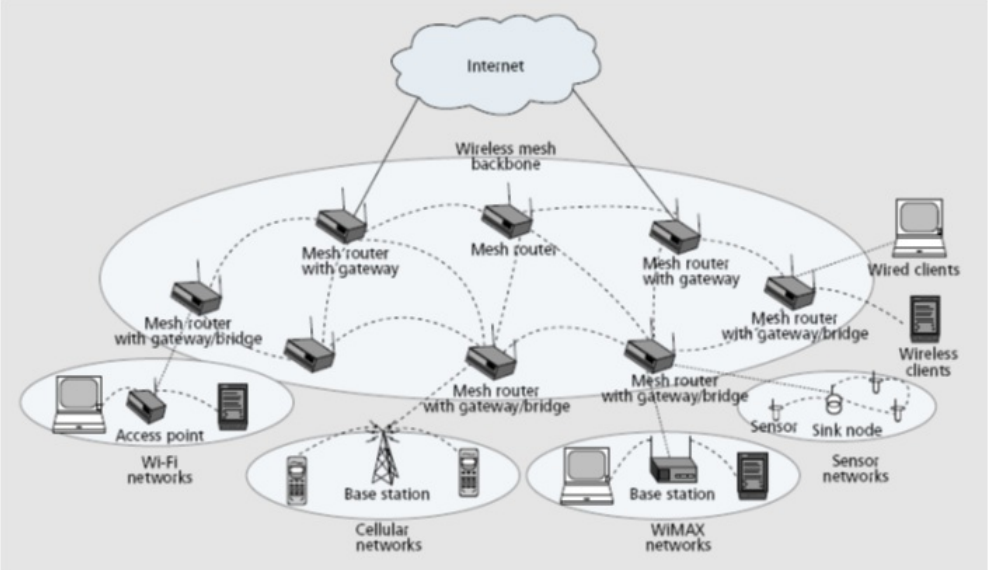

Figure 2.3: Hybrid WMN [6]

- Proactive
- Reactive

Proactive routing approach is based on traditional distance- vector and link- state protocols and each node maintains route to each other. Nodes periodically and/or based on events, exchange routing updates. In this type of routing, we may have higher overhead in most scenarios and longer route convergence time. Example of such approach includes destination-sequenced distance-vector routing (DSDVR), topology broadcast based on reverse-path forwarding (TBRPF) and optimized link state routing protocol (OLSR).

In reactive routing, the nodes build routes on- demand by "flooding through a rout discovery clycle. Nodes maintain only active routes. Typically reactive approach has less control overhead, better scaling properties but it has route acquisition latency. Ad-hoc on demand distance vector routing algorithm (AODV) and dynamic source routing (DSR) are examples of reactive routing protocols.

Ad-hoc on-demand distance vector routing (AODV)

AODV routing protocol builds on the DSDV algorithm and is used as a popular routing protocol in WMNs. AODV creates routes on demand basis by minimizing the number of required broadcasts unlike the DSDV algorithm which maintains a list of routes and is an improvement

to DSDV algorithm. AODV is also called as a pure on-demand routing acquisition system because only the nodes which are on the selected path maintain the routing information are involved in the routing table exchange.

When a source node wants to communicate with some destination node but does not have a valid route to that destination, the source node initiates a path discovery process to discover the other node. To achieve this, source node broadcasts Route Request (RREQ) packet to its neighbors. This request is in turn sent to its neighbors until either the destination route or an intermediate node route to the destination is traced.

To ensure all routes are loop free and contains the most recent route information, a destination sequence number is being maintained by AODV. Each node along with the broadcast ID maintains its own sequence number. For every RREQ that the node initiates, the broadcast ID is incremented and together with the nodes IP address, the RREQ is uniquely identified. The source node along with its own sequence number and the broadcast ID includes the most recent sequence number it has for the destination to the RREQ. Intermediate nodes reply to the RREQ only if they have a route to the destination and also the destination sequence number should be greater than or equal to the current destination sequence number contained in the RREQ.

The intermediate nodes records the address of the neighbors from which the first copy of the broadcast packet is received in their route tables. This helps in establishing a reverse path. If nodes later receive additional copies of same RREQ, then such packets are discarded. Once the RREQ reaches the destination or an intermediate node with a new route to the destination, the destination or intermediate node replies by sending a Route Reply (RREP) packet back from which it first received the RREQ packet.

Fig. 2.4 shows the steps performed in route discovery in AODV. In Fig. 2.4 node S wants to route its packets to destination D. By sending a RREQ to its neighbors, it starts discovering possible routes to D. Based on RRREP received from its neighbor, it will send its data packets to destination D through the best discovered path.

It should be noted that in AODV if a node receives more than one RREQ with same source IP and Broadcast ID, it will drop all except the first one. Moreover, a node that receives more than one RREP for one destination with the same sequence number may forward the new one if it has smaller hop count.

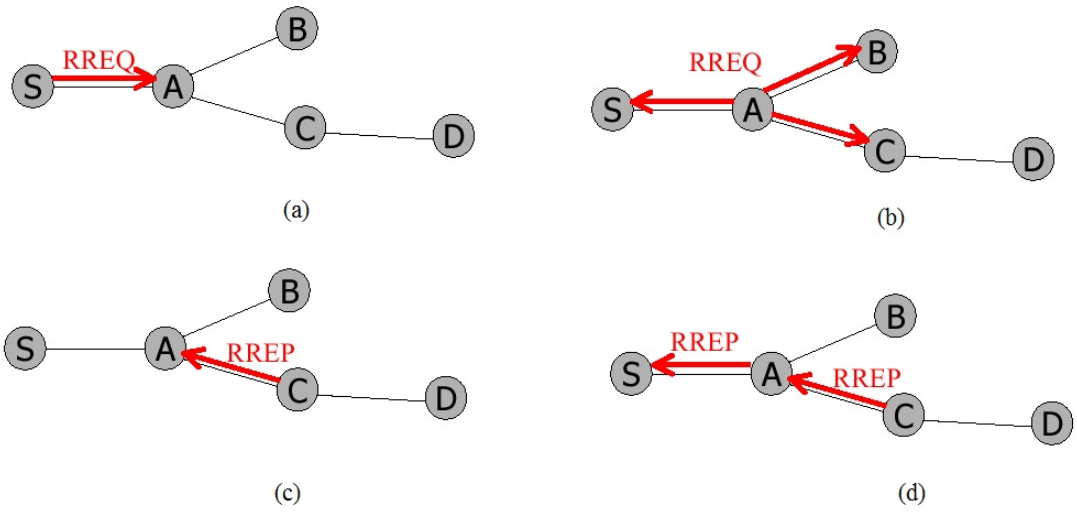

Figure 2.4: AODV routing discovery steps [58]

2.1 Security Challenges in WMNs

The broadcast nature of transmission and the dependency on the intermediate nodes for routing the user traffic leads to security vulnerabilities making WMNs prone to various attacks [53]. The attacks can be external as well as internal in nature. External attacks are launched by attackers who are not part of the WMN and gain illegitimate access to the network. For example, an intruding node may eavesdrop the packets and replay those packets at a later stage of time to gain access to the network resources. Attacks from external nodes can be prevented by resorting to cryptographic techniques such as encryption and authentication.

On the other hand, the internal attacks are launched by the nodes that are part of the WMN. One example of such attack is an intermediate node dropping the packets, which it was supposed to forward, leading to a denial of service attack (DoS). Similarly, the intermediate node may keep the copy of all the data that it forwards (internal eavesdropping) for offline processing and meaningful information retrieval without the knowledge of any other node in the network. Such attacks are typically launched either by selfish nodes or by malicious nodes, which may have been possibly compromised by attackers. There is a subtle difference in their motives. The selfish node is seeking to greedily acquire greater than its fair share of the network resources at the expense of other users. On the contrary a malicious attacker's sole aim is to undermine the performance of the entire network. Note that in an internal attack, the misbehaving node is

part of the WMN and hence has access to all the keying and authentication information [53].

DoS category is one of the most important attacks that can target WMNs and degrades their performance drastically. In following sections, we describe possible DoS attacks that can occur in WMNs considering different OSI layers.

2.1.1 Physical layer

Jamming signals in the form of continuous or periodic noise is generated to disrupt the transmission of bits in the physical layer. The Jamming signal can be reactive in which case it intercepts the channel only when an ongoing transaction is detected and disrupts the transmission. The jamming signal can significantly reduce the capacity of the channel [67]. Scrambling is a special kind of jamming attack where the attacker scrambles a few selected frames or part of frames. The targeted frames are usually management or control frames and the attack targets on disrupting the normal operation of the network [79].

2.1.2 MAC layer

A secure MAC layer is responsible for ensuring that a mesh network carries traffic only for authorized stations, thus preventing attacks by unauthorized ones. Attacks that can target MAC layer are discussed below.

Link layer jamming attack

In this attack the jammer instead of transmitting random bits constantly as radio jamming in physical layer, it transmits regular MAC frame headers without any payload which conform to the MAC protocol. As a result, the legitimate nodes will sense the channel busy and back off for random period of time. This prevents the nodes from transmitting their legal frames, therefor, reducing the network performance.

MAC Spoofing attack (Man in the middle attack)

Modifying the MAC address in transmitted frames is called MAC spoofing and can be used to launch DoS attacks. Networks administrators often use MAC addresses in access control lists (ACLs) so that only registered MAC addresses are allowed to connect to the network. An

attacker can easily eavesdrop on the network to determine legitimate MAC addresses. Once it gains a legitimate MAC address, it can use it to launch DoS attacks. An example of such attacks is where the attacker injects a large number of bogus frames in to the network to deplete the network's resources.

2.1.3 Network layer

Attacks targeting network layer can be divided into two categories: control plane attacks and data plane attacks [85], [67]. In the former, the attacker interferes with the proper functionalities of the routing protocol and the latter prevents proper forwarding of data packets.

Control plane attack

In control plane attacks, the attacker tries to make the routes unavailable or control the routing path by targeting the routing fucntionalitites.

Rushing attack

To intrude into the forwarding group the attacker suppresses the flooding of RREQ packets from legitimate nodes. The main goal of this attack is to suppress valid paths from being established or to increase its chance of it being part of the optimal path selected. It rushes its RREQ packet to its neighbor nodes before any other legitimate nodes can broadcast their RREQ packets. As a result, the nodes processes or forward only the first RREQ packet it receives and drops the others. This chain continues and the packet is broadcasted across the network increasing the rushing nod's chance of being a part of the selected route [54] , [44].

Wormhole attack

During a wormhole attack, two or more malicious nodes collude together by establishing a tunnel (wormhole link) using an efficient medium (i.e., wired connection or high speed wireless connection). Once the wormhole link is established, the adversary captures wireless transmissions on one end, sends them through the wormhole link and replays them at the other end.

An example of wormhole attack is shown in Fig 2.5. Here X and Y are the two end-points of the wormhole link. X replays in its neighborhood (in area A) everything that Y hears in its own neighborhood (area B) and vice versa. The net effect of such an attack is that all the nodes

in area A assume that nodes in area B are their neighbors and vice versa. This, as a result, affects routing and other connectivity based protocols in the network. Once the new routes are established and the traffic in the network starts using the X-Y shortcut, the wormhole nodes can start dropping packets and cause network disruption. They can also spy on the packets going through and use the large amount of collected information to break any network security [13].

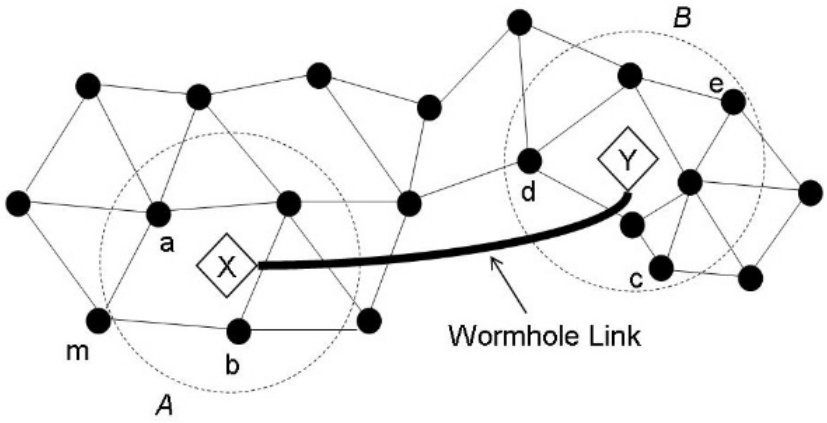

Figure 2.5: An example of a Wormhole attack [1]

Black hole and Gray hole attacks

The malicious node upon receiving RREQ it claims that it has the best route to the destination even though it does not have any valid routes to the destination. Since the malicious node does not check its routing entries, it is the first one to reply the RREQ messages. Thus, almost all traffic within the neighborhood of the malicious node will be directed towards the malicious node which may drop all the packets. Gray hole is a variant of black whole in which the attacker drops packets selectively.

Data forwarding plane attack

Selfish and compromised nodes can cause data plane attacks and result in performance degradation of the network. As nodes in WMN are greatly dependent on each other for forwarding data packets, selfish behavior can be a major security issue in the WMNs. While selfish node

may drop all or some of the packets, the malicious node may inject junk packets into the network consuming network's resources such as bandwidth.

In order to prevent attacks targeting routing protocol in WMN, secure routing protocols that have been proposed in MANET can be adopted for WMNs; Authenticated routing for ad-hoc networks (ARAN) which is suitable for on-demand routing protocols and uses public key certificate and trusted CA to verify the routing information have already been implemented for MANETs; Secure efficient ad-hoc distance vector (SEAD) uses hash chains to detect tampering of routing information within the network; secure ad-hoc on-demand distance vector (SAODV) which employs two mechanisms to ensure the security of AODV. One is the digital signature with ensure the integrity of data that does not need to be modified while forwarding. The other one is the one-way has chain to verify mutable parts such as number of hops in the packet [85].

A key challenge will be their adoption to WMNs as new metrics and security designs will be needed for WMNs in which the use of multiple radios alters some of the basic assumptions in such approaches [13].

2.1.4 Transport Layer

Data injection

Data injection attack is another attacks that can occur at TCP layer in which a compromised node tries to exhaust the bandwidth as well as the resources of its neighbors (e.g., memory and processing power). Table 2.1 lists the attacks targeting different OSI layers.

Table 2.1: Wireless mesh network attacks.

Attack	Layer
Physical Layer	Signal jamming
MAC Layer	Link layer jamming, MAC Spoofing
Network Layer	Rushing, Wormhole, Black hole, Gray hole, Packet dropping
Transport Layer	SYN flooding, Data injection

2.2 Authentication and key distribution mechanisms in WMNs

Initially, security schemes developed for other wireless networks have been used for WMNs. IEEE 802.11i security schema, i.e., Robust Security Network Association (RSNA), are adopted

for use in WMNs while in Fall 2011, IEEE 802.11s task group published a security framework for 802.11s WMN called Simultaneous Authentication Equals (SAE) which is based on zero knowledge proof [29].

The security framework for 802.11i, RSNA-based Security Framework of 802.11s WMN and, SAE are discussed in following sections.

2.2.1 802.11i security framework

Although the IEEE 802.11 standard upon which Wi-Fi wireless LAN networks are based addressed somewhat security with the Wired Equivalent Privacy (WEP) protocol in its first instantiation, WEP proved relatively easy to crack. The IEEE 802.11 group became aware of the issues with WEP early on and IEEE Standards Association approved an amendment to the original IEEE 802.11 specification that addresses these issues. The amendment adds stronger encryption, authentication, and key management strategies that go a long way toward guaranteeing data and system security.

The resulting IEEE 802.11i amendment has many components, the most obvious of which are the two new data-confidentiality protocols, temporal key integrity protocol (TKIP) and counter cipher mode with block chaining message authentication code protocol (CCMP). IEEE 802.11i also uses IEEE 802.1X's key-distribution system to control access to the network. Because IEEE 802.11 handles unicast and broadcast traffic differently, each traffic type has different security concerns. With several data-confidentiality protocols and the key distribution, IEEE 802.11i includes a negotiation process for selecting the correct confidentiality protocol and key system for each traffic type. Other features introduced include key caching and pre-authentication.

Wi-Fi Protected Access (WPA)

The Wi-Fi Alliance intended WPA as an intermediate measure to take the place of WEP using TKIP. TKIP is a data confidentiality protocol that was designed to improve the security of products that implemented WEP, in other words, legacy products. Among WEP's numerous flaws are its lack of a message integrity code and its insecure data-confidentiality protocol. To get around these limitations, TKIP uses a message integrity code called Michael. Basically, Michael enables devices to authenticate that the packets are coming from the claimed source. This authentication is especially important in a wireless technology where traffic can be easily injected.

TKIP uses a mixing function to defeat weak-key attacks, which enabled attackers to decrypt traffic. Since the decryption could be done passively, it meant that an attacker could watch WEP traffic from a distance, be undetected, and know the original traffic. TKIP fixes this situation by using a mixing function. The mixing function creates a per-frame key to avoid the weaknesses pointed out by Fluhrer, Mantin, and Shamir [].

WPA2

WPA2 has replaced WPA and introduces CCMP, an Advanced Encryption Standard (AES)-based encryption mode with strong security. Although WPA improves security especially for legacy hardware, a stronger alternative was needed for newer hardware.

CCMP is a data-confidentiality protocol that handles packet authentication as well as encryption. For confidentiality, CCMP uses AES in counter mode. For authentication and integrity, CCMP uses Cipher Block Chaining Message Authentication Code (CBC-MAC). In IEEE 802.11i, CCMP uses a 128-bit key. The block size is 128 bits. The CBC-MAC size is 8 octets, and the nonce size is 48 bits. There are two bytes of IEEE 802.11 overhead. The CBC-MAC, the nonce, and the IEEE 802.11 overhead make the CCMP packet 16 octets larger than an unencrypted IEEE 802.11 packet. Although slightly slower, the larger packet is not a bad exchange for increased security.

CCMP protects some fields that are not encrypted. The additional parts of the IEEE 802.11 frame that get protected are known as additional authentication data (AAD). AAD includes the packets source and destination and protects against attackers replaying packets to different destinations. IEEE 802.11i phase of operation is shown in Fig 2.6.

IEEE 802.1X

IEEE 802.1X provides a framework to authenticate and authorize devices connecting to a network. It prohibits access to the network until such devices pass authentication. IEEE 802.1X also provides a framework to transmit key information between authenticator and supplicant. IEEE 802.1X has three main pieces as shown in Fig. 2.6:

- Supplicant
- Authenticator
- Authentication server

For IEEE 802.11i, the access point takes the role of the authenticator and the client acts as supplicant. The supplicant authenticates with the authentication server through the authenticator.

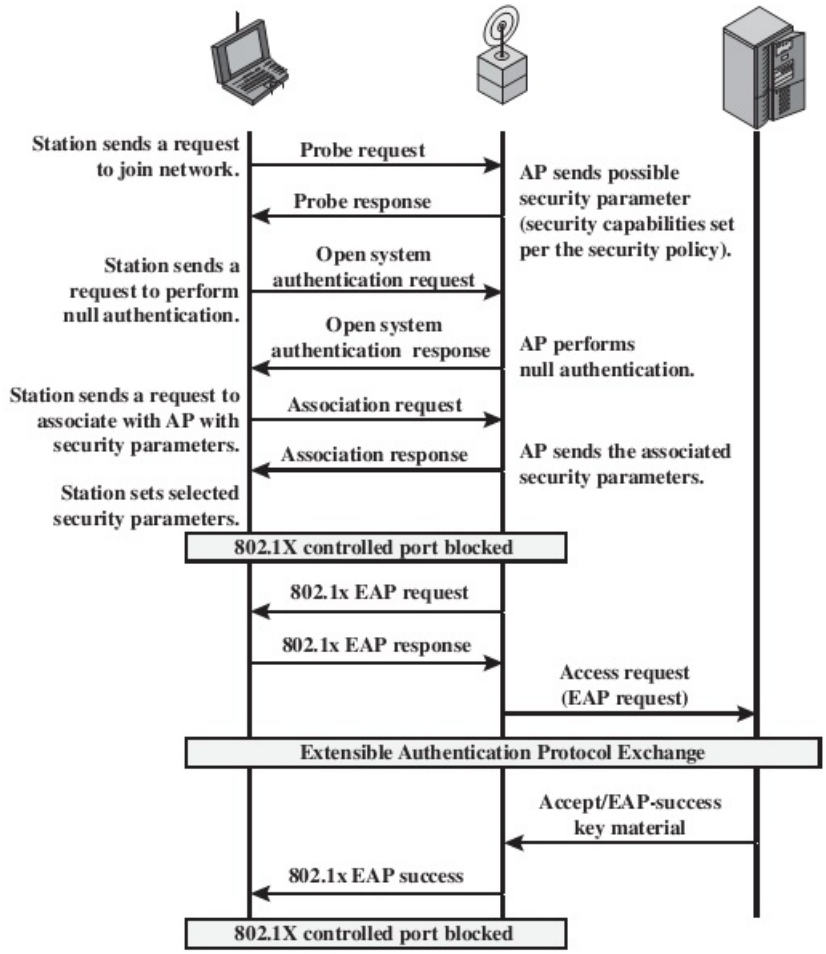

Figure 2.6: IEEE 802.11i phases of operation: capability discovery, authentication, and association [71]

In IEEE 802.1X, the authenticator enforces authentication. The authenticator does not need to do the authentication. Instead the authenticator exchanges the authentication traffic between the supplicant and the authentication server.

Between the supplicant and the authenticator, the protocol is IEEE 802.1X. The protocol between the authenticator and authentication server is not defined in IEEE 802.1X nor IEEE 802.11i. However, Remote Authentication Dial-In User Service (RADIUS) or DIAMETER is typically used between authenticator and authentication server.

The uncontrolled port is used to pass authentication traffic between the supplicant and the authentication server. Once the authentication server concludes authentication with the suppli-

cant, the authentication server informs the authenticator of the successful authentication and passes established keying material to the authenticator. At this point, the supplicant and the authenticator share established key material through an EAPOL-key exchange. (EAPOL, the Extensible Authentication Protocol over LANs) And if all exchanges have been successful, the authenticator allows traffic to flow through the controlled port, giving the client to access to the network.

Negotiation

Because IEEE 802.11i has more than one data-confidentiality protocol, IEEE 802.11i provides a way for the IEEE 802.11i client card and access point to negotiate which protocol to use during specific traffic circumstances and to discover any unknown security parameters. For instance, broadcast-key data-confidentiality protocol isn't negotiated. However, clients need to know that the protocol is in use. For instance, a client that's configured to run with CCMP may or may not be configured to associate with an access point that's using TKIP for its broadcast traffic. The access point advertises its parameters in beacons and will also reply to a probe request with a probe response containing the access point's security parameters. Some of the parameters that the access point advertises are:

- The group cipher suite is the data-confidentiality protocol used to send broadcasts
- The pairwise cipher suite list is a list of all data-confidentiality protocols allowed to be used for unicast traffic
- The authentication and key management suite advertises if pre-shared key or IEEE 802.1X is being used

Once the client knows these parameters, it chooses parameters and sends the choices in the associate request to the access point. The choices must match from the list of available options provided by the access point. Otherwise, the access point will deny the association by sending a association response failure. Up to this point, the negotiation isn't protected. But, the negotiation does get authenticated later during the EAPOL-key exchange.

