2014

RAPID: A Traffic-Agnostic Intrusion Detection for Resource-Constrained Wireless Mesh Networks

Hassanzadeh, Amin; Stoleru, Radu; Polychronakis, Michalis; Xie, Geoffrey

http://hdl.handle.net/10945/46774

RAPID: A Traffic-Agnostic Intrusion Detection for Resource-Constrained Wireless Mesh Networks

Amin Hassanzadeh, Radu Stoleru, Michalis Polychronakis[†], Geoffrey Xie[‡]
Department of Computer Science and Engineering, Texas A&M University, USA
[†]Computer Science Department, Columbia University, USA
[‡]Department of Computer Science, Naval Postgraduate School, USA
{hassanzadeh,stoleru}@cse.tamu.edu, mikepo@cs.columbia.edu, xie@nps.edu

ABSTRACT

Due to the recent increased interest in wireless mesh networks (WMN), their security challenges have become of paramount importance. An important security mechanism for WMN, intrusion detection, has received considerable attention from the research community. Recent results show that traditional monitoring mechanisms are not applicable to real-world WMN due to their constrained resources (memory and processing power), which result in high false negative rates since only few IDS functions can be activated on monitoring nodes. Cooperative solutions, on the other hand, have high communication overhead and detection delay when traffic is high. A practical traffic-aware IDS solution was recently proposed for resource-constrained WMN, however, traffic-awareness might not be feasible for some WMN applications. This paper proposes a traffic-agnostic IDS solution that uses a link-coverage approach to monitor both local and backbone WMN traffic. Using real-world experiments and extensive simulations we show our proposed IDS solutions outperform traffic-aware IDS solutions while requiring lower computation and communication overhead.

1. INTRODUCTION

Wireless Mesh Networks (WMN) have emerged as a self-managing and cost-effective broadband wireless networking technology to provide Internet, Intranet, and other networking services in large remote areas without networking infrastructures. WMN can also serve as a backbone communication infrastructure among mobile or fixed clients, and hosts (e.g., local file servers). Recently, the number of WMN deployments has been increasing since they are suitable for many applications such as disaster response [1–3], environmental monitoring [4], rural IT services [5,6] and many others [7].

As the interest in WMN increases, security issues, e.g., intrusion detection, become of paramount importance. Adopting traditional intrusion detection mechanisms from wired networks is not practical because: a) WMN lack single vantage points (e.g., gateways in wired networks) where network traffic can be inspected; b) WMN hardware has *limited resources* (e.g., CPU and RAM) to run resource-demanding intrusion detection systems (IDS). In light of these issues, researchers have proposed *distributed* and *resource-aware* solutions for network-wide intrusion detection in WMN. The state-of-the-art distributed solutions for resource constrained WMN can be categorized as: 1) *monitoring node* solutions; 2) *cooperative IDS* solutions; and 3) *traffic-aware IDS* solutions.

Monitoring node solutions [8–10] select a subset of nodes (called monitoring nodes), assign each selected node the same set of IDS functions for monitoring a distinct part of network (i.e., either communication links [10] or WMN nodes [9]). These solutions, however, suffer from high false negative rates because some IDS functions cannot be activated on monitoring nodes due to limited resources (e.g., memory and processing power). Recently, OpenLIDS [11], proposed a *Lightweight* detection engine for WMN that imposes less computational load than off-the-shelf IDS. However, when compared to off-the-shelf IDS, OpenLIDS has even higher false negative rates because fewer IDS functions are implemented in the detection engine.

In *Cooperative IDS* solutions [12,13], resource-constrained nodes are assigned a few distinct IDS functions for local intrusion detection and exchange information for cooperative intrusion detection (i.e., to achieve higher detection rates). Cooperative IDS thus have low false negative rates, however, they incur high detection delay and high communication overhead due to the message exchange required for intrusion detection. Hence, considering the relatively high traffic rates in WMN, these solutions are not practical and scalable for WMN.

Traffic-aware IDS was recently proposed [14] as a practical intrusion detection mechanism for resource-constrained WMN. These solutions use the knowledge that a security administrator has about network traffic to distribute IDS functions only along routing paths. Considering the distinct set of IDS functions on each node along a routing path, the entire traffic on that path is investigated by more IDS functions while none of the nodes is overloaded. This mechanism, unlike cooperative IDS, is non-cooperative, providing a real-time detection mechanism, and has higher detection rates than monitoring solutions [14]. However, traffic-awareness is sometimes a strong assumption for many WMN applications where routing paths change frequently.

The research presented in this article is motivated by the

Technical Report 2014, Texas A& M University

fact that in many WMN applications traffic paths change very often, which consequently degrades the performance of traffic-aware IDS solutions. For example, routing paths in large scale WMN that provide networking services for mobile clients are subject to change due to client mobility. Additionally, WMN topology, especially in outdoor deployments, may change due to node failures or drastic link-quality changes. Hence, the traffic knowledge has to be very accurate and up-to-date in traffic-aware solutions, which is not always feasible. In this article, we propose a traffic-agnostic intrusion detection mechanism for resource-constrained WMN that monitors all communication links, instead of only few paths. Such an approach in WMN IDS is traffic-independent, but requires more mesh nodes to participate in detection mechanism. Thus, traffic-agnostic solutions are more complex than traffic-aware solutions since all of WMN nodes must be considered in the optimal IDS function distribution problem (as opposed to traffic-aware solutions that only consider few nodes on the routing paths).

Our proposed IDS mechanism is based on traffic-agnostic and link-coverage approaches inspired by the PRIDE [14] and EEMON [10] intrusion detection systems. Our solution, irrespective to the changes in WMN traffic paths, is able to monitor the entire WMN traffic, at the price, however, of putting IDS load on all WMN nodes instead of those located only along routing paths. In our proposed solution, each node, depending on its available resources, is assigned a subset of IDS functions, i.e., a customized IDS configuration, and investigates the entire network traffic on the set of communication links it can monitor (i.e., in its coverage area). This customized IDS allows resource conservation on resource-constrained WMN nodes and also increases the probability of monitoring a WMN link with multiple distinct IDS functions activated on all WMN nodes that can monitor the link. The decision of activating the optimal subset of IDS functions on each node, to achieve the maximum possible link coverage and consequently maximum detection rates, is shown [14, 15] to be an optimization problem. It is worth mentioning that for a given network size, the complexity of traffic-agnostic solution is larger than traffic-aware solution as it needs to find optimal IDS function distribution for all nodes. Hence, our proposed solution has to be fast and scalable. More precisely, the contributions of this article are as follows:

- It demonstrates that distributing IDS functions among WMN nodes increases the intrusion detection rate when compared to state-of-the-art monitoring mechanisms.
- It proposes a traffic-agnostic IDS solution for resource-constrained WMN based on a link-coverage mechanism that monitors all WMN links instead of only routing paths.
- It formulates a novel IDS Function distribution problem, called Link Coverage Problem (LCP), with the objective to maximize the intrusion detection rate while ensuring that nodes are not overloaded by IDS functions.
- It proposes RAPID (Randomized APproach Intrusion Detection), a protocol for solving LCP, and two centralized and distributed implementations of it. It also provides an analysis of the two implementations to illustrate the tradeoff of time and communication overhead for intrusion detection rate.
- It presents the performance of RAPID for intrusion detection rates and