Key hierarchy

The next step is key exchange. The IEEE 802.11i EAPOL-key exchange uses a number of keys and has a key hierarchy to divide up initial key material into useful keys. The two key hierarchies are:

- Pairwise key hierarchy

- Group key hierarchy

These keys get used in the EAPOL-key exchanges. In the IEEE 802.11i specification, these exchanges are referred to as the 4-way handshake and the group key handshake [71].

2.2.2 RSNA-based Security Framework of 802.11s WMN

The IEEE 802.11s draft standard uses efficient mesh security association (EMSA) to prevent unauthorized devices from sending and receiving traffic on the mesh, to both preserve resources and protect against malicious attacks. Like single-hop wireless LANs, EMSA uses the 802.11i link level authentication model, which includes 802.1X authentication, key distribution, and encryption of management frames. However, the key difference in security for mesh networks as opposed to traditional WLANs is that mesh APs must act in both authenticator and supplicant roles.

The 802.11s WMN security requires the RSNA functionality to be supported. Thus, pre-RSNA schemes such as WEP cannot be used.
The RSNA in 802.11s WMN is called mesh security association (MSA). Via MSA security functionalities similar to 802.1X are built into a distributed multi-hop WMN. There are two types of security key holders: mesh key distributor (MKD) and mesh authenticator (MA). A mesh point (MP) can be MKD and MA, MA, or neither. An MP with MA functionality plays the 802.1X authenticator's role and and MP without MA functionality plays the 802.1X supplicant's role. An MKD and MA can be co-located with MA and can manage authentication and key distribution for both MA and supplicant. In an 802.11s WMN, there exists one MKD, multiple MAs ,and supplicants. A supplicant can become a authenticator if it passes the security key holder association with MKD [5].

It should be noted that the 802.1X in MSA does not mean that 802.11s security needs an extra 802.1X authenticator server (AS)in the system. In fact, the MSA can act based on pre-shared key (PSK) mechanism in which case it does not need to have an AS. However, in order to enhance WMN security, an 802.1X As can be used. If this functionality is used, the entire security system actually consists of two 802.1X processes organized hierarchically: 802.1X AS and MKD at the upper level and MKD and MA are the lower level.

Whether using PSK or master session key (MSK) which is provided through successful authentication between the AS and the MDK, a security key hierarchy is established after an MP has passed the initial security authentication through an authenticator MP and MKD of the

mesh network. This MP's subsequent secure link setup with other MPs can be done directly. Therefor, some steps of authentication and key establishment can be omitted.

The key hierarchy consists of tow branches: link security branch and key distribution branch. the former is for generating keys for a secure link and the latter is for generating keys for key distribution. On the link security branch, pairwise master key (PMK) is first derived for MKD based on a PSK or a MSK. PSK is used when 802.1X authentication is not applied; otherwise, an MSK is provided through a successful authentication between the AS adn the supplicant MP. Based on PMK-MKD, PMK-MAs, are then derived. Key delivery and key management between the MKD and the MA are handled by mesh key transport and EAP message transport protocol. With a PMK-MA, an authenticator MP and its supplicant MP mutually derive a pairwise transient key (PTK). on the key distribution branch, a mesh key distribution key (MKDK) is first derived form PSK or MSK, and then a mesh PTK for key distribution (MPTK-KD) is derived mutually by an authenticator MP (after it becomes an MA) and the MKD.

In an 802.11s WMN, the support for MSA is advertised by MPs in the mesh security capability information element (MSCIE) of beacon or probe response frames. In addition to the mesh security capability field identified by the element ID and length field, MSCIE contains two other very important fields: MKD domain ID (MKDD-ID) and mesh security configuration.

There are several scenarios for selecting authenticator and supplicant for an MP and its peer MP:

- if only one MP that has already been an MA, i.e., its "connected to MKD"bit is one, then this MP is usually selected as an 802.1X authenticator , while the other one is an 802.1X supplicant.

- If both MPs have zero in "Connected to MKD" bit, then the MP with a larger MAC address or the selector MP is the 802.1X authenticator.

- If both MPs have one in the "Connected to MKD" bit, then the MP that requests authentication becomes the supplicant. Otherwise, if both request or neither requests authentication, then the selector MP is the 802.1X authenticator.

  With authentication role determined, MSA authentication is carried out in an MKD domain (MKDD). However, depending on whether this MP has established a secure link

with a peer MP before, the procedures of MSA authentication are different. If no such a secure link was setup before withing the same MKDD, a procedure of initial MSA authentication is needed to set up the mesh key hierarchy. Considering the requirement, and peer link management procedures, we know that usually an 802.1X supplicant needs an initial MSA authentication.

During initial MSA authentication, a mesh key hierarchy is created for the supplicant MP through the interactions among MKD, authenticator MP, and supplicant MP. If 802.1X authentication is needed, the authenticator MP will initiate the 802.1X authentication with the supplicant MP using EAPOL messages in 802.11 data frames (step 1.1 in Fig. 2.7). The 802.1X message may be transported between the MA and the MKD, so an EAP message transport protocol is defined between the MKD and the authenticator MP (step 1.2 in Fig 2.7). Upon successful completion of 802.1X authentication, the MKD receives the MSK and then generates PMK-MKD and PMK-MA. If no 802.1X authentication is needed, PSK is used to generate PMK-MKD and PMK-MA. Thus, steps 1.1 and 1.2 do not exist if PSK instead of 802.1X authentication is adopted in initial MSA authentication.

For 802.1X authenticatior MP, it can establish a mesh key holder security association with the MKD (step 2 in Fig. 2.7). After successful completion of this step , the authentiactor MP becomes an MA. In this step, encryption keys for security key distribution between MA and MKD are derived. It should be noted that this step may not be necessary if the authenticator MP has already been an MA and established a security association with MKD. Finally, the MKD delivers the PMK-MA to the MA using a mesh key transport protocol (step 3 in Fig.2.7).

After the above steps are done, the MSA authentication proceeds with an MSA four-handshake (step 4 in Fig. 2.7) using the existing mesh key hierarchy to set up a PTK between the MA and the supplicant MP. After the four-way handshake, the two MPs can initiate the group key handshake procedure to update their group temporal key (GTK) for broadcast messages. Fig. 2.8 shows the overall procedure of MSA in WMN.

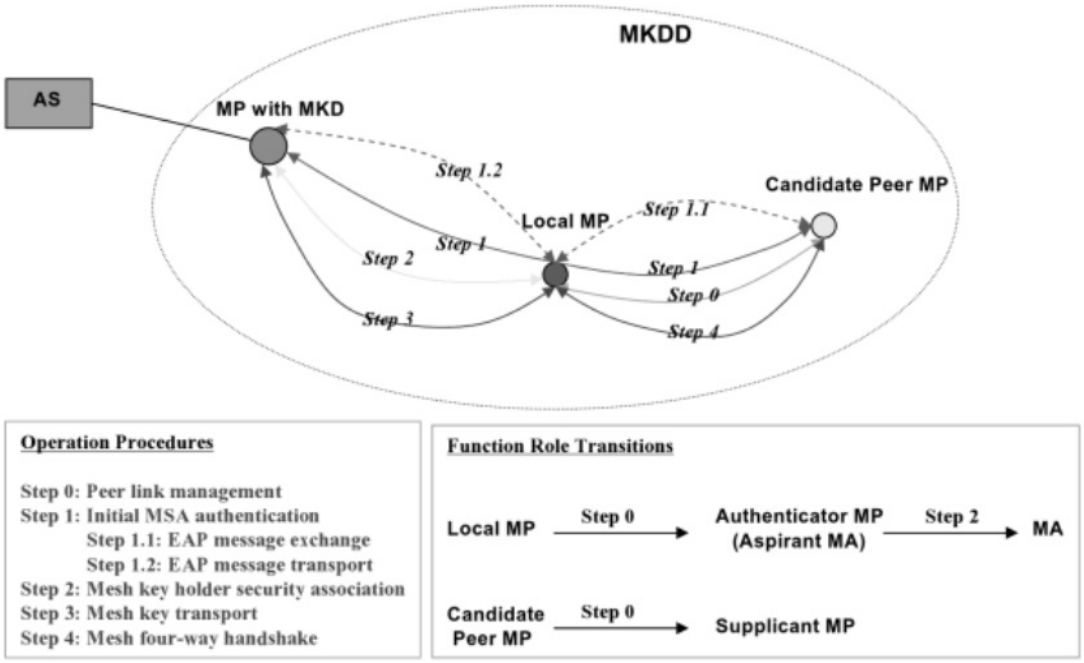

Figure 2.7: Architecture and main function blocks of 802.11s mesh security based on RSNA mechanism [5]

2.2.3 Simultaneous authentication of equals (SAE)

In Fall 2011, IEEE task group standardized SAE protocol to simultaneously authenticate two arbitrary peers and both of them can initiate the authentication process where they do not need to be direct neighbors. SAE results in a pairwise master key (PMK) shared between two peers. The authentication protocol assumes a pre-shared secret, i.e., a password, to be known to all legitimate network entities. Abbreviated Handshake is used for authenticating peers that already share a PMK by using less messages than SAE [29]

The computations used by SAE are either based on Elliptic Curve Cryptography (ECC) or prime modulus finite cyclic groups. In the following we use the notation of ECC-based SAE in which $P(x, y)$ represents a point on a publicly known elliptic curve of the form $y^3 = x^3+ax+b$. By inv we refer to the additive inverse element of a point on the elliptic curve. SAE uses four messages to authenticate two peers in a simultaneous fashion. The message flow of SAE between parties A and B is depicted in Fig. 2.9.

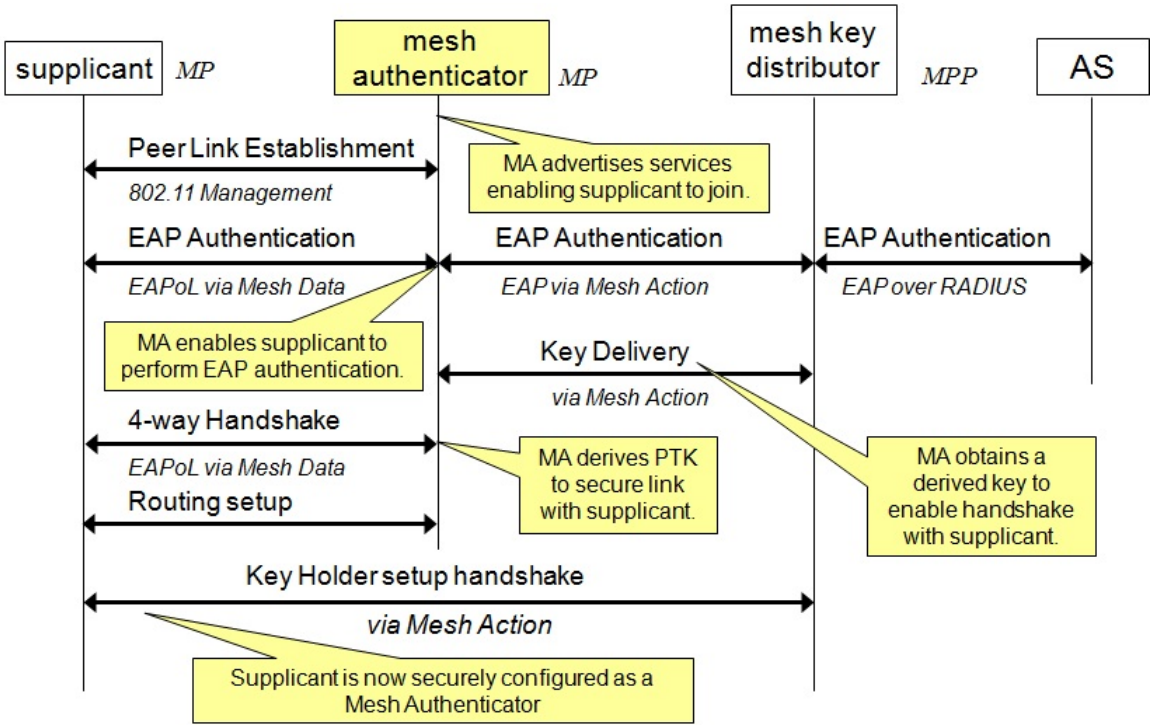

Figure 2.8: mesh security association in WMN [2]

In the first step, the initiating peer generates a password element (PWE) which represents a point on an elliptic curve. The PWE is combined with a hash m containing a combination of MAC addresses of the respective two peers by scalar multiplication to $N = PWE \times m$. The initiating peer A constructs The initiating peer A constructs:

- a commit scalar $csA = (randA + maskA)\ mod\ r$
- and a commit element $ceA = inv(maskA \times N)$.

$randA$ refers to a random number which is essential to computing the key to be shared by both peers. $maskA$ is another value used to blind the transferal of the random number and r is the order of the curve. Upon reception of a peers commit, both peers are able to compute the same secret k using a pre-defined key derivation function F . k is derived by each party based on the other partys commit message, its own random random number, and N such that A computes

$$k = F((randA \times (csB \times N + ceB))$$

and B computes

$$k = F((randB \times (csA \times N + ceA))$$

The computation effectively represents a password authenticated ECC Diffie-Hellman key exchange. Both peers will then build a confirmation message, namely a hash of the secret k, a replay-protection counter and the previously exchanged cs and ce values. If the received confirm message equals the expected result, authentication is considered successful. If authentication was successful, both peers will generate a pairwise master key as

$$PMK = H(k \parallel counter \parallel (csA + csB)\ mod\ r \parallel F(ceA + ceB))$$

Once a PMK has been successfully established, it can later on be used during the Abbreviated Handshake [29].

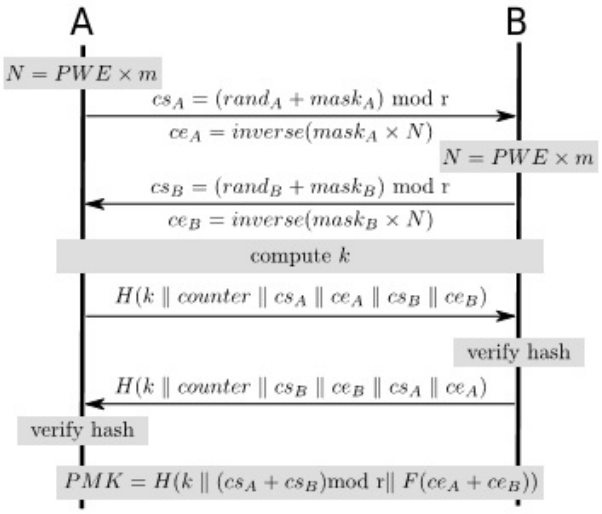

Figure 2.9: ECC-based simultaneous authentication of equals (SAE) [25]

The PMK is used to construct a key hierarchy in which a 128-bit Abbreviated Handshake Key Confirmation Key (AKCK), a 256-bit Abbreviated Handshake Key Encryption Key (AKEK), and a 128-bit Mesh Temporal Key (MTK) are computed. The keys AKCK and AKEK are static in the sense that they can be used to provide data origin authenticity and data confidentiality in multiple runs of the Abbreviated Handshake and Group Key Handshake.

The AKEK is used to encrypt the GTK during the Abbreviated Handshake. The MTK is used to protect the communication between two peers and derived in a more dynamic manner by also using freshly generated random numbers of both peers as input to the key derivation function. The PMK, AKCK and AKEKs lifetime is limited by the passwords lifetime, whereas the MTK should be regenerated on each peering instance.

Abbreviated Handshake: The goal of the protocol is to generate a fresh MTK between two peers that already share a PMK. The new MTK is randomized by using two fresh random numbers selected by the peers. Since the peers share a PMK and therefore AKCK and also AKEK, the exchange of the nonces can be integrity protected. The protocol consists of two messages, i.e. a Peering Open Frame which also contains the random number and a Peering Confirm Frame containing the nonce of the respective other peer. Because of the pairwise encryption, each link is independently secured and a mesh station is required to update its broadcast traffic key with every new peering it establishes [30].

Chapter 3

Smart Grid Neighborhood Area Network

In smart gird, neighborhood area network (NAN) refers to a network of smart meters that are connected to each other in order to send/rely metering data to concentrating nodes, collectors, which in return, send the data over a wide area network (WAN) to the utility center. Wireless mesh network has attracted more attention among other architectures for NAN in which smart meters are deployed in an adaptive wireless mesh network [50] ,[80]. Wireless mesh provides several advantages over the other type of technologies including flexibility, minimal infrastructure, scalability and low configuration cost [35]. Such a wireless mesh network can provide customer-oriented information on electricity use to the operational control systems, which monitor power grid status and estimate electric power demand [66].

In a neighborhood area, smart meters send their data through single/multi-hop communication to collectors. Figure 3.1 shows a multi-tier smart grid network where Tier 2 represents the NAN. In the NAN, smart meters scan perform routing and find their best path to collectors. Each smart meter maintains a list of peers so that in case of failure of one peer, it can switch to the next available peer. Hence, redundant paths make the network more reliable. A fully redundant routing requires each smart meter to discover the best single/multi-hop possible collector in its vicinity and establish a connection with it. In case of detecting loss of connectivity, smart meters are able to re-configure themselves to re-establish the connection to the network [41].

NAN Technologies

In addition to wireless mesh (e.g., 802.11s), other wireless and wired technologies can be utilized. On the wireless side, Worldwide Interoperability for Microwave Access (WiMAX)

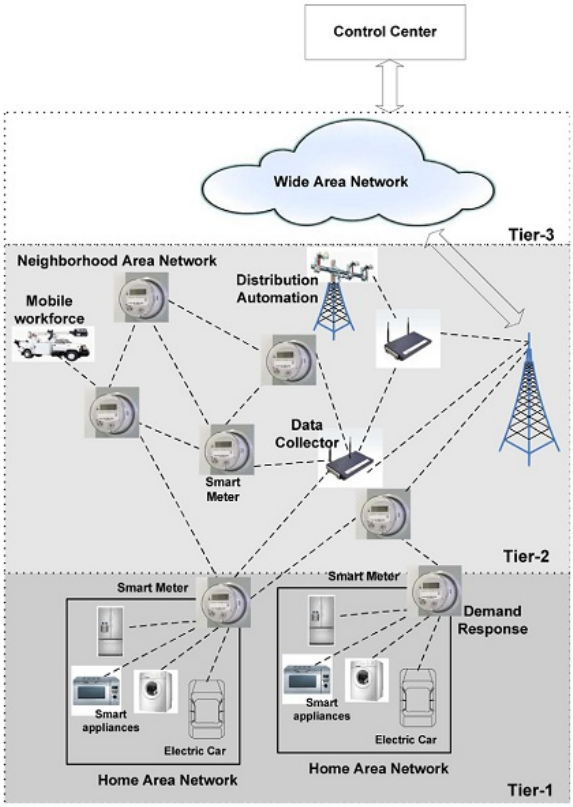

Figure 3.1: Neighborhood area network (NAN) (Tier 2) adopted from[52].

and cellular standards, such as 3G, 4G, and LTE, are some of the stronger candidates. On the wired side, Ethernet, Power-line Communications (PLC) or Data over Cable Service Interface Specification (DOCSIS) are possible options to use. In this work, we consider WiFi-mesh as the network architecture of NAN.

NAN Components

NAN should provide scalable, secure access and device management for mesh-connected AMI devices such as water, electric and gas meters with instantaneous enterprise-to-gateway connectivity to residential and commercial locations. Information is available on-schedule, on-demand, or on-event from virtually anywhere via these wireless communication devices. Generally, a NAN can consist of several smaller NANs where each NAN is defined by a set of smart meters that communicate with one collector. Each NAN can have an ID (e.g., mesh ID) which can be identified by its collector.

A NAN consists of different components which are mainly classified into followings:

- **Collector:** is a communications gateway that coordinates communication within the NAN. It operates as the intermediary data concentrators, collecting and filtering data from groups of mesh-enabled meters, and economically sharing wide area network resources making communication more affordable while ensuring high performance.

- **Smart Meter:** measures and transmits-fine-grained electric power usage information and information on the quality of electricity to the utility which can use this information for generating customer bills and also automatically control the consumption of electricity through delivery of load control messages to the smart meters. Smart meters automatically establish connection with the collectors based on application performance settings to ensure timely and secure delivery of data [23].

- **Advanced Meter Reading Application**: is the most important application in the Advanced metering infrastructure that records customer consumption and transmits the measurements over the NAN to collectors hourly or at a faster pace [57]. Typically, in North America, meters measure 15-minute meter readings, and transmit them to meter data management system (MDMS) in the utility center. When the meter reading is lost, MDMS checks that the communication with the meter is recovered. The meter retransmits the data in response to a recollection request from MDMS.

In addition to reading functionality, NAN might include capabilities such as remote meter management (connecting/disconnecting smart meters), recording and transferring event logs, security logs and outage reporting. Table 3.1 represents the message types and their specification being transferred in the NAN [3].

3.1 NAN Communication Requirements

In this section, we discuss main requirements and features of NAN including security and privacy, threat analysis, routing, scalability, and intrusion detection system.

Table 3.1: Data traffic between utility company and NAN adopted from [3].

Type		Latency	Packet Size	Frequency	Reliability
Safety and Maintenance	Monitoring Factor and Quality	NRT[1]	1536 Bytes	Once in a cycle	$> 98\%$
	Meter Health	NRT	512 bits	E[2]	$> 99\%$
	Cable Health	RT[3]	512 bits	E	$> 99\%$
	Outage Detection and Management	RT	1024 bits	E	$> 99\%$
	Tamper or theft detection	RT	512 bits	E	$> 98\%$
	Local Energy Generator Health	NRT	512 bits	E	$> 99\%$
Meter Reading	Electricity Consumption at Specific Times	NRT	512 bits	On Command	$> 98\%$
	Electricity Consumption	RT	1024 bits	Periodically	$> 99\%$
	Water and Gas Consumption	NRT	1024 bits	On Command (Every 5 Min)	$> 98\%$
Demand Response	Energy Consumption of each Appliance	NRT	512 bits	On Command	$> 98\%$
	Control Signals from Utilities to Smart Appliances	NRT	512 bits	On Command (Peak hour)	$> 98\%$
Time of Use Rates and Peak Hours	Peak Rate	NRT	512 bits	Every 30-60 Min	$> 98\%$
	Moderate Peak Rate				
	Off-Peak Rate				
	Peak Hours	NRT	512 bits	Every 30-60 Min	$> 98\%$
	Moderate Peak Hours				
	Off-peak hours				

[1] Near real time, in order of minutes
[2] Event-driven
[3] Real time

3.1.1 Security and Privacy

Security as a major requirement covers all aspects of the NAN, from physical devices to routing protocol operations. Many end-point devices in power transmission and distribution networks, and power generation networks are located in an open, potentially insecure environment which makes them prone to malicious physical attacks. These devices must be protected properly against unauthorized access such as modifying the routing table or some network information stored in the compromised device.

Another major concern in the NAN is the privacy of the power data. Many customers would be reluctant to expose their power usage data (as well as the electric vehicle locations). Hence confidentiality and anonymity should be provided at all times. Non-repudiation is also required in some electricity transaction applications such as in the future electricity trade-market, and electric vehicles power usage in public or private charging stations.

Routing protocols should be designed by taking into account the security and privacy requirements of the specific NAN applications. Confidentiality, integrity, and authentication should also be provided for routing functionalities.

For securing NAN, the effective mesh security association (EMSA) which was discussed in Chapter 2 can be used; collectors can play the role of Mesh key distributors (MKDs) which are responsible for key management with their domains. The collector as a MKD can also provide a secure link to an external authentication server (e.g., a RADIUS server) in the utility server[40]. A NAN can be an example of a domain. An already authenticated smart meter can act as mesh authenticator (MA) to participate in key distribution and therefore authenticating a candidate smart meter to join the network.

In this work, however, we suppose that the security of NAN wireless mesh network is based on simultaneous authentication of equals (SAE) which was discussed in the Chapter 2. SAE is a more recent security standard for wireless mesh networks. In this security scheme, two arbitrary smart meters can initiate the authentication process where they do not need to be direct neighbors. Therefore, there is no need to have key hierarchies and a key distribution mechanism. When smart meters discover each other (and security is enabled), they take part in an SAE exchange. If SAE completes successfully, each smart meter knows the other party possesses the mesh password and, as a by-product of the SAE exchange, the two peers establish a cryptographically strong key. This key is used with the authenticated mesh peering exchange (AMPE) to establish a secure peering and derive a session key to protect mesh traffic, including routing traffic [29].

3.1.2 Threat Analysis

For a hacker to attack a system, he/she must have an incentive. That incentive may be financial gain or the pure thrill of being the first to crack a system. In case of the smart grid, besides the obvious incentive of causing widespread chaos, another strong incentive for an attacker is making money. Similar to most of the infrastructural services, we identify following incentives

for attacking NAN:

- **Financial Gain:** One of main incentives to attack smart grid is energy fraud in which attackers try to tamper with metering infrastructure so that they are not billed for the energy they consume. This includes simple energy theft by manipulating smart meters. Thefts might aim at obtaining consumption behavior patterns by eavesdropping meter readings to find out when someone is not at home. Hackers might further act on behalf of others due to economic reasons, e.g., harming competitors with targeted blackouts [68].

- **Personal Revenge:** An attacker may blackout particular households or companies due to personal reasons. A more sophisticated attack can involve the transmission of tampered meter data so that a victim is billed an extraordinary high amount of energy.

- **Looking for Hacker Community Acceptance or Chaos:** Here an attacker wants to prove his own capabilities, e.g., by provoking wide-spread power outages.

The primary functionalities of NAN is that smart meters push meter readings toward collectors in the one direction and on the reverse direction the utility center sends control messages to smart meters e.g., blackout a customer who is unwilling to pay his bill. Since NAN communication plays an important role in achieving such functionalities, here we focus on the denial of service (DoS) attacks that target the availability of the NAN functions. Such attacks can be derived form the attacks targeting wireless mesh networks which were elaborated in Chapter 2. Jamming attacks in both physical and MAC layers can target the proper functionality of the NAN. Signal jamming prevents the smart meters from transmitting their meter readings or their sensitive information such as alarms. The attacker can also target the MAC protocols at the link layer by jamming only Request to Send (RTS) packets. Trivial jamming, periodic jamming and reactive jamming [18] are some examples of MAC layer jamming attacks.

Another DoS attack that can target NAN is data injection or flooding. The attacker tries to flood the NAN by fabricated status messages and false readings in short time intervals [68].
A compromised smart meter can act as a black hole or gray hole to attract most, if not all, of the NAN traffic toward itself and simply drop the packets making some part of the NAN unavailable.

In a wormhole attack, two colluding compromised smart meters can target the availability of the NAN. In this attack, the smart meters in the NAN which are not direct neighbors are

connected to each other via a high-speed connection. One of the compromised smart meters sends route requests (RREQ) that it hears form its neighbors during the route discovery phase through the wormhole link to the other malicious smart meter. The other compromised smart meter which is in the vicinity of destination (collector) sends the RREQ to the collector. Since such a RREQ is the first one to reach the collector, the collector replies the route response (RREP) to the malicious smart meter and ignores later received RREQs with the same ID. Replaying RREP by the first compromised smart meter, makes the neighbor smart meters think that the wormhole path is the best path to the collector. As a result, smart meters choose the wormhole link as the best path to reach the collector. There is another possibility that the two colluding nodes are in different NANs in which they make the smart meters in NAN 1 think that other smart meters in NAN 2 are their neighbors.

After launching wormhole attack, compromised nodes can either act actively or passively. They can simply drop all data packets (black hole attack) or they can selectively drop packets (gray hole attack) e.g., dropping a packet every n packets, a packet every t seconds, or a randomly selected number of the packets. The attackers may also keep intercepting the packets to derive useful information about the availability of individuals at homes for burgling purposes. When wormhole attack is performed between two neighbor NANs, some critical smart meter messages such as status messages or alarms may miss their deadline. In such an attack, in first place wormhole nodes attract such traffics and make them travel longer distance (e.g., through another NAN) than their real shortest paths.

3.1.3 Routing

Designing the best practical routing protocol for the NAN has been a hot topic in the research community. Some approaches have suggested using reactive routing protocols such as Ad hoc On-Demand Distance Vector (AODV), some proposed to use proactive routing protocols such as Destination Sequenced Distance Vector (DSDV) while others believe that a combination of the reactive and proactive routing suits the requirement of the NAN. The work in [4] analyzes the resiliency of the NAN against a DoS attack considering three types of routing protocols including AODV, Dynamic Source Roting (DSR) and DSDV. Based on the simulation results, it has been concluded that AODV outperforms others considering some performance metrics such as packet delivery ratio, average end-to-end delay and etc. In [38], two modifications have been proposed to 802.11s routing protocol to make the protocol applicable for smart including

modification to the calculation method of metric defined in the 802.11s and the route fluctuation prevention algorithm.

Routing Protocol for Low Power and Lossy Networks (RPL) is currently under development by the Internet engineering task force (IETF) to support various applications for low power and lossy networks (LLNs) such as in urban environment. RPL is a distance vector routing algorithm that uses a destination oriented directed acyclic graph (DODAG) to maintain the state of the network. In this algorithm, each node keeps its position in a DODAG calculating a rank to determine its relations with the root and other nodes in the DAG. The specification of this protocol is found in [72]. Authors in [74] modify RPL for NAN by proposing a DAG rank computation to fit the requirements of NAN. In [41], RPL has been enhanced by designing self-organizing mesh solution based on which smart meters can automatically discover the more suitable collectors in their vicinity, detect loss of connectivity and re-configure themselves to connect to the NAN. Distributed autonomous depth-first routing (DADR) [34] is a proactive routing algorithm suggested for the use in NAN which acts exactly the same as traditional distance vector algorithms when network in its normal operation. In case where topology changes frequently, it uses a light weight control plan and uses its forwarding plane to inform the network about any link failures [12].

Hybrid Routing Protocol (Hydro) [24] is another routing protocol suitable for NAN. It is a link-state routing protocol for LLNs. It uses a distributed algorithm for directed acyclic graph (DAG) formation that provides multiple paths to a border router. Figure 3.2 depicts the state of the art routing protocols that have been suggested for NAN.

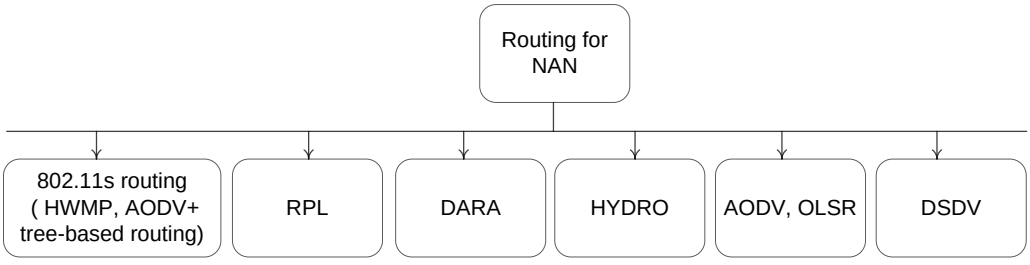

Figure 3.2: State of the art routing protocols for NAN.

In this work, we consider AODV as the routing protocol for NAN. Our justification relies on the fact that the network topology in NAN is stationary and smart meters do not need to keep the synchronized map of the whole network. In addition, we assume a smart meter constructs a

dedicated path to the collectors and it keeps using it until there is a problem with the path (e.g., loosing the connection to its next hop) [14].

3.1.4 Scalability

The ability to provide an acceptable level of service with a huge number of nodes is very crucial for the smart grid. Millions of smart meters will be attached to communication networks to deliver power usage data from each household to utility companies. The number of nodes connected to the network at a certain location would vary depending on the population density in that area. For instance, while urban areas will have a high density of customers, the number of houses distributed in rural areas will be low. Therefore, any proposed routing protocol for smart grid should be able to scale under a variety of use cases with their distinct operational requirements. Route discovery, maintenance and key distribution in case of secure routing will grow rapidly with the network size. This design issue may significantly affect the way the routing protocols are designed depending on the application, underlying network and the link metrics used [52].

3.1.5 Intrusion Detection System

Current security solutions to protect NAN usually include physical controls (e.g., tamper-resistant seals on meters), meter authentication and encryption of all network communications, and network controls (firewalls are deployed at the access points and in front of the headend). On the security detection side, intrusion detection systems (IDSs) are usually deployed inside the utility network to identify attacks against the headend [15]. This means that current intrusion detection solutions for NAN are based on a central location e.g., in the utility center and they can suffer from scalability issues (a large-scale network can reach several million smart meters.) More importantly, security administrators have no ability to see the traffic among meters at the edge of the NAN, and they have to rely on encryption, secure key storage, and the use of protected radio frequency spectrum to prevent intrusions. As a result, NAN lacks a reliable monitoring solution so that in case of any security breaches, the violation can be detected or deterred.

In Chapter 5, we propose an IDS for NAN and evaluate the proposed solution in OPNET based on generated false positive and false negative alarms.

Chapter 4

Architectural Design of NAN IDS

In this chapter we discuss our IDS architecture and design steps. Initially we highlight main features of OPNET Modeler which was used to simulate our IDS. Moreover, we explain the NAN infrastructure and its architecture. Later on, we will talk about integrating analytical and simulation model.

4.1 OPNET Modeler Overview

OPNET Modeler provides a comprehensive development environment supporting the modeling of communication networks and distributed systems. Both behavior and performance of modeled systems can be analyzed by performing discrete event simulations. The OPNET Modeler environment incorporates tools for all phases of a study, including model design, simulation, data collection, and data analysis.

OPNET Modeler supports model specification with a number of tools, called editors, which capture the characteristics of a modeled systems behavior. Because it is based on a suite of editors that address different aspects of a model, OPNET Modeler is able to offer specific capabilities to address the diverse issues encountered in networks and distributed systems. To present the model developer with an intuitive interface, these editors handle the required modeling information in a manner that is parallel to the structure of real network systems. Therefore, the model-specification editors are organized hierarchically. Models built in the Project Editor rely on elements specified in the Node Editor; in turn, when working in the Node Editor, models are defined in the Process Editor and External System Editor. The remaining editors are used

to define various data models, typically tables of values, that are later referenced by process- or node-level models. This organization is reflected in the following list:

- Project Editor-Develop network models: Network models are made up of subnets and node models. This editor also includes basic simulation and analysis capabilities.
- Node Editor-Develop node models: Nnode models are objects in a network model. Node models are made up of modules with process models. Modules may also include parameter models.
- Process EditorDevelop process models: Process models control module behavior and may reference parameter models. They can be used to develop models of decision-making processes representing protocols, algorithms, resource managers, operating systems, and so on.
- External System Editor: Develop external system definitions. External system definitions are necessary for co-simulation.
- Link Model Editor: Creates, edits, and views link models.
- Packet Format Editor: Develop packet formats models. Packet formats dictate the structure and order of information stored in a packet.
- ICI Editor: Creates, edits, and views interface control information (ICI) formats. ICIs are used to communicate control information between processes.
- PDF Editor: Creates, edits, and viewes probability density functions (PDFs). PDFs can be used to control certain events, such as the frequency of packet generation in a source module.

The Network, Node, Process, and External System modeling environments are sometimes referred to as the modeling domains of OPNET Modeler because they span all the hierarchical levels of a model and are shown in Figs. 4.1, 4.2 and 4.3 respectively.

The Network, Node, Process, and External System modeling domains are hierarchically related to each other. Process models and external systems are instantiated in the Node Domain, and node models are instantiated in the Network Domain. Within each domain, objects used to define models may also have hierarchical relationships to each other. The hierarchy of objects

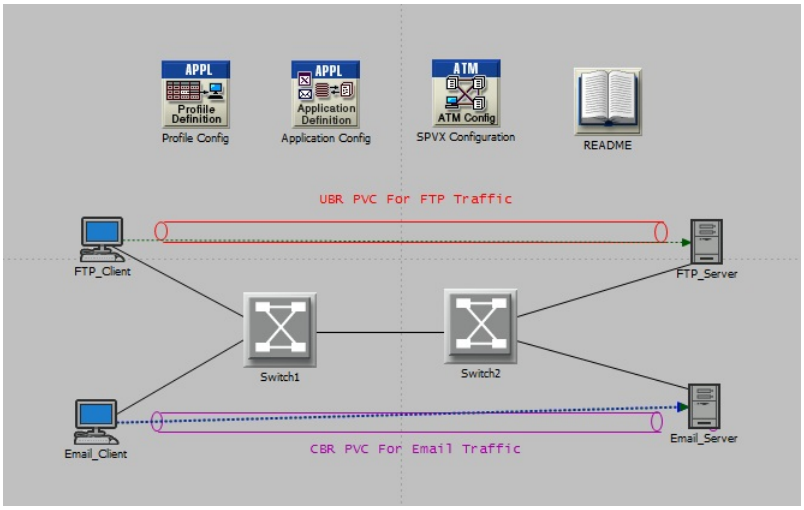

Figure 4.1: OPNET network model.

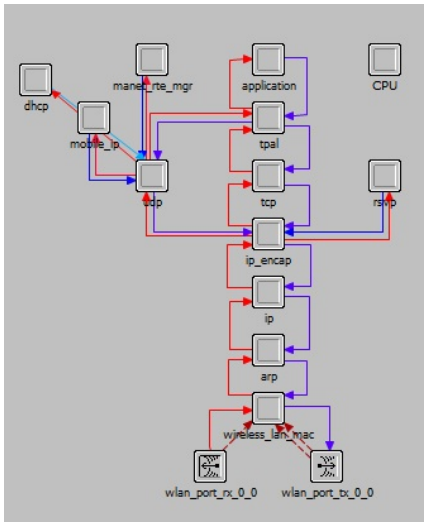

Figure 4.2: OPNET node model.

is specific to each domain to support appropriate decomposition of particular types of objects. Many object types do not support decomposition. A global view of the object hierarchy is illustrated in Fig. 4.4. Here we explain node model, process model, and packet format editor in more details as they have been extensively used in developing our IDS.

Node Editor

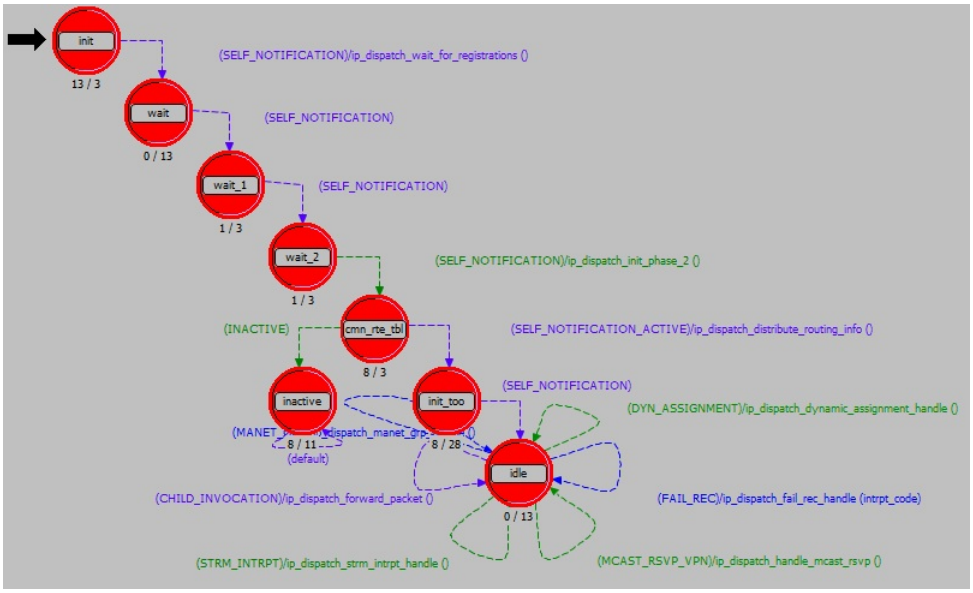

Figure 4.3: OPNET process model.

The Node editor is used to specify the structure of device models. These device models can be instantiated as node objects in the Network Domain (such as computers, packet switches, and bridges). In addition to the structure, the node model developer defines the interface of a node model, which determines what aspects of the node model are visible to its user. This includes the attributes and statistics of the node model. This section summarizes the objects used in the Node Editor and the operations that it provides.

Nodes are composed of several different types of objects called modules. At the node level, modules are black boxes with attributes that can be configured to control their behavior. Each one represents particular functions of the nodes operation and they can be active concurrently. Several types of connections support flow of data between the modules within a node. The objects used to build node models are listed in the following table. The node model embraces processor,queue transmitter, receiver, packet stream, statistic wire, logical association and Esys modules [56].

Process Editor

The Process Editor is used to specify the behavior of process models. Process models are instantiated as processes in the Node Domain and exist within processor, queue, and esys modules. Processes can be independently executing threads of control that do general com-

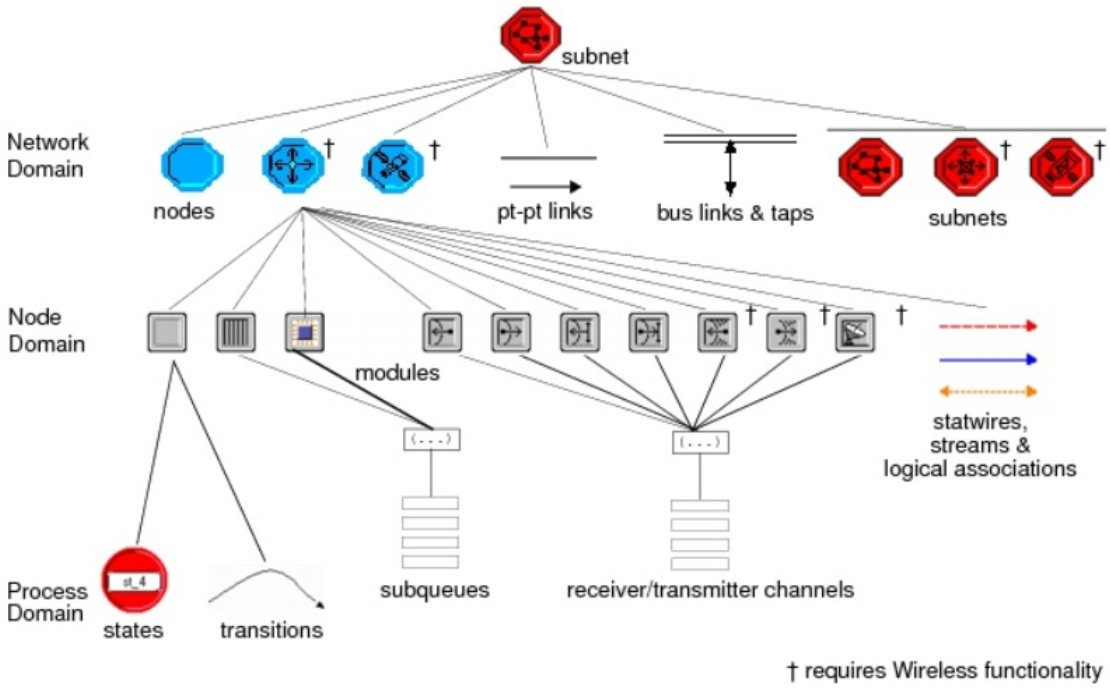

Figure 4.4: Object hierarchy in OPNET [56].

munications and data processing functions. They can represent functionality that would be implemented both in hardware and in software. In addition to the behavior of a process, the process model developer defines the models interfaces, which determines what aspects of the process model are visible to its user. This includes the attributes and statistics of the process model. This section summarizes the objects used in the Process Editor and the operations that it provides.

Process models use a finite state machine (FSM) paradigm to express behavior that depends on current state and new stimuli. FSMs are represented using a state transition diagram (STD) notation. The states of the process and the transitions between them are depicted as graphical objects. The objects used to build process models are listed in the following table.
The process model comprises of states, transitions, and model level information blocks. States represent a mode of the process which has been attained due to previous stimuli and corresponding decisions. States contain code expressing processing that is performed immediately after they are entered, or immediately before they are exited. A state can be forced or unforced. A process blocks immediately upon executing the enter code of an unforced state, at which

point it waits for a new interrupt before continuing.
Transitions indicate a possible path that a process can take from a source state to a destination state. Each state can be the source and destination of any number of transitions. A transition has a condition statement which specifies the requirements for the process to follow the transition. An executive statement specifies actions that are to be taken when the process does follow the transition.

Several blocks of text specify additional components of the process, including: declaration of state (persistent), and temporary (scratch) variables; user-defined functions that can be called by the process states and transitions; code to be executed upon process termination; and declaration of globally-scoped variables, data structures, etc.

Packet Format Editor

The Packet Format Editor provides a way to specify the collection of fields contained by a formatted packet. Each field has attributes specifying its name, type, size, and default value. Modeling each fields size allows the overall size of the packet to be calculated. Fields may be one of several types: integer, double, structure, information, and packet.The Packet Format Editor provides operations to support the creation and editing of packet fields.

- Structure fields: allow inclusion of arbitrary complex data in packets.
- Information fields: model bulk data in terms of its size, without concern for actual content.
- Packet fields: model packet encapsulation by layered protocols.

There are several types of analyses that can be done using OPNET analysis software: Discrete Event Simulation (DES), Flow Analysis, Survivability Analysis, and NetDoctor Validation. We have used DES which provides the most detailed results but has the longest run times. This is because it does a more thorough analysis than the others, handling explicit traffic, conversation pair traffic, and link loads. The other types of analyzes answer specific types of questions, but generate results much faster than a discrete event simulation. A flow analysis, for example, handles only conversation pair traffic (flows) and a NetDoctor validation does not consider traffic at all.

OPNET has been developed with Proto-C which provides a flexible platform that has the ability to model a wide range of systems. The language provides specific support by adopting

a state-transition approach which is well-suited to discrete event systems, and by supplying a number of Kernel Procedures (KPs) that are oriented toward network and distributed systems modeling. At the same time however, Proto-C preserves generality by incorporating all the capabilities of the C/C++ programming language.

Due to the generality of Proto-C and the flexibility of the state-transition paradigm, most systems can be accurately represented by many different Proto-C models. The diversity of possible implementations results from the power of the tool but can also present some disadvantages; namely, the model designer may experience hesitation at the start of a models development effort, as the merits of each approach are considered. This is particularly true in the case of a beginning OPNET Modeler user. In addition, the models that one developer creates over time may vary in approach rather than converge on one consistent style; this adds difficulty to the future task of maintaining the models, particularly if different developers are involved.

4.2 OPNET Simulation Kernel API

One of the most important OPNET modeler's API is Kernel API which was frequently used in development of our IDS. Kernel API allows to access most-used functions and are categorized by:

- Attribute Acess: Used to get/set attributes.
- Distribution: Loads distributions by name; Obtains outcomes from loaded distributions.
- Dynamic processes: Creates a new child process of a given type; destroys a process and identify the current process. Invokes another process (cause it to execute now). As an invoked process, gets optional state that is passed.
- Events and time: Cancels an event, obtains current simulation time and terminates simulation.
- Identification and discovery: Finds the containing object and the parent of an object.
- Interrupt processing: Schedules an interrupt for this object or another at a given time. Optionally passes a code.

- Packet generation and processing: Creates, copies, or destroys a packet. Gets or sends a packet.

- Statistics Recording: Obtains a handle for a statistic, given its name. Writes a new value to a particular statistic.

4.3 Network Architecture

We have simulated a smart grid deployment scenario which mainly focuses on the NAN part. Our simulation consists of NAN, wide area network (WAN), and the utility site. Fig. 4.5 depicts a subnet level view of the scenario that is used in our simulation. We have used wireless LAN workstation (wkstn _wireless _lan) model for smart meters. Node wkstn_wireless_lan represents a workstation with client-server applications running over TCP/IP and UDP/IP. Gateway manet_gtwy_wlan_ethernet_splip 4 was used as collectors node model which has two interfaces one for the connectivity to the NAN and the other on for connecting to the WAN. We have utilized the node model ip32_cloud to simulate the WAN. The ip32_cloud represents an IP cloud supporting up to 32 serial line interfaces at selectable data rate through which an IP traffic can be modeled. IP packets arriving on any cloud interface are routed to the appropriate output interface based on their destination IP address.

For defining the application running on the smart meters, we choose automatic reading application which was listed in the Table 3.1 in Chapter 3. Automatic reading is a non-polling event where smart meters send their meter readings in a predefined frequency. For defining such an application we had to create a custom application as OPNET's default application formats did not match our need. Section 4.3.1 describes how we created a custom application for the purpose of automatic reading in the NAN.

4.3.1 Custom Application

The custom application is an application modeling framework that can be utilized to model a broad class of applications. It can be used when the application of interest does not correspond to any of the standard applications. The custom application provides attributes that allow for configuring various aspects of the application in detail. A custom application can be used to represent any number of tiers.

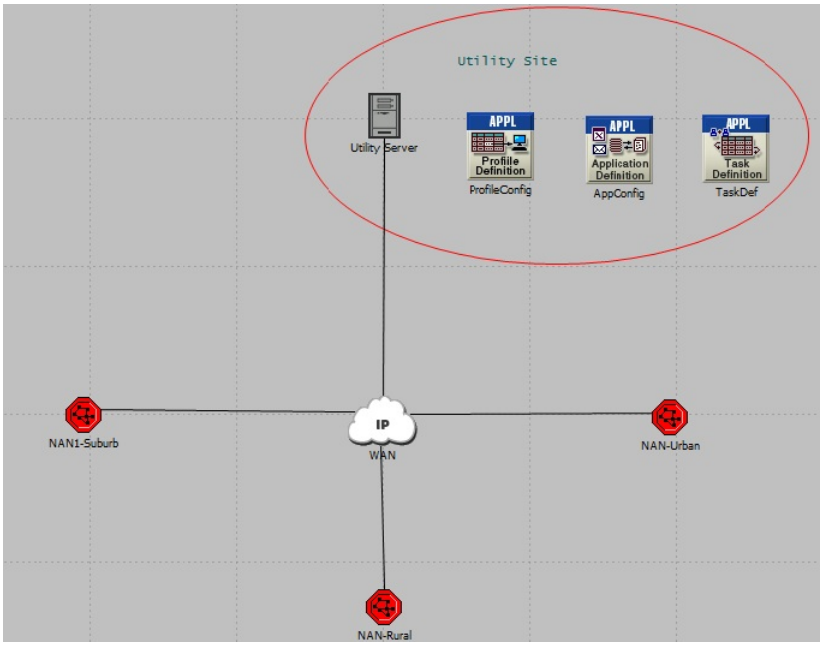

Figure 4.5: High level simulation scenario.

There is a hierarchy in defining a custom application including Phase, Task, Application, and Profile from the lowest to the highest level respectively. The core functionality of an application is performed in the task part. A task consists of many interactions between a client and a server or between successive servers. Tasks can include several phases by which the lowest level functions in a task can be defined. Each phase consists of a data transfer and/or a processing event, which can occur at any end device. This end device becomes a tier for the application. Subsequent phases are typically set up to occur in a chain, where the destination of one data transfer phase becomes the source of the next data transfer phase. The entire task is complete only when the last phase of the task has been completed.

In the Application level, the tasks are associated with applications. In this level we define the number of tasks we have in the application. Also the order, and the weight of different tasks are specified within Application. Profile captures the usage pattern of a set of applications used in the network. Essentially, this captures all of the applications in use, the time at which each application is started, the duration of use of an application, and the frequency of application use. Fig. 4.6 illustrates the automatic reading application that is created using custom application. We can see that meter readings are sent by a smart meter every 30 minutes in a duration of 2

hours.

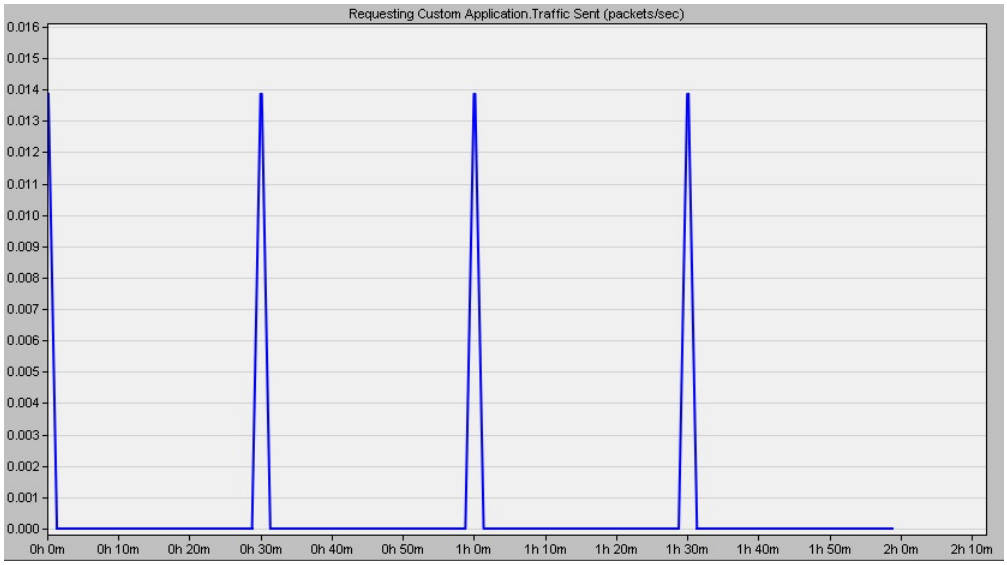

Figure 4.6: Automatic meter reading data sent by a smart meter during 2 hours

4.4 IDS Architecture and Design

The proposed IDS is a distributed solution in which the task of intrusion detection is performed by some nodes which have enough communication and computation capacities. As smart meters are nodes with limited communication and computation features, this seems as a suitable solution for intrusion detection in smart grid NAN.

Figure 4.7 depicts the proposed IDS in the NAN in which some nodes are capable of IDS. We choose collectors in each NAN as monitoring nodes since they have higher capacity and computational power and tamper resistant hardware. To justify our choice for selecting collectors as IDS nodes, we first need to know the functionality of collectors in the NAN.

In order to save energy in collection of data coming from smart meters, collectors instead of retransmitting the received data, forward the aggregated data to the utility center by combining the packets (saving headers) or even removing redundant information [11].

We assume that there is an end-to-end security between smart meters and collectors (as trust points) which means collectors decrypt the smart meter data, aggregate, then re-encrypt, and

forward it to the utility center over the wide area network. We are aware that that aggregation can be performed on the encrypted data (e.g., using additive privacy homomorphism protocols [11]), but the first approach (aggregation after decryption at collectors) better fits our IDS solution. This enhances the IDS features in many ways. The detection task is performed in a faster pace. For example, false data packets can be detected sooner at the collectors rather than remaining undetected until they are decrypted at the utility center. More importantly, by distributing IDS nodes on collectors, we solve the problem of scalability which can occur in a central approach which will be discussed more in next chapter.

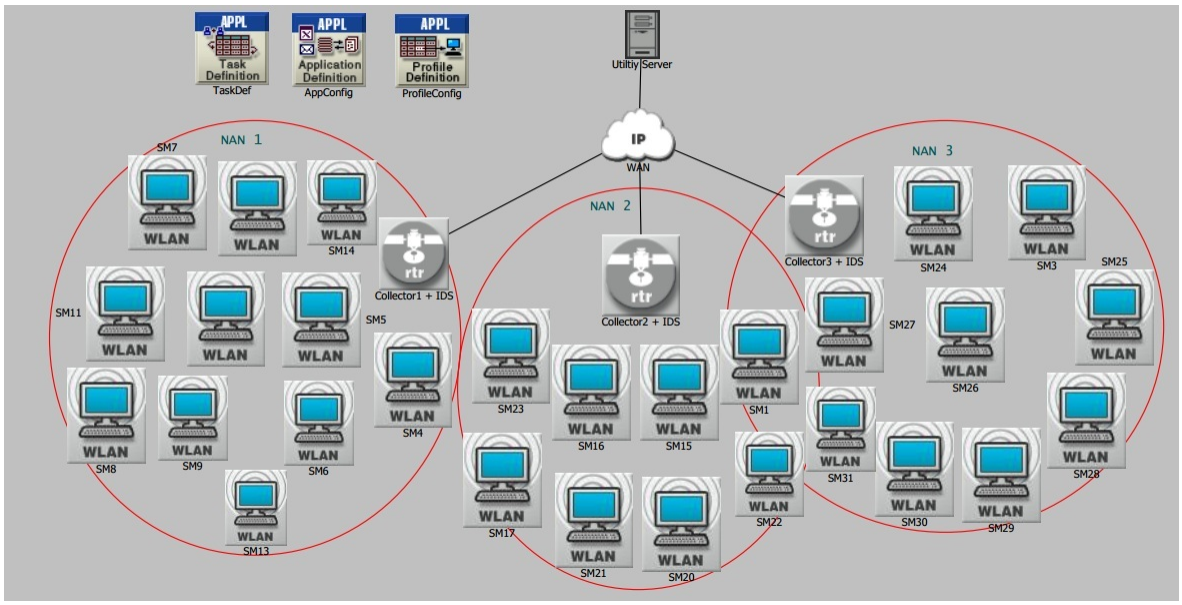

Figure 4.7: NAN topology: One IDS per NAN

As a result, we set up our IDS on the collector in each NAN to monitor the connected smart meters. To implement the IDS, we needed to study the collector model shown in Fig. 4.10 in details to find the best place for implementing IDS.

As Wormhole attack occurs during the routing control plane, AODV module should be used to be examined by IDS. AODV is implemented at the IP layer. In Fig. 4.8, which represents an example of a node node model, ip_dispatch (Fig. 4.9) is the root process for IP and has as a child process, manet_ mgr. Manet_mgr acts as manager process for AODV and provides a common interface to multiple MANET routing protocols (AODV, DSR, and TORA). manet_ mgr is responsible for spawning the AODV child process when a node is configured for AODV.

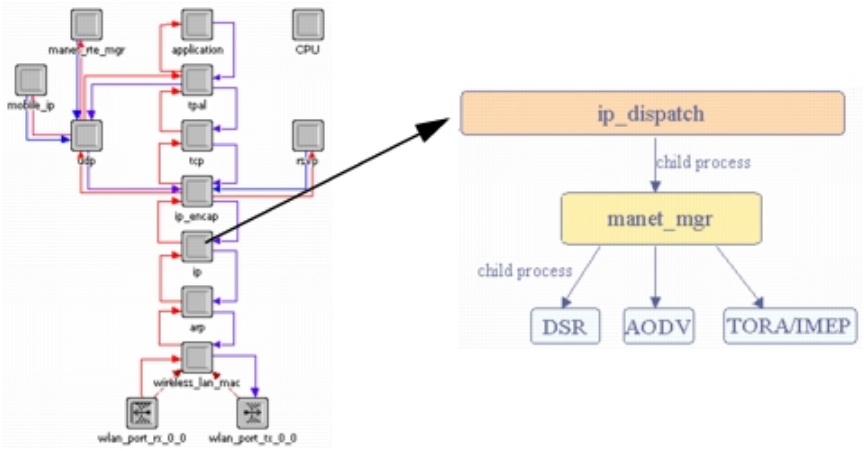

Figure 4.8: Node model surrounding AODV

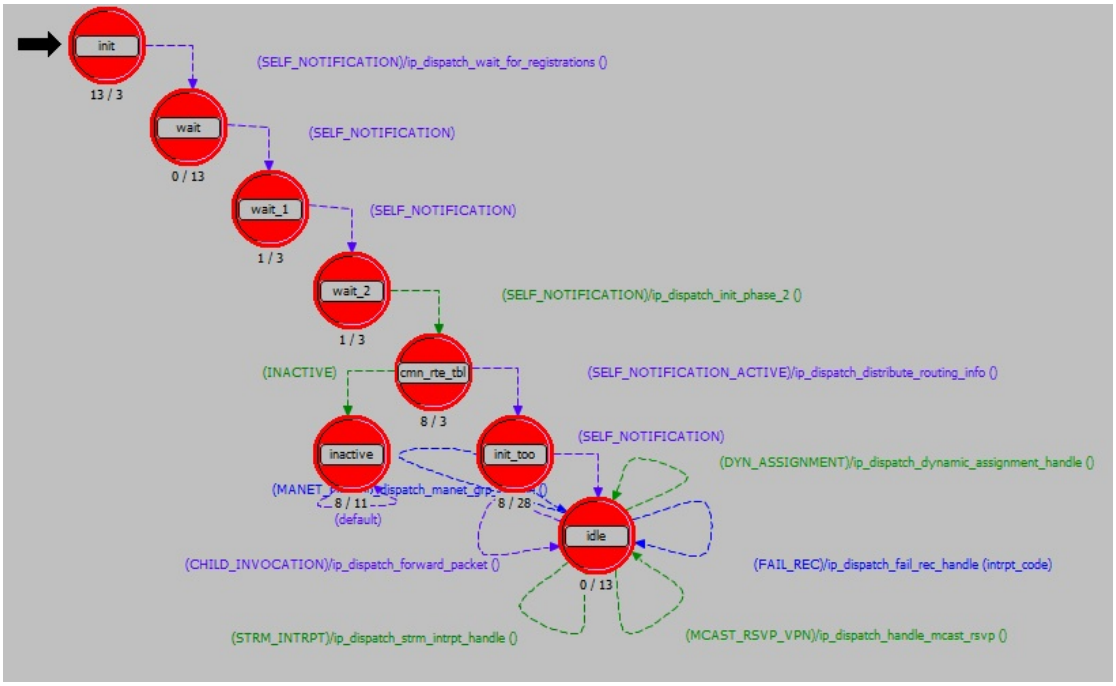

Figure 4.9: ip_dispatch module

We have developed a separate module called NAN-IDS as shown in Fig. 4.10 to host our IDS engine. We have implemented new manet_mgr and aodv-rte processes to facilitate

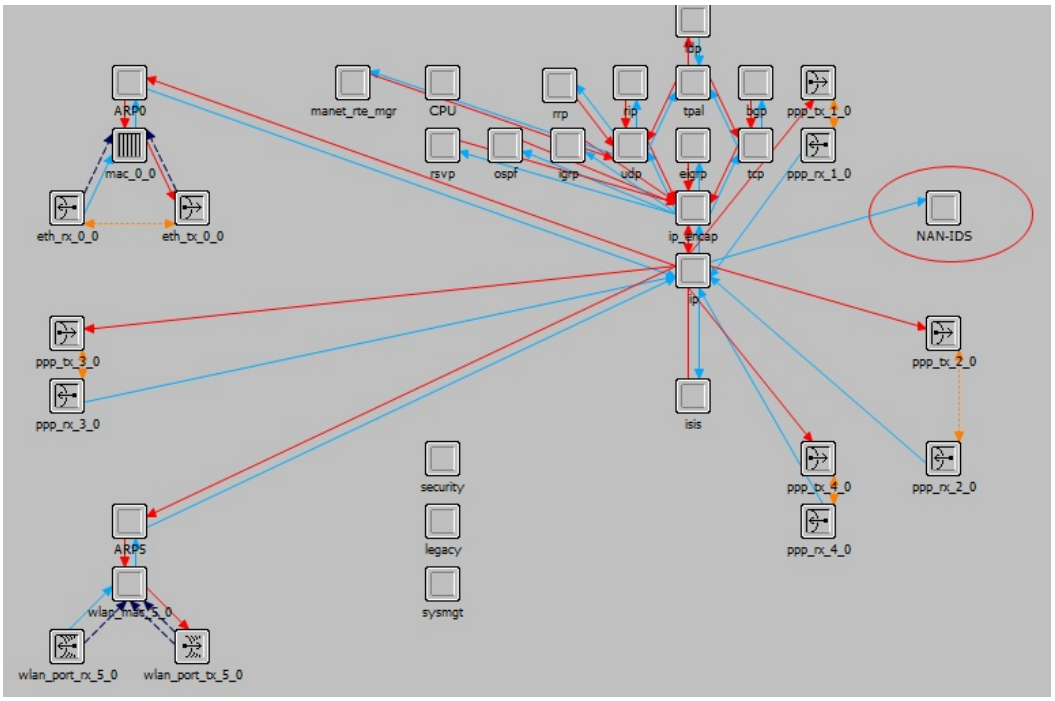

Figure 4.10: Collector node model.

our IDS operation. Our IDS makes use of an analytical model for computing estimated hop counts. The analytical model has been implemented in Maple from MapleSoft [33]. Since our IDS is a hybrid solution of simulation and analytical modeling, for the first time we integrated Maple engine into OPNET Modeler. The detailed structure of the IDS building blocks and its detection results will be discussed in Chapter 5.

Chapter 5

An Intrusion Detection System for Smart Grid Neighborhood Area Network

In order to ensure the reliability and security of neighborhood area network (NAN) in smart grid, attack prevention techniques are of paramount importance to protect the NAN communications from attackers. On the other hand, intrusion detection systems (IDSs), as a second wall of defense, should be in place to detect security breaches and the attacks that go unnoticed by the security mechanisms.

Current security solutions to protect NAN usually include physical controls (e.g., tamper-resistant seals on meters), meter authentication and encryption of all network communications. On the detection side, the IDS resides on a central point (i.e., utility server) which includes core data processing and detection intelligence of the IDS. Such a traditional approach will fail in the context of NAN [16]. The reason is that advanced metering infrastructure can reach several millions smart meters and the traffic load, required storage and computational capabilities will be overwhelming for a central point [37]. Thus, a central approach will suffer from scalability issues. More importantly, security administrators have no ability to see the traffic among meters at the edge of the NAN, and they have to rely on encryption, secure key storage, and the use of protected radio frequency spectrum to prevent intrusions. As a result, a distributed approach should be adopted.

In this chapter, we propose a distributed IDS which monitors the smart meter communications and detects malicious behavior in the NAN.

5.1 Proposed IDS for NAN

The proposed IDS is a hybrid of signature-based and anomaly-based detection systems. We seek for signature of attacks in the communication performed in the smart meter networks and compare it with the behavior that is expected from the nodes. If anomalies are found within the network, the IDS will generate alarms. Following the detailed description of our proposed IDS, its architecture and detection mechanisms are explained.

Collectors monitor the smart meters in their corresponding NAN in terms of network behavior and detect attacks and identifies possible attacks. The NAN IDS can provide reports of the detected attacks and transmit them either to a higher level IDS, e.g., wide area network (WAN) IDS, or to the utility center to take appropriate responses. In some cases, the response mechanisms can also be implemented in the NAN (e.g., excluding attackers from communication paths). However, the main focus of this work is on detection mechanisms rather than response techniques.

5.2 Detection Mechanism

Among the attacks discussed in Chapter 2, we focus on detecting Wormhole attack in our IDS. Our detection mechanisms will be explained in following sections.

5.2.1 Wormhole Attack

As discussed previously, Wormhole attack occurs during route discovery phase where a smart meter wants to connect to the network or when it loses the connection and wants to reconnect to the network. The Wormhole attack places the attacker in a very powerful position allowing him to gain unauthorized access, disrupt routing, or perform a Denial-of-Service (DoS) attack [32].

There are a number of detection techniques for Wormhole attacks in the literature. Hu et al. [32] introduce the concept of Wormhole attacks and the concept of geographical and temporal packet leashes in order to detect them. For geographical leashes, their method requires that each node has accurate location information and loose clock synchronization. When a node receives packets, it computes the distance between previous nodes and itself by using a send/receive timestamps to derive the velocity between nodes. If the calculated distance falls above an

upper bound, the node decides that a Wormhole attack has taken place. For temporal leashes, each node should be accurately synchronized in time, and each packet should be delivered to the next node within the computed lifetime of the packet; otherwise, the next node should regard the path as a Wormhole link.

Song et al. [70] have considered the characteristic frequencies of links on network routes, finding that the frequencies of Wormhole links tend to be much higher than those of normal links. If a Wormhole attack is detected with the investigation, the scheme sends a data packet, and waits for an acknowledgment (ACK).

Chiu et al. [20] introduce a simple delay analysis approach, DelPHI, which calculates the mean value of the delay per hop for every possible route, based on sender initiation of detection packets, such as route requests (RREQ) and response by the receiver to every received detection packet. After collecting all responses, the sender computes the mean value of the delay per hop for each packet, with the assumption that a Wormhole would have more hops than its hop count would indicate. The scheme then analyzes computed delays to determine if there is a large difference between any two of the values.

Evans et al. [31] employed directional antennas in order to prevent Wormhole attacks. In their study, each node is equipped with a directional antenna; a sender broadcast a HELLO message bearing its identity, and receivers send back a response containing the direction from which the received HELLO message has come, allowing the sender to verify whether the response came from the same direction as the HELLO had been sent. The method is expensive, as each node needs to be equipped with a directional antenna.

Awerbuch et al. [9] have designed a new secure routing protocol, (ODSBR), in order to mitigate attacks that exploit Byzantine fault tolerance limits. To detect such Wormholes, the protocol requires that the destination returns an acknowledgment to the source for each data packet. If there is a fault in acknowledgment, the source will increase the weight of the link involved. Subsequently, links with higher weights will not be used to build routes. The disadvantage of this protocol is that nodes will be comparatively burdened and network traffic will be filled with enormous amount of acknowledgments.

Wang et al. [75] have developed a method for observing the occurrence of a Wormhole in a static sensor network. Their approach employs multidimensional scaling to reconstruct the network, detecting an attack by observing Wormhole links. Based on signal strength, each node estimates the distances to its immediate neighbors and sends this information to a centralized controller. By modeling a virtual position map of the sensors, the controller computes

a Wormhole indicator for each node.

Khalil et al. [39] have suggested a method for detection of Wormhole attacks for mobile ad hoc networks. In this method, information is gathered on neighbors within two hops of a node. As each node can overhear both the adjacent forwarder and its nexthop neighbor, it monitors two sets of packets forwarded, ensuring that both of these are the same. In using this approach, several monitors should be activated for links and equipped with buffers to store information on each packet delivered. The method requires a certified authority to verify exact location information on each node, and also requires that, whenever it moves, each node acquire authentication messages in order to transmit messages.

In this work, our method for detecting Wormhole attack makes use of hop count metric and is adopted from [78], [81]. Our approach is based on geographical locations of smart meters. As smart meters are static nodes and their locations remain unchanged, we can obtain their location easier compared to Mobile Ad-hoc nodes. One approach is to use global positioning system (GPS) to get the exact location of smart meters. Another approach for obtaining location of smart meters is when smart meters turn on and try to connect to the network during authentication process, they are required to register their location to the IDS nodes (i.e., collectors). Hence, geographical location can be used as a reliable measurement for estimating shortest path length between each smart meter and the corresponding collector in each NAN.

Using estimated shortest path, we can compute the estimated minimum hop, h_e count value for each flow from a smart meter to the collector. When a tunneling Wormhole attack is launched by malicious nodes, the number of hops indicated in the packet's field, h_r, will be less estimated minimum hop count, h_e, [66] as colluding malicious smart meters remove hops between the smart meters and collector.

All smart meters should send their data through the collector to the utility center. When receiving RREQs, the collector computes the expected hop counts between itself and the smart meter who has issued the RREQ using the location information. By calculating the shortest path length, the collector computes the estimated hop count between itself and the smart meter. If the received hop count value is smaller than the estimation, that is $h_r < \alpha h_e$, then the collector predicts a Wormhole attack and will mark the corresponding route as Wormhole link. Parameter α is adjustable based on the network characteristics. If some shortest routes have smaller hop count than the estimated value, it is with high probability that the route has gone through a Wormhole link as Wormhole link tends to bring nodes that are far away to be neighbors. Later we explain how we estimate the shortest path length. We enable the "destination

only flag"in RREQ messages so that all RREQs reach the collector to be examined by IDS.

5.3 Shortest Path Length Estimation

We adopt the Euclidean distance estimation model in [81] for our smallest hop count estimation. The model describes the relation of Euclidean distance and the corresponding hop count along the shortest path. Based on the model, given the Euclidean distance between the sender and receiver, the receiver (i.e., collector) can estimate the smallest hop count to the receiver.

The collector measures the minimum Euclidean distance between itself and a smart meter as

$$d = |l_d - l_s| \tag{5.1}$$

where l_d is the location of the collector and l_s is the location of the smart meter.

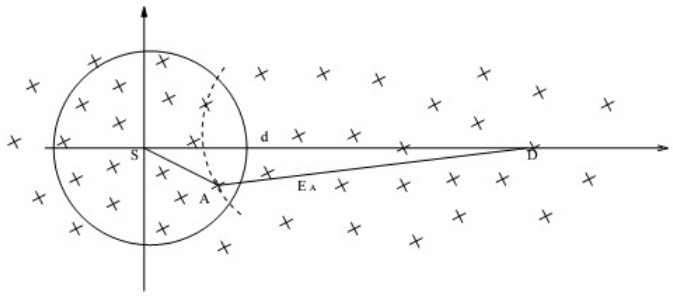

Figure 5.1: First hop estimation adopted from [81].

Fig. 5.1 shows a smart meter as the source (S) and the collector as Destination (D) in a NAN. We use arbitrary $(0, 0)$ as the coordinates of S and $(d, 0)$ as the coordinates of D in our calculations. The average density of the network is N_A nodes per unit area, then on the average there are $N_A * \pi r^2$ nodes in the set Φ within S's transmission range, r. For an arbitrary node i in Φ with coordinates (X_i, Y_i) the distance between i and D is:

$$e_i = \sqrt{(X_i - d)^2 + Y_i^2}$$

in which X_i and Y_i are random variables with a uniform distribution

$$f_{(X_i,Y_i)}(x_i, y_i) = \begin{cases} 1/\pi r^2, & P_i \in \Phi \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

Then the density function of E_i can be derived as

$$f_{E_i}(e_i) = \frac{2}{\pi r^2} e_i \cos^{-1} \frac{e_i^2 + d^2 - r^2}{2e_i d}$$

We assume there is a node A within S's transmission range and has the shortest Euclidean distance to D. A is selected for the next hop along the shortest path to the destination. Since A is the closest node to D, we have

$$E_A = min\{E_i | i \in \Phi\}$$

Accordingly, the density function of E_A can be derived as

$$f_{E_A}(e_A) = N_A \pi r^2 (1 - P_{E_i})^{N_A \pi r^2 - 1} f_{E_i}(e_i)$$

and the mean value is obtained

$$E(e_A) = d - r + \int_{d-r}^{d+r} (1 - P_{E_i}(e_i))^{N_A \pi r^2} de_i \tag{5.3}$$

where

$$P_{E_i}(e_i) = \int_{d-r}^{e_i} f_{E_i}(e_i) de_i$$

$E(e_A)$ gives us our first hop and the value of hop count is increased by 1. Recursively applying the above method, we can obtain the hop count of the shortest path from the source to the destination. For each recursion, we establish a new coordinate. For example, in Fig 5.2, A is located at $(0, 0)$ and D locates at $(E(e_A), 0)$. Then we can get the second hop B and $E(e_B)$. This procedure is repeated until the remaining distance to D (e.g. the distance between E and D in Fig. 5.2) is no longer than r.

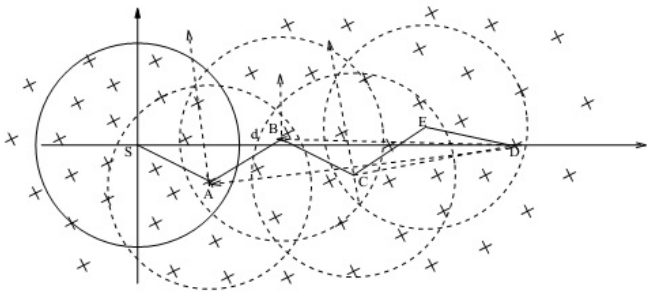

Figure 5.2: Recursive algorithm for computing minimum hop count adopted from [81].

Following algorithm describes the whole estimation process with regards to the mentioned model:

Algorithm 1 Hop count estimation of the shortest path between source and destination adopted from [78]

Input: l_s, l_d
Input: h_e
  $h_e \leftarrow 0$
  calculate d using equation 5.1
  while $d \geq r$ **do**
    calculate $E(e_A)$ using equation 5.3
    $++h_e$
    $d \leftarrow E(e_A)$
  end while
  $++h_e$

We model the shortest path length estimation algorithm in Maple 16 [33]. After modeling the estimation algorithm and obtaining estimated hop count in Maple, we need to plug it into our simulation model in OPNET.

5.4 Simulation Model

In this section, we explain the network topology and simulation settings for a smart grid NAN. We also develop our IDS solution to detect the Wormhole attack.

5.4.1 NAN Topology

Different areas of the North America were identified based on a census data. According to a report issued by the Congressional Research Service, the Federal Government classifies three major population types by geographic region: urban, suburban, and rural [22]. Urban environments are those with population densities greater than 1000 people per square mile (621 people per kilometer). Suburban is classified as those areas with population densities of 500-999 people per square mile (311-620 people per square kilometer), and rural is any region with a population density greater than 1 and less than 499 people per square mile (1-619 people per square kilometer). Utilizing Census data by population density per square mile, a determination was made to select the size and topology of the NAN.

Since the households per kilometer varies with the geographic locations and the density of population, there is a lower density of meters in lower household density areas (rural areas) and a higher density of meters in higher population density areas (urban). Thus the categorization into urban, suburban, and rural is used to determine region specific requirements of NAN deployments due to changes in the population density.
We have modeled 3 real geographical regions including suburban, shown in Fig. 5.3, rural shown in Fig. 5.4 and urban shown in Fig. 5.5.

Figure 5.3: Geographical image of the simulated suburb NAN

In a NAN, when a smart meter turns on, it starts discovering neighbors in order to connect to the NAN. After successful authentication using authentication schema discussed in Chapter 2, the smart meter needs to find the best path to the collector in order to send its data. As

Figure 5.4: Geographical image of the simulated rural NAN

Figure 5.5: Geographical image of the simulated urban NAN

mentioned before, we use AODV for discovering paths to the collector in the routing discovery phase. After discovering the best path to the collector, smart meters keep using the discovered path (i.e., building a path tree) unless there is a problem with the path. As a result, the routing discovery takes place only once when smart meters turn on unless they loose their connection to their path tree. Such a routing process seems as the most suitable solution due to the limited communication and computation capabilities of smart meter networks as discussed in [14].

For discovering the best route to the collector, smart meters broadcast route requests (RREQs) and use the best path among the received route replies (RREPs) as discussed in details in Chapter 2. As explained in Chapter 3, Wormhole attack can target routing discovery phase by tunneling RREQ from one end of Wormhole link to the other end. Wormhole attack targets smart meters such that they use the Wormhole link to send their data putting the attackers in a powerful position.

Fig. 5.6 shows the scenario in which tow colluding smart meters WH1 and WH2 are attacking the network. WH1 and WH2 are connected via an Ethernet link and they attack their neighbor smart meters and make them send their data through the Wormhole link.

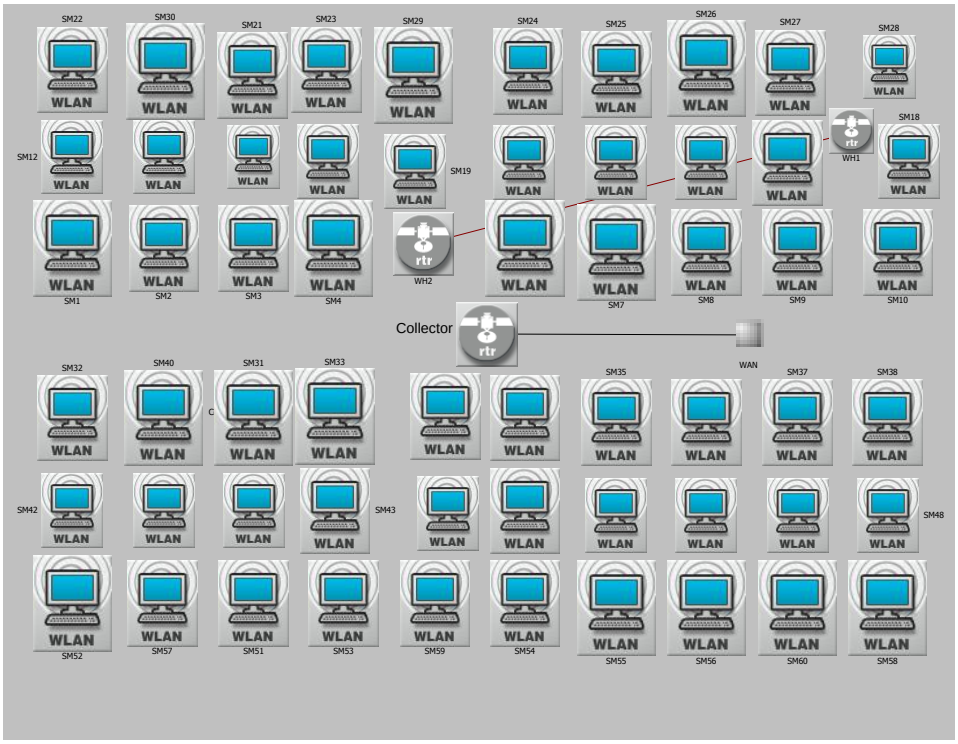

Figure 5.6: A NAN under Wormhole attack

In order to detect the Wormhole attack, we use shortest path length estimation algorithm discussed in section 5.3 to find the minimum hop count for each smart meter to reach the collector. As wormholes tend to bring nodes which are far away close to each other, the number of hop counts will be less than what it should be. Our method makes use of hop count field in the RREQ packets to determine the number of hop counts each message takes to reach

the collector. We enable the "destination only flag"in RREQ messages, which means only destination node should respond to the RREQs, so that all RREQs reach the collector to be examined by IDS. Moreover, we should integrate our analytical model with OPNET such that the collector only processes RREQs receiving from other nodes, not those that it is receiving from its own application layer.

5.5 Integrating Analytical Model and Simulation

Maple 16 provides an interface, called OpenMaple, which allows interaction with Maple engine from an external environment. More specifically, OpenMaple is a suite of functions that allows to access Maple algorithms and data structures in compiled C, Java, or Visual Basic programs. We develop a hybrid model by integrating our analytical model with simulation model using OpneMaple. To the best of our knowledge, this is the first time that Maple has been integrated into OPNET. The analytical model calculates the estimated minimum hop count based on Algorithm 5.1. The result will be used by IDS in order to detect Wormhole attack. Following code represents the analytical model implemented in C using OpenMaple interface.

Listing 5.1: Hop count calculation

```
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>
#include "maplec.h"

/* callback used for directing result output */
static void M_DECL textCallBack(void *data, int tag, char *output) {
        printf("%s\n", output);
}
static int hopCount(FLOAT64 x_src, FLOAT64 y_src, FLOAT64 x_dest, FLOAT64 y_dest,
 M_INT N, M_INT range);
int main(int argc, char *argv[]) {
        int hop = hopCount(-1089, -31.08, 1000, 0, 1, 609);
        printf("Hop Count is: %d", hop);
}

static int hopCount(FLOAT64 x_src, FLOAT64 y_src, FLOAT64 x_dest, FLOAT64 y_dest,
 M_INT N, M_INT range) {
        FLOAT64 euc_dis;
        M_INT hop_count;
        ALGEB F_E, P_E, E, d, l, r, a, b; /* Maple data-structures */
        char err[2048]; /* command input and error string buffers */
        MKernelVector kv; /* Maple kernel handle */
        MCallBackVectorDesc cb = { textCallBack, 0, /* errorCallBack not used */
        0, /* statusCallBack not used */
        0, /* readLineCallBack not used */
        0, /* redirectCallBack not used */
        0, /* streamCallBack not used */
        0, /* queryInterrupt not used */
        0 /* callBackCallBack not used */
        };
        hop_count = 0;
        /* initialize Maple */
```

```
34  if ((kv = StartMaple(0, NULL, &cb, NULL, NULL, err)) == NULL ) {
35      printf("Fatal error, %s\n", err);
36      return (1);
37  }
38  /* find out where maple is installed */
39  r = MapleKernelOptions(kv, "mapledir", NULL );
40  if (IsMapleString(kv, r))
41      printf("Maple directory = \"%s\"\n\n", MapleToString(kv, r));
42  EvalMapleStatement(kv, "restart:");
43  EvalMapleStatement(kv, "with(Student[Calculus1]):");
44  EvalMapleStatement(kv, "Digits:= 30:");
45
46  l = EvalMapleStatement(kv, "evalf(sqrt((x1-x2)^2+(y1-y2)^2)):");
47  MapleAssign(kv, ToMapleName(kv, "x1", TRUE), ToMapleFloat(kv, x_src));
48  MapleAssign(kv, ToMapleName(kv, "y1", TRUE), ToMapleFloat(kv, y_src));
49  MapleAssign(kv, ToMapleName(kv, "x2", TRUE), ToMapleFloat(kv, x_dest));
50  MapleAssign(kv, ToMapleName(kv, "y2", TRUE), ToMapleFloat(kv, y_dest));
51  EvalMapleStatement(kv, "x1:= floor(x1);");
52  EvalMapleStatement(kv, "y1:= floor(y1);");
53  EvalMapleStatement(kv, "x2:= floor(x2);");
54  EvalMapleStatement(kv, "y2:= floor(y2);");
55
56  l = MapleEval(kv, l);
57  euc_dis = MapleToFloat64(kv, l);
58  MaplePrintf(kv, "Init_Euc_Dis = %f", euc_dis);
59  while (euc_dis > range) {
60      F_E = EvalMapleStatement(kv, "(2/(Pi*r^2)) * e * arccos((e^2+d^2-r^2)/(2*e*d)):");
61      MapleAssign(kv, ToMapleName(kv, "r", TRUE), ToMapleInteger(kv, range));
62      MapleAssign(kv, ToMapleName(kv, "d", TRUE), ToMapleFloat(kv, euc_dis));
63      a = ToMapleName(kv, "a", TRUE);
64      MapleAssign(kv, a, F_E);
65      P_E = EvalMapleStatement(kv, "evalf(int(a,e=d-r..e)) assuming e=d:");
66      b = ToMapleName(kv, "b", TRUE);
67      MapleAssign(kv, b, P_E);
68      E = EvalMapleStatement(kv, "d-r + evalf(ApproximateInt((1-b)^(N*Pi*r^2),e=d-r..d+r
```

```
69                  , method=simpson)):");
70                  MapleAssign(kv, ToMapleName(kv, "N", TRUE), ToMapleInteger(kv, N));
71                  E = MapleEval(kv, E);
72                  d = MapleSelectRealPart(kv, E);
73                  euc_dis = MapleToFloat64(kv, d);
74                  MapleALGEB_Printf(kv, "\nNew Euc_dis = %a\n", d);
75                  EvalMapleStatement(kv, "F_E:='F_E':");
76                  EvalMapleStatement(kv, "P_E:='P_E':");
77                  EvalMapleStatement(kv, "a:='a':");
78                  EvalMapleStatement(kv, "b:='b':");
79                  EvalMapleStatement(kv, "E:='E':");
80                  EvalMapleStatement(kv, "d:='d':");
81                  ++hop_count;
82          }
83          ++hop_count;
84          StopMaple(kv);
85          return (hop_count);
86  }
```

In order for the collector to calculate the minimum hop count from each smart meter, it requires to know the location of each smart meter. In reality, smart meter locations can be registered in the collectors ahead of time (e.g., when smart meters are registered within the utility center). However, in order to support high degree of scalability in our simulation, we require each smart meter to send its location information along with their RREQ packets. To this end, we have modified the RREQ packet structure in OPNET to carry the location information (fields *x_pos* and *y_pos* as framed in red box in Fig. 5.7). When smart meters want to find a path to the collector, they put the location information in the RREQs, sign and broadcast them. When the IDS in the collector receives the RREQs, it starts examining them. After calculating the estimated hop count, h_e, using the location information, the IDS checks the legitimacy of the hop count in the received RREQ packets, h_r, using following equation:

$$h_r > \alpha h_e \tag{5.4}$$

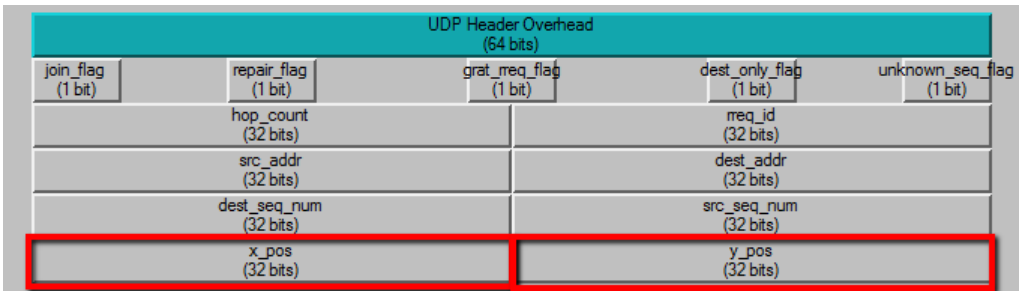

Figure 5.7: Modified AODV RREQ packet

If the above condition is satisfied, the source smart meter is not under Wormhole attack, otherwise IDS flags the smart meter as attacked. Parameter α is adjustable to the network characteristics.

Fig. 5.8 represents the relation between h_e and real minimum hop count, h_{real}. It should be noted that h_{real} is different from h_r because h_{real} is only the expected hop count for each smart meter merely based on the distance while h_r is affected by the network operation.

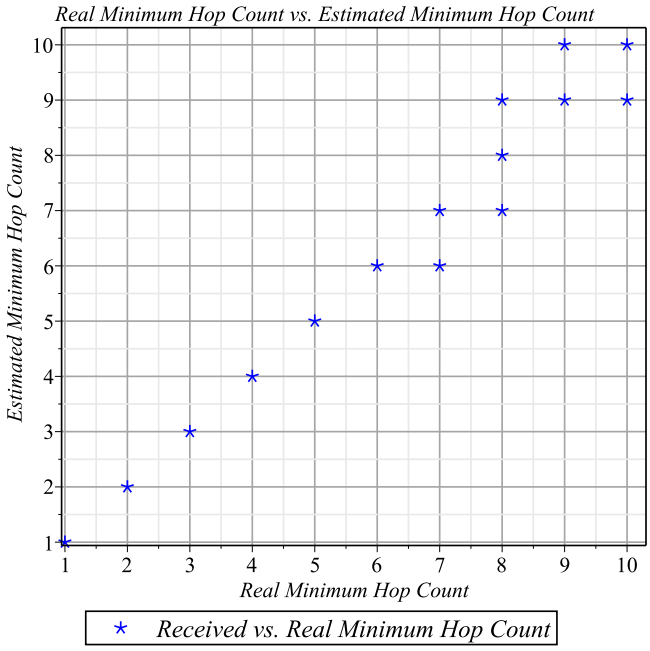

Figure 5.8: Real minimum hop count vs. estimated minimum hop count

5.6 Simulation Setup

In this section we discuss our simulation scenarios in terms of topology, node configuration, and attack description. We have used OPNET 17.1 to simulate the NAN and Wormhole attack scenarios. We have simulated the three regions discussed in section 5.4.1 including suburb, rural and urban areas. Parameter α in our simulation is set to 1.

5.6.1 Suburb Area

Fig. 5.9 presents the network topology of a NAN for suburb area of $8 \times 2Km^2$ in our simulation model according to Fig. 5.3. The chosen region allows placing meters uniformly and placing the collector at the center of the region. We position smart meters in a way that in $1km^2$ area $N_A = 9$ in Formula 5.3.

The NAN is a wireless mesh network of smart meters which are connected to the collector using single/multihop paths for relaying their metering data. There are maximum of 85 smart meters in the NAN including colluding Wormhole nodes.

Figure 5.9: Suburb NAN

Table 5.1 represents the smart meters simulation configuration according to [4], [7]. We suppose the nodes' transmission in the NAN is perfect and signals propagate through open space, with no environmental effects. However, there are a couple of propagation models in OPNET which are neither free nor in the scope of this work.

Fig. 5.10 represents the minimum hop count distribution in case of no attack scenario. We

Table 5.1: Suburb Smart Meters Configuration.

Parameter	Value
Physical Channel Property	802.11g
Data Rate	24 Mbps
Transmission Power	0.005 W
Receiver Sensitivity	-95 dBm
Meter Reading Payload	1 KB
Meter Reading Transmission Frequency	30 min
Density (N_A)	9 per $1km^2$

can see that, for example, one smart meter is 11 hops away from the collector or ten smart meters are 7 hops away and so on. Also the largest minimum hop count in this NAN is 12.

We have designed Wormhole attacks by connecting malicious nodes by an Ethernet link. The Wormholes have a wireless interface to connect to the mesh network for communicating with the other NAN nodes.

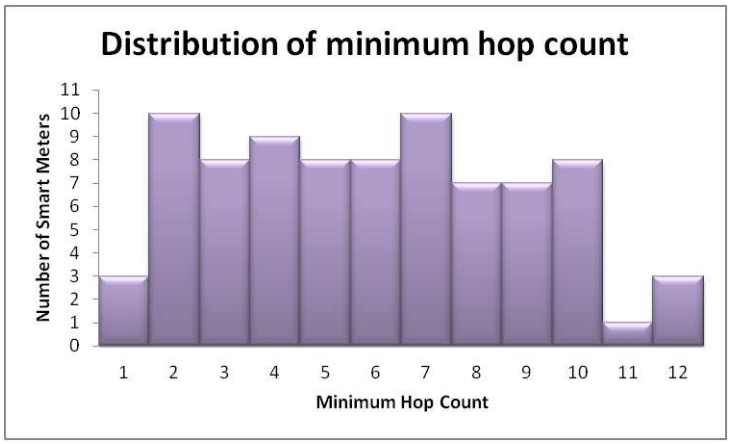

Figure 5.10: Distribution of minimum hop counts in suburb NAN

We have simulated different scenarios by changing the location of Wormholes in order to affect different parts of the network. We intend to observe how our IDS performs with respect to these scenarios. Figs. 5.11, 5.12, 5.13 depict the scenarios where the position of Wormholes are changed. We refer to these Wormhole attacks as Pair attacks as there is a pair of attackers. Here we call the attack presented in Fig.5.14 Delta Wormhole which comprises three colluding attackers where one of them is connected to two others aiming to attack a wider range of smart

Figure 5.11: Wormhole attack: Pair 1 in suburb NAN

Figure 5.12: Wormhole attack: Pair 2 in suburb NAN

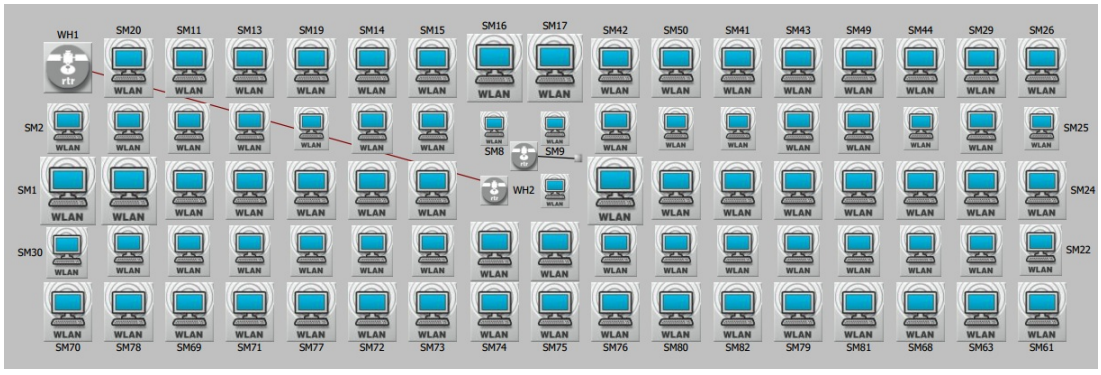

Figure 5.13: Wormhole attack: Pair 3 in suburb NAN

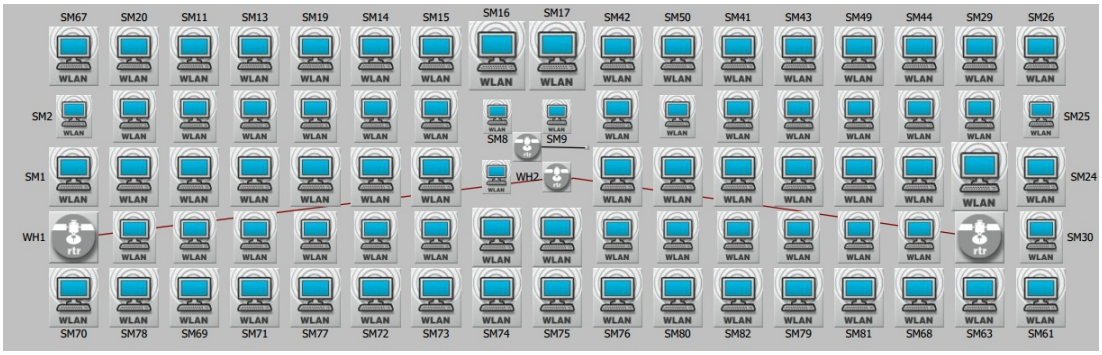

Figure 5.14: Wormhole attack: Delta in suburb NAN

meters.

The effect of Wormhole attacks on hop count distribution and also the results of IDS for detecting wormhole attacks in suburb NAN are given in Section 5.7.1.

5.6.2 Urban Area

Fig. 5.15 demonstrates the network topology of a NAN for urban area of $4 \times 1Km^2$ in our simulation model according to Fig. 5.5. We have placed smart meters uniformly and position the collector at the center of the region. There are maximum of 89 smart meters in the NAN including colluding Wormhole nodes. We place the smart meters such that there are 25 smart meters per $1km^2$ which means $N_A = 25$ in Formula 5.3. The simulation configuration of smart meters for the urban NAN is shown in Table 5.2.

Table 5.2: Urban smart meters configuration.

Parameter	Value
Physical Channel Property	802.11g
Data Rate	48 Mbps
Transmission Power	0.005 W
Receiver Sensitivity	-95 dBm
Meter Reading Payload	2 KB
Meter Reading Transmission Frequency	15 min
Density (N_A)	25 per $1km^2$

Fig. 5.16 represents the minimum hop count distribution in case of no attack scenario in the urban NAN. It can be seen that there are eight smart meters with the hop count of 1 (one hop

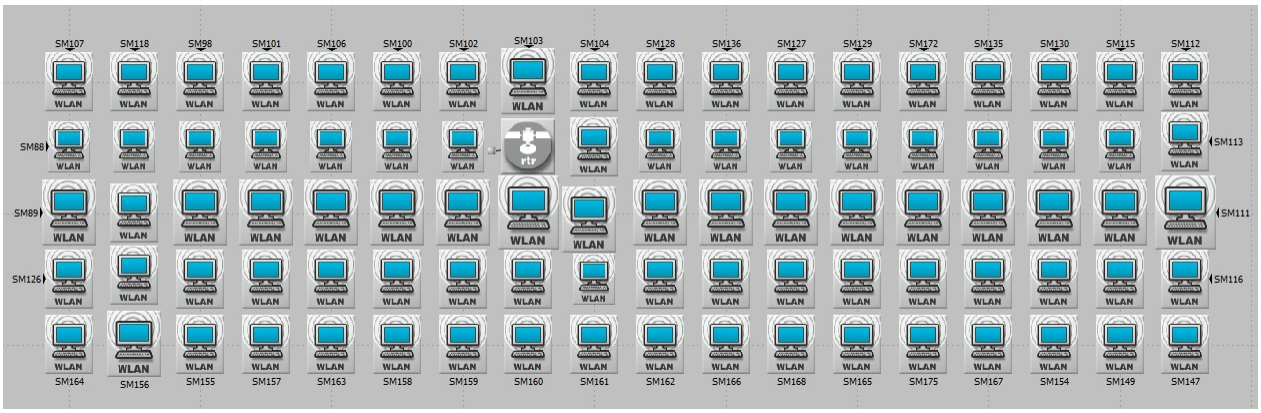

Figure 5.15: Urban NAN

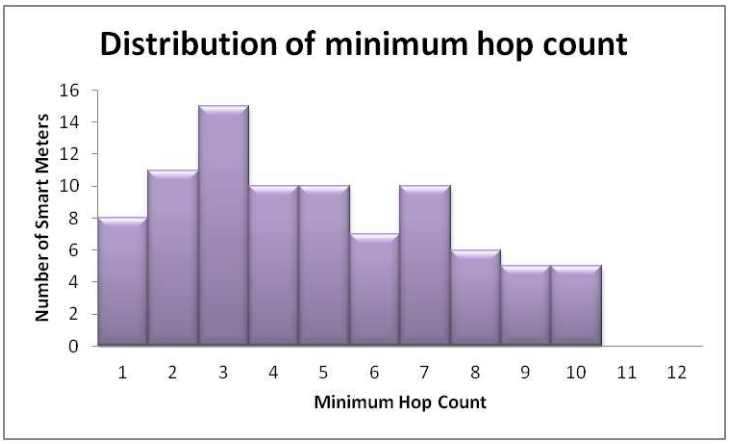

Figure 5.16: Distribution of minimum hop counts in urban NAN

neighbors of the collector) and five smart meters which are 10 hops away from the collector. Figs. 5.17, 5.18, 5.19, and 5.20 depict the Wormhole attack scenarios considered for the urban area.

5.6.3 Rural Area

Fig. 5.21 presents the network topology of a NAN for rural area of $2.7 \times 10.8 Km^2$ in our simulation model according to Fig. 5.4. Overall there are 60 smart meters and their simulation configuration are demonstrated in Table 5.3.

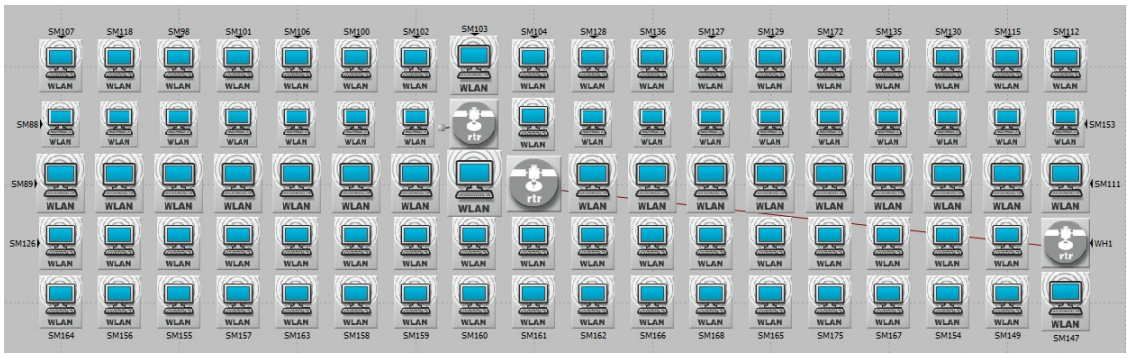

Figure 5.17: Wormhole attack: Pair 1 in urban NAN

Figure 5.18: Wormhole attack: Pair 2 in urban NAN

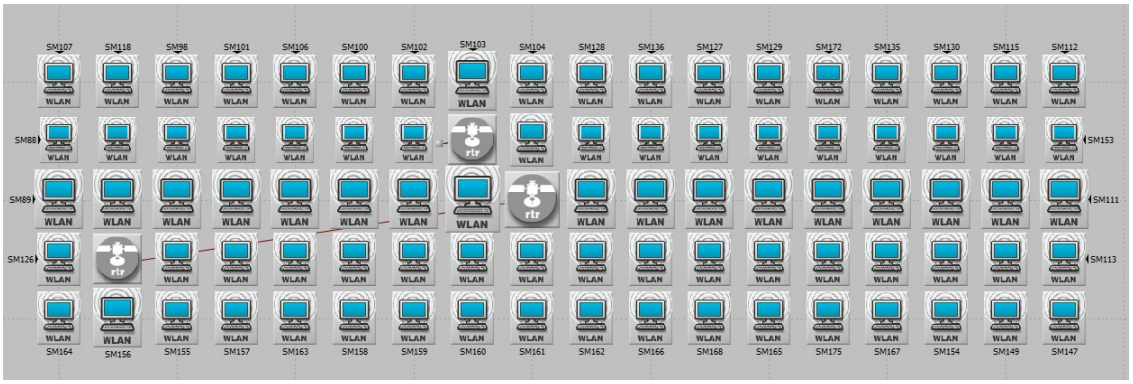

Figure 5.19: Wormhole attack: Pair 3 in urban NAN

Fig. 5.22 represents the minimum hop count distribution in case of no attack scenario. We can see that, for instance, 8 smart meter are 4 hops away from the collector or 1 smart meter is

Figure 5.20: Wormhole attack: Delta in urban NAN

Figure 5.21: Rural NAN

Table 5.3: Smart meters configuration in rural Area .

Parameter	Value
Physical Characteristic	802.11g
Data Rate	18 Mbps
Transmission Power	0.005 W
Receiver Sensitivity	-95 dBm
Meter Reading Payload	1 KB
Meter Reading Transmission Frequency	30 min
Density (N_A)	3.25 per km^2

12 hops away and so on. From hop count distribution of three areas suburb, urban and rural, we can see that the number of smart meters with smaller hop count is larger in urban, suburb and rural area respectively. This is because the urban area is denser than suburb area and suburb

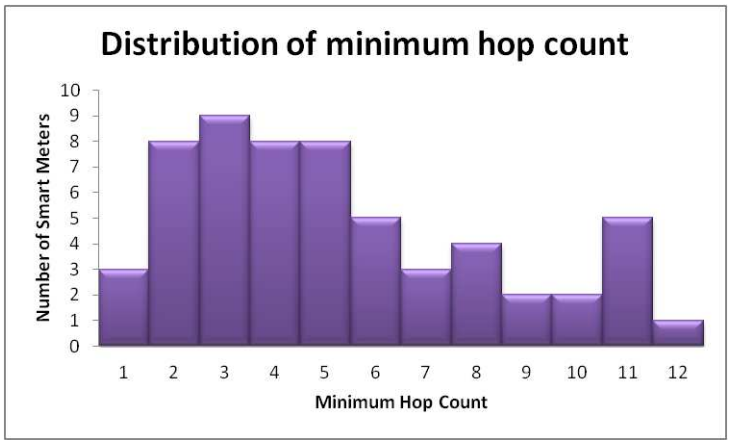

Figure 5.22: Distribution of minimum hop counts in rural NAN

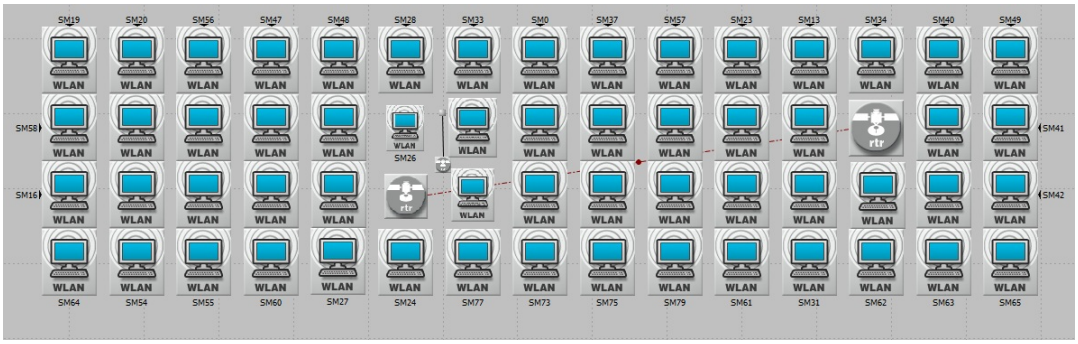

Figure 5.23: Wormhole attack: Pair 1 in rural NAN

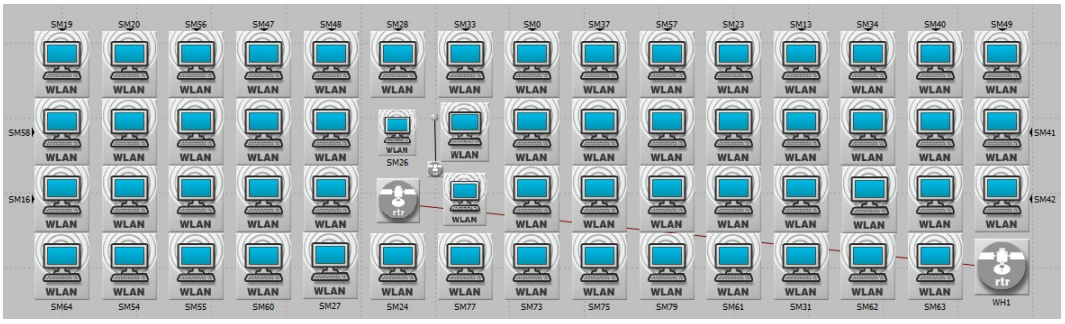

Figure 5.24: Wormhole attack: Pair 2 in rural NAN

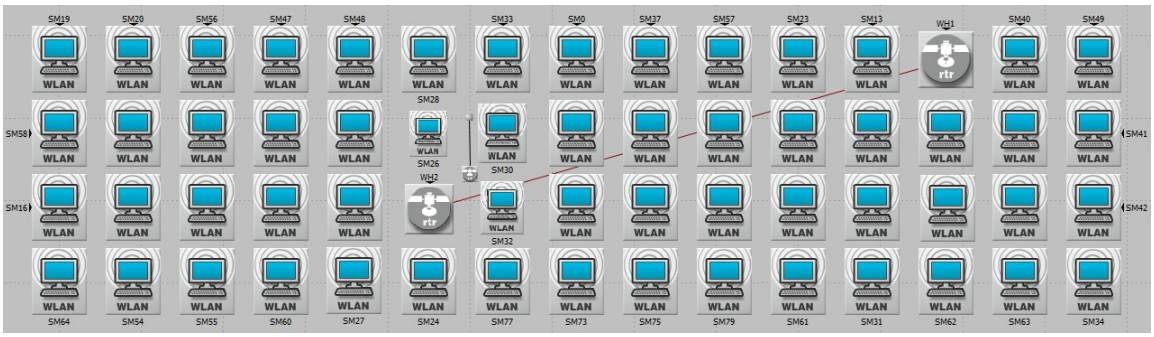

Figure 5.25: Wormhole attack: Pair 3 in rural NAN

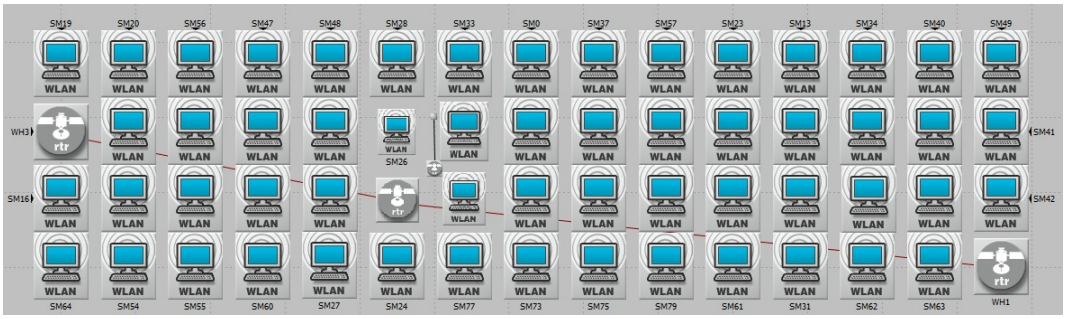

Figure 5.26: Wormhole attack: Delta in rural NAN

area is denser than rural area and there more smart meters in the vicinity of the collector. The largest minimum hop count is 12. The attack scenarios are illustrated in Figs. 5.23, 5.24, 5.25, and 5.26 which depict the Pair 1, Pair 2, Pair 3 and Delta attacks respectively.

5.7 Results From Simulation Experiments

In this section, the simulation results for suburban, rural and urban NAN scenarios are presented. We demonstrate our IDS performance for detecting Wormhole attacks by measuring false positive (FP), false negative (FN), and detection rate (DR) which were discussed in Chapter 1. The simulation time was set to 12 hours for all scenarios.

5.7.1 Suburb NAN

The effects of Wormhole attacks on hop count distribution in suburb area are presented in Figs. 5.27, and 5.28. As can be seen Wormhole attacks decrease the number of larger hop counts and add up to the number of smaller hop counts in all attack scenarios. We have the largest decrease in hop count distribution in Delta attack because two parts of the NAN are under attack. The IDS can also be aware of the possible number of colluding Wormholes in the NAN using the real hop count distribution. More specifically, if the hop counts of nodes from two far corner of the network have decreased at the same time, the IDS will conclude that there are more than two attackers targeting the network.

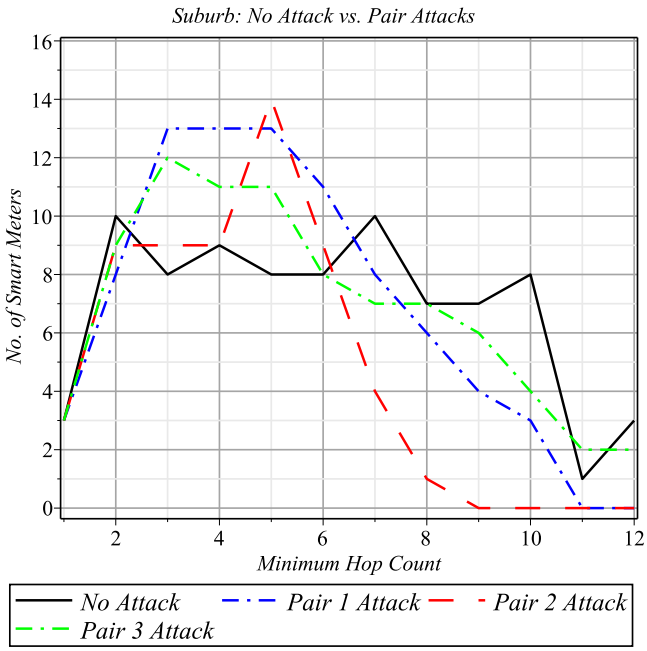

Figure 5.27: Distribution of minimum hop counts of no-attack, Pair 1, Pair 2 and Pair 3 attack scenarios in suburb NAN

The results of IDS detection for suburb area is presented in Table 5.4. The simulation time was set to 12 hours. The number of smart meters is 85 including attackers. The results embrace no attack scenario and scenarios in Figs. 5.11, 5.12, 5.13 and 5.14 presenting Pair 1, Pair 2, Pair 3, and Delta attacks .

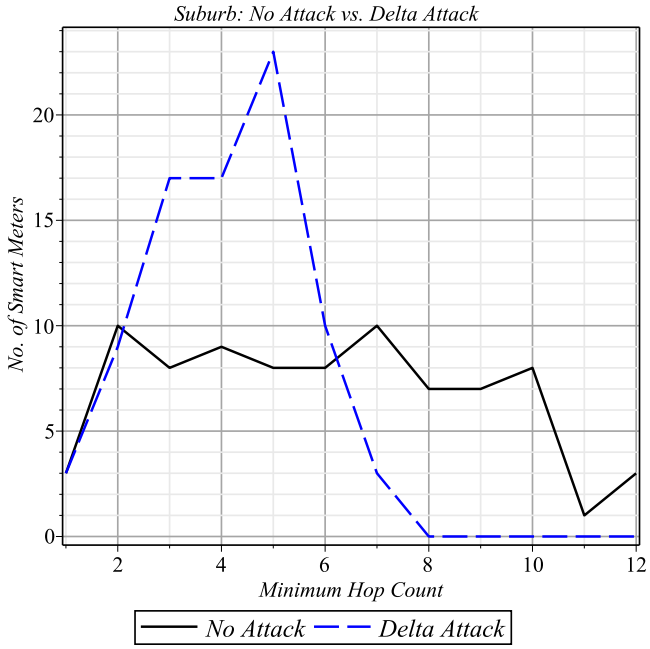

Figure 5.28: Distribution of minimum hop counts of no-attack and Delta attack scenarios in suburb NAN

Table 5.4: IDS result for suburb NAN

Wormhole Attack Type	FP	FN	DR	No. of Attackers
No Attack	1%	N/A	N/A	0
Pair 1	7%	5%	95%	2
Pair 2	6%	6%	94%	2
Pair 3	5%	5%	95%	2
Delta	3%	8%	92%	3
Overall	4.4%	6%	94%	2

5.7.2 Urban NAN

Figs. 5.29 and 5.30 show how the Wormhole attacks discussed in Section 5.6.2 affect the hop count distribution in urban NAN. As can be seen in these Figs., Wormhole attacks tend to decrease the hop counts between smart meters and the collector by using their short cut link. The amount of decline gets even larger in case of Delta attack. The results of IDS performance have been presented in Table 5.5.

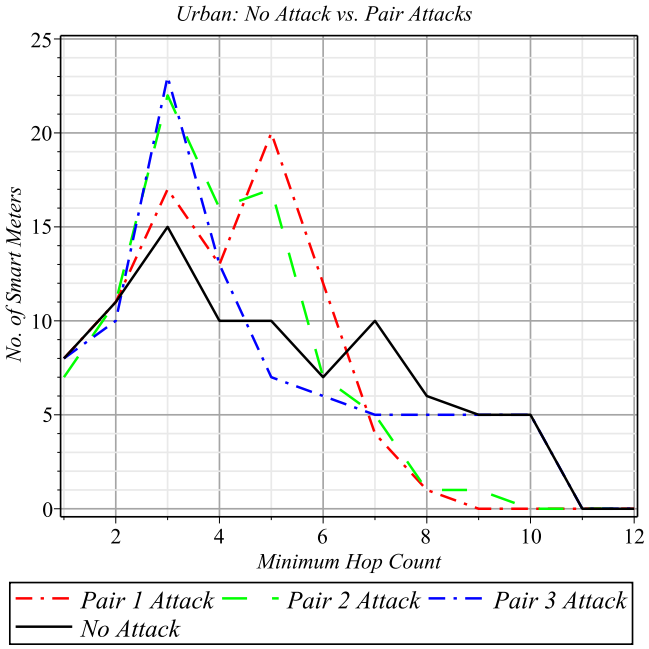

Figure 5.29: Distribution of minimum hop counts of no-attack, Pair 1 and Pair 2 attack scenarios in urban NAN

Table 5.5: IDS result for urban NAN

Wormhole Attack Type	FP	FN	DR	No. of Attackers
No Attack	4%	N/A	N/A	0
Pair 1	7%	5%	95%	2
Pair 2	5%	0%	100%	2
Pair 3	5%	6%	94%	2
Delta	3%	2.8%	97%	3
Overall	4.8%	3.45%	96.5%	2

5.7.3 Rural NAN

The effect of Wormhole attacks on hop count distribution are presented in Figs. 5.31 and 5.28 for rural region. As same as the suburb and urban scenarios, we can see that the Wormhole attacks tend to increase the number of smaller hop counts reducing the number of larger hop counts in the rural area. We have examined our IDS performance in a simulation time of 12 hours for detecting the discussed attacks in rural NAN.The number of smart meters are 60

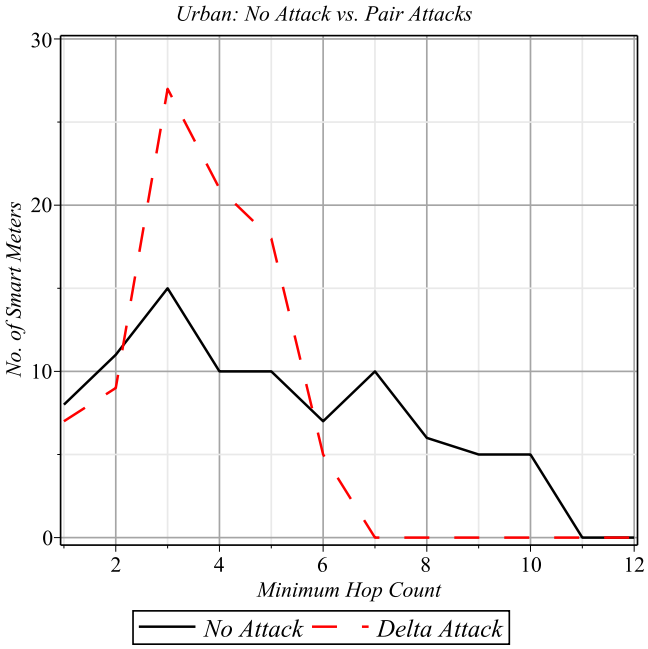

Figure 5.30: Distribution of minimum hop counts of no-attack and Delta attack scenarios in urban NAN

including attackers. The results are presented in Table 5.6.

Table 5.6: IDS result for rural NAN

Wormhole Attack Type	FP	FN	DR	No. of Attackers
No Attack	0%	N/A	N/A	0
Pair 1	0%	5%	95%	2
Pair 2	0%	8%	92%	2
Pair 3	0%	6%	94%	2
Delta	0%	4%	96%	3
Overall	0%	5.7%	94.2%	2

From Tables 5.4, 5.5, and 5.6 it can be seen that the FP rate gets increased with density. Urban area with the average of 4.8% has the highest FP rate while rural area has average FP of 0%. This lies in the fact that when density, N_A, gets bigger in Formula 5.3, the estimation hop count tends to be larger, therefore, in the equation 5.4, estimation hop count, $h_e$becomes larger than received hop count, h_r. As a result, the IDS might detect more normalities as attack

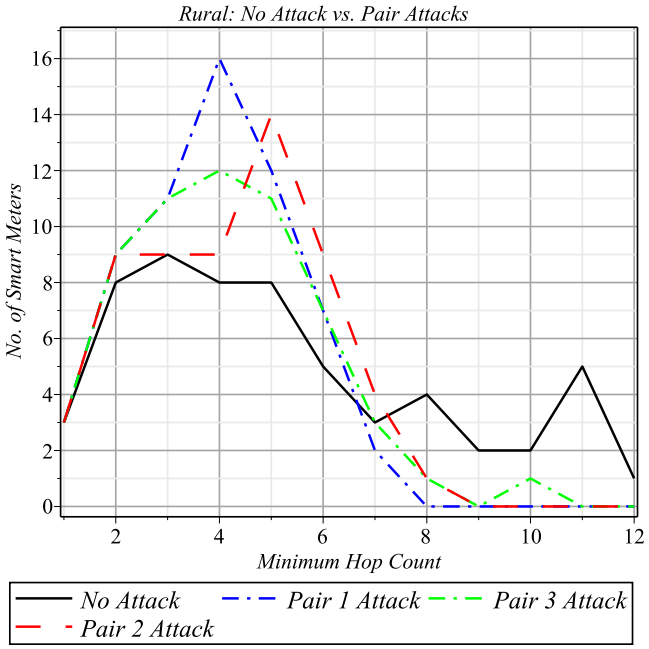

Figure 5.31: Distribution of minimum hop counts of no-attack, Pair 1, Pair 2, and Pair 3 attack scenarios in rural NAN

which leads to a larger FP rate.

On the other hand, from Tables 5.4, 5.5, and 5.6, FN is smaller in denser area, i.e., urban area, than suburb and rural areas. The reason is that when h_e tends to be larger than h_r, there are less number of cases where h_e becomes less than h_r which results in a lower FN rate in the urban area compared to rural and suburb areas. Therefore, depending on the network topology, security concerns, and administrative preferences, parameter α can be adjusted to obtain desirable FP, FN, and DR rates.

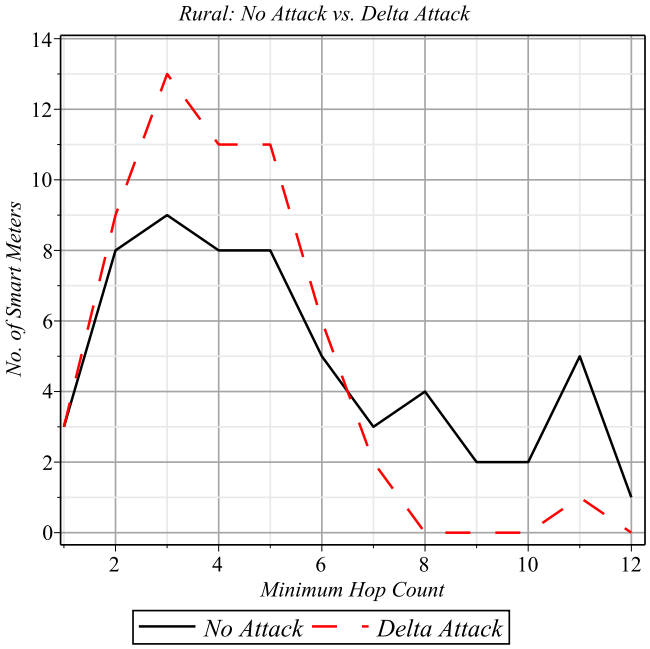

Figure 5.32: Distribution of minimum hop counts of no-attack and Delta attack scenarios in rural NAN

Chapter 6

Summary and Future Work

Smart grid intends to enhance the current energy delivery systems in terms of efficiency, reliability, and economics. A two-way digital communication network is an essential requirement for developing the smart grid. There is currently an ongoing debate surrounding what would be the best communication technology for smart grid realization. Advanced metering infrastructure (AMI) is a critical part of smart grid that is responsible for automatic reading-related functions. AMI requires the highest level of security and a comprehensive architecture with security built in as it deals with sensitive type of data such as customer meter readings, financial transactions information, utility commands and etc. While deploying AMI becomes more and more widespread, the security threats also grow in parallel even at a faster pace. Therefore, security mechanisms should be in place to protect such a critical infrastructure from malicious attacks. As a second wall of defense, intrusion detection systems are of paramount importance to protect the AMI if security mechanisms cannot prevent the attacks.

In this thesis, we discussed communication technologies and requirements of smart grid. We explained AMI in smart grid along with its security requirements. We focused on the wireless mesh network that exists in AMI as neighborhood area network (NAN). We considered the NAN characteristics and its requirements. The security threats that can target the NAN have been identified by studying the state of the art security mechanisms. Moreover, we highlighted the intrusion detection systems that have been so far proposed for AMI and NAN.

In this work, we proposed an IDS taking into account the specifications and requirements of the NAN. Our solution is specifically designed to detect Wormhole attack which can have severe effects on the network. We have studied the impacts of Wormhole attack in depth and

simulated different types of Wormhole attacks by changing the location and the number of attackers. Our detection mechanism takes advantages of an analytical model which calculates the estimation hop count of RREQ messages being transmitted in the NAN. We used Maple for implementing our analytical model. By integrating the analytical model with the simulation model in OPNET Modeler, we evaluated our IDS for three different areas including rural, suburb and urban scenarios. The detection rates showed that Our IDS performs well in detecting wormhole attacks in all three scenarios. The FP rate in urban area was the highest due to the density and high number of nodes while the FN rate had the highest value in rural area because of less number of nodes in the network.

Future Work

A number of modifications and extensions can be made to enhance the the proposed IDS:

- In our IDS, we only considered automatic meter reading traffic in the NAN. Automatic reading traffic is an up-link traffic (from smart meters to the utility center) and is only a one-way transmission. Another application traffic that can be considered is demand response (DR). DR traffic is used to manage customer consumption of electricity in response to supply conditions, for example, having electricity customers reduce their consumption at critical times or in response to market prices. Hence, a down-link traffic (from utility center to customers) can also be added to the NAN traffic and IDS performance can be examined in both down-link and up-link directions with various load sizes. In addition, remote disconnects, firmware updates and etc are other examples of down-link traffic that can also be considered.

- In our simulation, we have used uniform distribution for placing smart meters. One future direction to this work would be to consider different distribution of smart meters placement depending on real arrangement of smart meters in the NAN.

- The main source of error in our IDS was related to the cases that the estimated hop count were equal to the received hop count. As a result, the IDS might fail in detecting real attacks. One approach that can solve this problem is to consider packet travel time in the IDS.

- Another improvements that can be made to our IDS is to add propagation model to the NAN. Such a modification will bring about the ability to evaluate the performance of the whole network along with the IDS option.
- Since we have established the NAN infrastructure and the IDS module, other type of attacks can be considered and evaluated on top of our proposed solution. In other words, due to modular design of the simulation model and IDS module, our solution can be considered as a base model to study the NAN and it's security threats.

Appendix 1

IDS Source Code

In this appendix, we present the main parts of IDS source code. The IDS has been implemented using Proto-C in OPNET Modeler. Proto-C provides a flexible platform that has the ability to model a wide range of systems. The language provides specific support by adopting a state-transition approach which is well-suited to discrete event systems, and by supplying a number of Kernel Procedures (KPs) that are oriented toward network and distributed systems modeling. At the same time however, Proto-C preserves generality by incorporating all the capabilities of the C/C++ programming language.

1.1 Route Request Structure

Listing 1.1: Route Request Structure

```
1  #include <ip_addr_v4.h>
2
3  #define MAX_MICRO_NAN_SIZE 128
4  #define MAX_NAME_SIZE 64
5  #define MAX_HOP_COUNT 30
6  #define SUBURB_SIZE  82
7  #define URBAN_SIZE  86
8  #define RURAL_SIZE  56
9
10 #define WH_IDS_IS_ENABALED OPC_TRUE
```

```
#define APP_IDS_IS_ENABALED OPC_FALSE

/* Smart Meters route request info. */
typedef struct
        {
        int     hop_count;
        int     rreq_id;
        InetT_Address  src_addr;
        InetT_Address  dest_addr;
        InetT_Address  prev_addr;
        double    x_pos;
        double    y_pos;
        char    src_name[MAX_NAME_SIZE];
        char     prev_name[MAX_NAME_SIZE];
        Boolean    is_under_attack;
        int     est_hop_count;
        } SM_RR;
```

1.2 Modification of AODV

Listing 1.2: init function of modified AODV

```
static void aodv_rte_sv_init (void)
        {
        /** Initialize the state variables **/
        FIN (aodv_rte_sv_init (void));

        /* Access the module data memory */
        module_data_ptr = (IpT_Rte_Module_Data*) op_pro_modmem_access ();

        /* Obtain own module ID */
        own_mod_objid = op_id_self ();

        /* Obtain the node's objid. */
```

```
13      own_node_objid = op_topo_parent (own_mod_objid);
14
15
16      // Added by Nasim
17      op_ima_obj_attr_get (own_node_objid, "name", &coll_name);
18      op_ima_obj_attr_get (own_node_objid, "x position", &coll_x_pos);
19      op_ima_obj_attr_get (own_node_objid, "y position", &coll_y_pos);
20
21      //--------------------------------
22
23      /* Obtain own process handle. */
24      own_prohandle = op_pro_self ();
25
26      /* Obtain parent process handle */
27      parent_prohandle = op_pro_parent (own_prohandle);
28
29      /* Own process ID */
30      own_pro_id = op_pro_id (own_prohandle);
31
32      /* Parent process ID */
33      parent_pro_id = op_pro_id (parent_prohandle);
34
35      /* Set up the display string. */
36      sprintf (pid_string, "aodv_rte PID (%d)", own_pro_id);
37
38      /* Initailize the identification values */
39      route_request_id = 0;
40      sequence_number = 0;
41
42      /* Initailize the variables used to keep track of the rate */
43      last_route_error_sent_time = 0.0;
44      num_route_errors_sent = 0;
45      last_route_request_sent_time = 0.0;
46      num_route_requests_sent = 0;
47      last_broadcast_sent_time = 0.0;
```

```
48
49          /* Obtain a handle to the global statistics */
50          global_stat_handle_ptr = aodv_support_global_stat_handles_obtain ();
51
52          /* Assign a unique name to this AODV process. It will include */
53          /* the name of the routing protocol, which in this case is AODV */
54          /* and the AS number of this process. Once the unique name is */
55          /* obtained, this process can be registered in IPs list of all */
56          /* routing processes.          */
57          aodv_proto_id = IP_CMN_RTE_TABLE_UNIQUE_ROUTE_PROTO_ID
58                                                  (IPC_DYN_RTE_AODV, IPC_NO_MULTIPLE_PROC);
59          Ip_Cmn_Rte_Table_Install_Routing_Proc
60                      (module_data_ptr->ip_route_table, aodv_proto_id, own_prohandle);
61
62          /* Create a /32 subnet mask for entries in the route table */
63          subnet_mask = inet_smask_from_length_create (32);
64
65          /* Create a /128 subnet mask for entries in the route table */
66          subnet_mask_ipv6 = inet_smask_from_length_create (128);
67
68          /* External routes are handled only by MANET gateways.  */
69          if (ip_manet_gateway_enabled(module_data_ptr))
70                  {
71                  /* Initialize the cache table used to store routes to   */
72                  /* external hosts (non-MANET hosts).      */
73                  external_routes_cache_table_ptr = manet_ext_rte_cache_table_init ();
74                  }
75
76          FOUT;
77          }
78
79  ////////////////////////////////////////
80  /* Initializa the state variables */
81
82  static void nasim_aodv_rte.init.Enter(void)
```

```
83  {
84  aodv_rte_sv_init ();
85
86  /* Register the local statistics */
87  aodv_rte_local_stats_reg ();
88
89  /* Parse the attributes and create  */
90  /* and initialize the various buffers */
91  aodv_rte_attributes_parse_buffers_create ();
92
93  /* Register the AODV Routing process as a higher layer in IP */
94  Ip_Higher_Layer_Protocol_Register ("aodv", &higher_layer_proto_id);
95
96  /* Add directly connected routes only if */
97  /* this is not a MANET gateway.    */
98  if (!ip_manet_gateway_enabled(module_data_ptr))
99          {
100         aodv_rte_add_directly_connected_routes ();
101         }
102
103
104 /* added by Nasim*/
105 if(WH_IDS_IS_ENABALED)
106         {
107         int i;
108   /* initialize Maple */
109         if ((kv = StartMaple(0, NULL, NULL, NULL, NULL, err)) == NULL ) {
110                 printf("Fatal error, %s\n", err);
111                 //FRET (1);
112         }
113
114    /* find out where maple is installed */
115         r = MapleKernelOptions(kv, "mapledir", NULL );
116         if (IsMapleString(kv, r))
117             printf("Maple directory = \"%s\"\n\n", MapleToString(kv, r));
```

```
118
119         rreq_index = 0;
120         density = 3.25;//rur=3.25; // sub=9; urb=25,
121         range = 1.089;//0.396;//rur=1.089; // range of node per kilometers
122
123         /* initialize the array*/
124         for(i=0;i<MAX_HOP_COUNT;i++)
125                 hop_count_array[i] = 0;
126
127         sim_round = 1; // initial for first round of simulation
128
129 }
130 FOUT;
131 }
```

1.3 Util Functions

Listing 1.3: IDS Utility Functions

```
1  static void print_ipv4(IpT_Address ip)
2      {
3       unsigned char bytes[4];
4       bytes[0] = ip & 0xFF;
5       bytes[1] = (ip >> 8) & 0xFF;
6       bytes[2] = (ip >> 16) & 0xFF;
7       bytes[3] = (ip >> 24) & 0xFF;
8       printf("%d.%d.%d.%d\n", bytes[3], bytes[2], bytes[1], bytes[0]);
9  }
10
11 // This function calculate the distribution of hop count for smart meters
12 static void calculate_hc_dis(void)
13          {
14          int i;
15          FIN(calculate_hc_dist(<args>));
```

```
16
17          for(i=0;i<rreq_index;i++)
18                  if(sm_rr[i].hop_count < MAX_HOP_COUNT)
19                          hop_count_array[sm_rr[i].hop_count]+=1;
20              else
21                          printf("###################################### ERROR\n");
22
23          printf("************************************************\n");
24          printf("Number of Smart Meters = %d\n", rreq_index-1);
25          printf("Min Hop Count ------> No. of Nodes\n");
26          for(i=1;i<MAX_HOP_COUNT;i++)
27                  printf(" %d ------> %d \n", i,hop_count_array[i]);
28          printf("************************************************\n");
29          FOUT;
30  }
31
32  ////////////////////////////////////////////////////////////////////////////
33
34  // This function checks whether the rreq has been processed before or not.
35  static Boolean rreq_exist(SM_RR smRR)
36          {
37      int i;
38          unsigned char last_ip_byte;
39
40          FIN(rreq_exist(<args>));
41
42          last_ip_byte = smRR.dest_addr.address.ipv4_addr & 0xFF;
43
44          if (last_ip_byte == 255 || is_not_sm(smRR.src_name))
45                  {
46            //printf("This is a Broadcast Packet; no need to be processed.\n");
47            FRET (OPC_TRUE);// do nothing
48          }
49
50          for(i=0;i<rreq_index;i++)
```

```
51                  if(sm_rr[i].src_addr.address.ipv4_addr == smRR.src_addr.address.ipv4_addr &&
52                          sm_rr[i].hop_count <= smRR.hop_count)
53                          FRET (OPC_TRUE);
54
55          FRET (OPC_FALSE); // not exist
56  }
57
58  ////////////////////////////////////////////////////////////
59  static void add_rr_info(SM_RR sm_rreq)
60          {
61          int i;
62          Boolean new_request;
63          FIN(add_rr_info(<args>));
64
65          new_request = OPC_TRUE;
66          for (i=0;i<rreq_index;i++)
67                  if(sm_rr[i].src_addr.address.ipv4_addr == sm_rreq.src_addr.address.ipv4_addr)
68                          {
69                          sm_rr[i].hop_count=sm_rreq.hop_count;
70                          new_request = OPC_FALSE;
71                          }
72
73          if(new_request) // if this is a new request
74              sm_rr[rreq_index++] = sm_rreq;
75
76          /* Check if examined all nodes once */
77
78          if(rreq_index >= RURAL_SIZE)
79                  {
80                  if(sim_round==3) // just print and calculate hop counts for the third round
81                      {
82                          print_all_sms_info();
83                          calculate_hc_dis();
84                  sim_round++;
85                  }
```

```
86                  else
87                          sim_round++;
88                  }
89          FOUT;
90  }
91
92  ////////////////////////////////////////////////////////
93  static Boolean is_not_sm (char* name)
94          {
95          FIN(is_not_sm(<args>));
96           if(prg_string_order_case_insensitive ("Office Network.NAN-Rural.WH1",name)==0 ||
97                   prg_string_order_case_insensitive ("Office Network.NAN-Rural.WH2",name)==0 ||
98                   prg_string_order_case_insensitive ("Office Network.NAN-Rural.WH3",name)==0 ||
99                   prg_string_order_case_insensitive ("Office Network.NAN-Rural.Collector",name)==0)
100             FRET(OPC_TRUE);
101         FRET(OPC_FALSE);
102 }
103
104 static void
105 print_all_sms_info(void)
106         {
107         int i;
108         FIN(print_all_sms_info(<args>));
109         printf("Number of Nodes = %d\n", rreq_index+1);
110         for(i=0;i<rreq_index;i++)
111                 {
112                 printf("_____ %d _____\n", i+1);
113                 printf("Src_addr = ");
114                 print_ipv4(sm_rreq.src_addr.address.ipv4_addr);
115                 printf("Prev_addr = ");
116                 print_ipv4(sm_rreq.prev_addr.address.ipv4_addr);
117                 printf("Smart Meter Name:-->  %s\n", sm_rr[i].src_name);
118                 printf("Prev addr Name: %s\n", sm_rr[i].prev_name);
119                 printf("(X,Y) = (%f , %f)\n", sm_rr[i].x_pos,sm_rr[i].y_pos);
120                 printf("Received hop Count = %d\n", sm_rr[i].hop_count);
```

```
121                 printf("Estimated hop count = %d\n", sm_rr[i].est_hop_count);
122                 if (sm_rr[i].is_under_attack)
123                         printf("==================================> WORMHOLE ATTACK\n");
124                 else
125                         printf("No attack\n");
126                 }
127     FOUT;
128 }
129
130 ////////////////////////////////////////////////////////////////////////////
131 static void print_sm_info(SM_RR sm_rreq)
132         {
133         FIN(print_sm_info(<args>));
134         printf("_____\n");
135         printf("Smart Meter Name:-->  %s\n", sm_rreq.src_name);
136         printf("Prev addr Name: %s\n", sm_rreq.prev_name);
137         printf("(X,Y) = (%f , %f)\n", sm_rreq.x_pos,sm_rreq.y_pos);
138         printf("Received hop Count = %d\n", sm_rreq.hop_count);
139         printf("Estimated hop count = %d\n", sm_rreq.est_hop_count);
140         if (sm_rreq.is_under_attack)
141                 printf("==================================> WORMHOLE ATTACK\n");
142         else
143                 printf("No attack\n");
144     FOUT;
145 }
```

1.4 Analytical Model

Listing 1.4: Analytical Model

```
1 static int
2 hopCount(FLOAT64 x_src, FLOAT64 y_src, FLOAT64 x_dest, FLOAT64 y_dest) {
3         FLOAT64 euc_dis;
4         M_INT hop_count;
```

```
5         ALGEB F_E, P_E, E, d, l,a, b; /* Maple data-structures */
6         //char err[2048]; /* command input and error string buffers */
7         //MKernelVector kv; /* Maple kernel handle */
8         MCallBackVectorDesc cb = { textCallBack, 0, /* errorCallBack not used */
9         0, /* statusCallBack not used */
10        0, /* readLineCallBack not used */
11        0, /* redirectCallBack not used */
12        0, /* streamCallBack not used */
13        0, /* queryInterrupt not used */
14        0 /* callBackCallBack not used */
15        };
16
17  FIN(hopCount(<args>));
18        hop_count = 0;
19  EvalMapleStatement(kv, "restart:");
20        EvalMapleStatement(kv, "with(Student[Calculus1]):");
21        EvalMapleStatement(kv, "Digits:= 30:");
22
23        l = EvalMapleStatement(kv, "evalf(sqrt((x1-x2)^2+(y1-y2)^2)/1000):");
24        MapleAssign(kv, ToMapleName(kv, "x1", TRUE), ToMapleFloat(kv, x_src));
25        MapleAssign(kv, ToMapleName(kv, "y1", TRUE), ToMapleFloat(kv, y_src));
26        MapleAssign(kv, ToMapleName(kv, "x2", TRUE), ToMapleFloat(kv, x_dest));
27        MapleAssign(kv, ToMapleName(kv, "y2", TRUE), ToMapleFloat(kv, y_dest));
28        EvalMapleStatement(kv,"x1:=floor(x1):");
29        EvalMapleStatement(kv,"y1:=floor(y1):");
30        EvalMapleStatement(kv,"x2:=floor(x2):");
31        EvalMapleStatement(kv,"y2:=floor(y2):");
32        l = MapleEval(kv, l);
33        euc_dis = MapleToFloat64(kv, l);
34        //MaplePrintf(kv, "Init_Euc_Dis = %f", euc_dis);
35
36        while (euc_dis > range) {
37              F_E = EvalMapleStatement(kv, "(2/(Pi*r^2)) * e * arccos((e^2+d^2-r^2)/(2*e*d)):");
38              MapleAssign(kv, ToMapleName(kv, "r", TRUE), ToMapleFloat(kv, range));
39              MapleAssign(kv, ToMapleName(kv, "d", TRUE), ToMapleFloat(kv, euc_dis));
```

```
40                  // MapleALGEB_Printf(kv,"\nF_E = %a\n", F_E);
41
42                  a = ToMapleName(kv, "a", TRUE);
43                  MapleAssign(kv, a, F_E);
44                  P_E = EvalMapleStatement(kv, "evalf(int(a,e=d-r..e)) assuming e=d:");
45                  // MapleALGEB_Printf(kv,"\nP_E = %a\n", P_E);
46
47                  b = ToMapleName(kv, "b", TRUE);
48                  MapleAssign(kv, b, P_E);
49                  E = EvalMapleStatement(kv, "d-r +
50            evalf(ApproximateInt((1-b)^(N*Pi*r^2),e=d-r..d+r, method=simpson)):");
51                  MapleAssign(kv, ToMapleName(kv, "N", TRUE), ToMapleFloat(kv, density));
52
53                  E = MapleEval(kv, E);
54                  // MapleALGEB_Printf(kv,"\nE = %a\n", E);
55
56                  d = MapleSelectRealPart(kv, E);
57                  euc_dis = MapleToFloat64(kv, d);
58                  //MapleALGEB_Printf(kv, "\nNew Euc_dis = %a\n", d);
59
60                  EvalMapleStatement(kv, "F_E:='F_E':");
61                  EvalMapleStatement(kv, "P_E:='P_E':");
62                  EvalMapleStatement(kv, "a:='a':");
63                  EvalMapleStatement(kv, "b:='b':");
64                  EvalMapleStatement(kv, "E:='E':");
65                  EvalMapleStatement(kv, "d:='d':");
66
67                  ++hop_count;
68          }
69          ++hop_count;
70          FRET (hop_count);
71  }
72
73  /* callback used for directing result output */
74  static void M_DECL textCallBack(void *data, int tag, char *output) {
```

```
75          printf("%s\n", output);
76  }
```

1.5 Packet Handling Function

Listing 1.5: Packet Handling Function

```
1   static void aodv_rte_rreq_pkt_arrival_handle
2   (Packet* ip_pkptr, Packet* aodv_pkptr, IpT_Dgram_Fields* ip_dgram_fd_ptr,
3                                                   IpT_Rte_Ind_Ici_Fields* intf_ici_fdstruct_ptr,
4                                                   AodvT_Packet_Option* tlv_options_ptr)
5           {
6           AodvT_Rreq*    rreq_option_ptr;
7           InetT_Address   prev_hop_addr;
8           AodvT_Route_Entry*  route_entry_ptr = OPC_NIL;
9           AodvT_Route_Entry*  rev_path_entry_ptr = OPC_NIL;
10          IpT_Interface_Info*  iface_elem_ptr;
11          IpT_Port_Info   in_port_info;
12          double     min_lifetime, lifetime;
13          int      dest_seq_num, new_ttl_value;
14          Packet*     ip_rreq_pkptr;
15          char     src_node_name [OMSC_HNAME_MAX_LEN];
16          char     src_hop_addr_str [INETC_ADDR_STR_LEN];
17          char     dest_node_name [OMSC_HNAME_MAX_LEN];
18          char     dest_hop_addr_str [INETC_ADDR_STR_LEN];
19          char     temp_str [2048];
20          double     existing_lifetime;
21          AodvC_Route_Entry_State route_entry_state;
22          Boolean     dest_seq_flag = OPC_FALSE;
23          InetT_Subnet_Mask  selected_subnet_mask;
24          Ici*     ip_iciptr;
25          int      mcast_major_port = IPC_MCAST_ALL_MAJOR_PORTS;
26          Manet_Rte_Ext_Cache_entry* ext_cache_entry_ptr = OPC_NIL;
27          SM_RR     new_rreq;
```

```
28
29          /** Handles the arrival of a route request message **/
30          FIN (aodv_rte_rreq_pkt_arrival_handle (<args>));
31
32          /* A route request packet has arrived at this node */
33          /* Get the request options from the packet   */
34          rreq_option_ptr = (AodvT_Rreq*) tlv_options_ptr->value_ptr;
35
36
37          /* Get the previous hop from which this packet arrived */
38          prev_hop_addr = ip_dgram_fd_ptr->src_addr;
39
40          /* The source address needs to be set at the first hop */
41          /* for the route request as when sending out the RREQ */
42          /* from the actual source node, the output interface */
43          /* address is not known as it is broadcast.    */
44
45
46          /* added by Nasim */
47          if (WH_IDS_IS_ENABALED)
48                  {
49              new_rreq.hop_count = (rreq_option_ptr->hop_count + 1);
50              new_rreq.rreq_id = rreq_option_ptr->rreq_id;
51              new_rreq.src_addr.address.ipv4_addr = rreq_option_ptr->src_addr.address.ipv4_addr;
52              new_rreq.dest_addr.address.ipv4_addr = rreq_option_ptr->dest_addr.address.ipv4_addr;
53              new_rreq.prev_addr.address.ipv4_addr = prev_hop_addr.address.ipv4_addr;
54              new_rreq.x_pos = rreq_option_ptr->x_pos;
55              new_rreq.y_pos = rreq_option_ptr->y_pos;
56                  inet_address_to_hname (rreq_option_ptr->src_addr, new_rreq.src_name);
57                  inet_address_to_hname (prev_hop_addr, new_rreq.prev_name);
58
59          if(!rreq_exist(new_rreq))
60                      check_wormhole_attack(new_rreq);
61
62                  }
```

```
63
64 .
65 .
66 .
```

References

[1] Wormhole attack. Website, Access date: 30 Mar., 2013. `http:www.wings.cs.sunysb.edu`.

[2] IEEE 802.11s Task Group. Draft ammendment to standard for information technology telecommuication and information exchange between systems-LAN/MAN specific requriements-Part II: Wireless medium acess control (MAC) and physical layer (PHY) specificiations: Amendment: ESS mesh networking, IEEE P902.11s/D1.06, Jul. 2007.

[3] F. Aalamifar. Viability of powerline communication for smart grid realization. Master's thesis, Queen's University, Apr. 2012.

[4] A. Aimajali, A. Viswanathan, and C. Neuman. Analyzing resiliency of the smart grid communication architectures under cyber attack. In *5th workshop on cyber security experimentaion and test*, Jul. 2012.

[5] I. Akyildiz and X. Wang. *Wireless Mesh Networks*. Advanced Texts in Communications and Networking. Wiley, 2009.

[6] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Comput. Netw.*, 47(4):445–487, Mar. 2005.

[7] Wi-Fi Alliance. Wi-Fi for the smart grid mature, interoperable, secure technology for advanced smart energy management communications, Sep. 2010.

[8] V. Aravinthan, V. Namboodiri, S. Sunku, and W. Jewell. Wireless AMI application and security for controlled home area networks. In *IEEE Power and Energy Society General Meeting*, pages 1–8, Jul. 2011.

[9] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens. Mitigating byzantine attacks in Ad Hoc wireless networks. Technical report, Department of Computer Science, Johns Hopkins University, Tech, 2004.

[10] H. Tai B. Davito and R. Uhlaner. Mckinseyon smart grid: The smart grid and the promise of demand-side management, 2010.

[11] A. Bartoli, J. Hernandez-Soriano, M. Dohler, A. Kountouris, and D. Barthel. Secure lossless aggregation for smart grid M2M networks. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 333–338, Oct. 2010.

[12] N. Beigi-Mohammadi, J. Mišić, V. B. Mišić, and H. Khazaei. A framework for intrusion detection system in advanced metering infrastructure. *Wiley Journal of Security and Communication Networks*, 1939-0122 2012.

[13] N. Ben Salem and J. P. Hubaux. Securing wireless mesh networks. *IEEE Wireless Communications*, 13(2):50–55, Apr. 2006.

[14] C. Bennett and S. B. Wicker. Decreased time delay and security enhancement recommendations for AMI smart meter networks. In *Innovative Smart Grid Technologies (ISGT)*, pages 1–6, Jan. 2010.

[15] R. Berthier and W. H. Sanders. Specification-based intrusion detection for advanced metering infrastructures. In *IEEE 17th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 184–193, Dec. 2011.

[16] R. Berthier, W. H. Sanders, and H. Khurana. Intrusion detection for advanced metering infrastructures: Requirements and architectural directions. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 350–355, Oct. 2010.

[17] M. Bishop. *Introduction to computer security*. Addison-Wesley, 2004.

[18] J. Blum, A. Neiswender, and A. Eskandarian. Denial of service attacks on inter-vehicle communication networks. In *International IEEE Conference on Intelligent Transportation Systems, ITSC*, pages 797–802, Oct. 2008.

[19] J. D. Camp and E. W. Knightly. The IEEE 802.11s extended service set mesh networking standard, 2012.

[20] H. Sun Chiu and K. Lui. Delphi: wormhole detection mechanism for ad hoc wireless networks. In *1st International Symposium on Wireless Pervasive Computing,*, Jan. 2006.

[21] F. M. Cleveland. Cyber security issues for advanced metering infrasttructure (AMI). In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–5, Jul. 2008.

[22] Federal Communications Commission. National broadband plan. Website, Mar. 2013. `http://www.broadband.gov/plan/12-energy-and-the-environment/`.

[23] Smart Grid communications networks from Trilliant. Secure mesh NAN, industrys most advanced network architecture. `http://www.trilliantinc.com/products/securemesh-nan`, Access date: Feb., 2013.

[24] S. Dawson-Haggerty, A. Tavakoli, and D. Culler. Hydro: A hybrid routing protocol for low-power and lossy networks. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 268–273, Oct. 2010.

[25] A. Egners. Evaluating IEEE 802.11s against security requirements of wireless mesh networks, 2010.

[26] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki. Toward intelligent machine-to-machine communications in smart grid. *IEEE Communications Magazine*, 49(4):60–65, Apr. 2011.

[27] A. Geriks. A survey of wireless mesh networking security technology and threats, 2007.

[28] NIST Smart Grid Interoperability Panel Cyber Security Working Group. Introduction to NISTIR 7628 guidelines for smart grid cyber security, Sep. 2010.

[29] D. Harkins. Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks. In *Second International Conference on Sensor Technologies and Applications, SENSORCOMM '08.*, pages 839–844, Aug. 2008.

[30] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN mesh standard. *Wireless Communications, IEEE*, 17(1):104–111, Feb. 2010.

[31] L. Hu and D. Evans. Using Directional Antennas to Prevent Wormhole Attacks. In *Network and Distributed System Security Symposium Conference Proceedings*, San Diego, Feb. 2004. Director of Conferences and Education, Internet Society.

[32] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1976–1986, Apr. 2003.

[33] Maplesoft Inc. Maple 16. Website, Mar. 2013. `http://www.maplesoft.com`.

[34] T. Iwao, K. Yamada, M. Yura, Y. Nakaya, A.A. Andrdenas, S. Lee, and R. Masuoka. Dynamic data forwarding in wireless mesh networks. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 385–390, Oct. 2010.

[35] G. Iyer, P. Agrawal, E. Monnerie, and R. S. Cardozo. Performance analysis of wireless mesh routing protocols for smart utility networks. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 114–119, Oct. 2011.

[36] H. Jerome. 802.11s mesh networking, 2011.

[37] P. Jokar, H. Nicanfar, and V. Leung. Specification-based intrusion detection for home area networks in smart grids. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 208–213, Oct. 2011.

[38] J. Jung, K. Lim, J. Kim, Y. Ko, Y. Kim, and S. Lee. Improving IEEE 802.11s wireless mesh networks for reliable routing in the smart grid infrastructure. In *IEEE International Conference on Communications Workshops (ICC)*, pages 1–5, Jun. 2011.

[39] I. Khalil, S. Bagchi, and N. B. Shroff. Mobiworp: Mitigation of the wormhole attack in mobile multihop wireless networks. In *Securecomm and Workshops*, pages 1–12, Sept. 2006.

[40] D. Kuhlman, R. Moriarty, T. Braskich, S. Emeott, and M. Tripunitara. A correctness proof of a mesh security architecture. In *21st Computer Security Foundations Symposium,CSF '08. IEEE*, pages 315–330, Jun. 2008.

[41] P. Kulkarni, S. Gormus, Zhong Fan, and B. Motz. A self-organising mesh networking solution based on enhanced RPL for smart metering communications. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, Jun. 2011.

[42] N. Kush, E. Foo, E. Ahmed, I. Ahmed, and A. Clark. Gap analysis of intrusion detection in smart grids. In Craig Valli, editor, *2nd International Cyber Resilience Conference*, pages 38–46. Secau-Security Research Centre, Aug. 2011.

[43] National Energy Technology Laboratory. Smart grid principal characteristic: Operates resiliently against attack and natural disasters. Website, 2009.

[44] Y.W. Law, P. Hartel, J. den Hartog, and P. Havinga. Link-layer jamming attacks on S-MAC. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, pages 217–225, Feb. 2005.

[45] M. LeMay and C. Gunter. Cumulative attestation kernels for embedded systems. In *Computer Security, ESORICS*, pages 655–670. Springer Berlin, Heidelberg, 2009.

[46] B. Li, Y. Qin, C. P. Low, and C. L. Gwee. A survey on mobile WiMAX [wireless broadband access]. *IEEE Communications Magazine*, 45(12):70–75, 2007.

[47] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang. Smart transmission grid: Vision and framework. *IEEE Transactions on Smart Grid*, 1(2):168–177, Sep. 2010.

[48] Z. Lu, X. Lu, W. Wang, and C. Wang. Review and evaluation of security threats on the communication networks in the smart grid. In *Military communication conference, MILCOM*, pages 1830–1835, Nov. 2010.

[49] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75–77, Jun. 2009.

[50] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 107–116, 2010.

[51] A. R. Metke and R. L. Ekl. Security technology for smart grid networks. *IEEE Transactions on Smart Grid*, 1(1):99–107, Jun. 2010.

[52] K. Akkaya N. Saputro and S. Uludag. A survey of routing protocols for smart grid communications. *Computer Networks*, 56(11):2742 – 2771, 2012.

[53] A. Naveed, S. S. Kanhere, and S. K. Jha. Attacks and security mechanisms, 2010.

[54] H. L. Nguyen and U. T. Nguyen. Study of different types of attacks on multicast in mobile ad hoc networks. In *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, page 149, Apr. 2006.

[55] NIST. Report to NIST on smart grid interoperability standards roadmap EPRI, Jun. 2009.

[56] OPNET Technologies, Inc. OPNET Modeler 17.1. Website, Mar. 2011. `http://www.opnet.com`.

[57] A. Patel, J. Aparicio, N. Tas, M. Loiacono, and J. Rosca. Assessing communications technology options for smart grid applications. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 126–131, Oct. 2011.

[58] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesely, 2001.

[59] F. Rahimi and A. Ipakchi. Demand response as a market resource under the smart grid paradigm. *IEEE Transactions on Smart Grid*, 1(1):82–88, Jun. 2010.

[60] T. Roosta, D. K. Nilsson, U. Lindqvist, and A. Valdes. An intrusion detection system for wireless process control systems. In *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS.*, pages 866–872, Oct. 2008.

[61] P. Swain S. Chakraborty and S. Nandi. Proportional fairness in MAC layer channel access of IEEE 802.11s EDCA based wireless mesh networks. *Ad Hoc Networks*, 11(1):570–584, 2013.

[62] D. Sanjeev and K. Sanjeev. Security challenges in multihop wireless mesh networksa survey. In Dasun Weerasinghe, editor, *Information Security and Digital Forensics*, volume 41 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 92–101. Springer Berlin Heidelberg, 2010.

[63] T. Sauter and M. Lobashov. End-to-end communication architecture for smart grids. *IEEE Transactions on Industrial Electronics*, 58(4):1218–1228, Apr. 2011.

[64] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (IDPS), recommendations of the national institute of standards and technology, Feb. 2007.

[65] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on TCP. In *IEEE Symposium on Security and Privacy, Proceedings.*, pages 208–223, May 1997.

[66] J. Seo and G. Lee. An effective wormhole attack defence method for a smart meter mesh network in an intelligent power grid. In *International Journal of Advanced Robotic Systems*, volume 9, 2012.

[67] S. Seth and A. Gankotiya. Denial of service attacks and detection methods in wireless mesh networks. In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing (ITC)*, pages 238–240, Mar. 2010.

[68] F. Skopik and Z. Ma. Attack vectors to metering data in smart grids under security constraints. In *IEEE 36th Annual on Computer Software and Applications Conference Workshops (COMPSACW)*, pages 134–139, Jul. 2012.

[69] H. K. So, S. H. M. Kwok, E. Y. Lam, and K. Lui. Zero-configuration identity-based signcryption scheme for smart grid. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 321–326, Oct. 2010.

[70] N. Song, L. Qian, and X. LI. Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach. In *Parallel and Distributed Processing Symposium, Proceedings. 19th IEEE International*, Apr. 2005.

[71] W. Stallings. *Cryptography And Network Security, 4/E*. Pearson Education, 2006.

[72] Internet Engineering task force (IETF) routing over low power and lossy networks (ROLL) working group. Routing over low power and lossy networks (RPL). Technical report, IEEE, Dec. 2010.

[73] K. Thambu, J. Li, N. Beigi-Mohammadi, Y. He, J. Mišić, and L. Guan. Toronto hydroelectric system Ltd-Ryerson center for urban energy: Secure and reliable data communication for smart grid, Dec. 2012.

[74] D. Wang, Z. Tao, J. Zhang, and A. A. Abouzeid. RPL based routing for advanced metering infrastructure in smart grid. In *IEEE International Conference on Communications Workshops (ICC)*, pages 1–6, May 2010.

[75] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *Proceedings of the 3rd ACM workshop on Wireless security*, WiSe '04, pages 51–60. ACM, 2004.

[76] W. Wang, Y. Xu, and M. Khanna. A survey on the communication architectures in smart grid. *Computer Networks*, 55(15):3604–3629, 2011.

[77] X. Wang and A. O. Lim. IEEE 802.11s wireless mesh networks: Framework and challenges. *Ad Hoc Network*, 6(6):970–984, Aug. 2008.

[78] X. Wang and J. Wong. An end-to-end detection of Wormhole attack in wireless Ad-hoc networks. In *31st Annual International Computer Software and Applications Conference, COMPSAC*, volume 1, pages 39–48, Jul. 2007.

[79] X. Wang, J.S. Wong, F. Stanley, and S. Basu. Cross-layer based anomaly detection in wireless mesh networks. In *Ninth Annual International Symposium on Applications and the Internet, SAINT '09.*, pages 9–15, Jul. 2009.

[80] X. Wang and P. Yi. Security framework for wireless communications in smart distribution grid. *IEEE Transactions on Smart Grid*, 2(4):809–818, Dec. 2011.

[81] H. Wu, C. Wang, and N. Tzeng. Novel self-configurable positioning technique for multihop wireless networks. *IEEE/ACM Transactions on Networking,*, 13(3):609–621, Jun. 2005.

[82] A. Yaar, A. Perrig, and D. Song. Pi: a path identification mechanism to defend against DDoS attacks. In *Symposium on Security and Privacy, Proceedings.*, pages 93–107, May 2003.

[83] Y. Yan, Y. Qian, and H. Sharif. A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 909–914, Mar. 2011.

[84] P. Yuan, V. Gambiroza, and E. Knightly. The IEEE 802.17 media access protocol for high-speed metropolitan-area resilient packet rings. *IEEE Network*, 18(3):8–15, 2004.

[85] F. Zou, N. Liu, P. Yi, and Y. Wu. A Survey on Security in Wireless Mesh Networks. *IETE Technical Review*, 27(1):6–14, 2010.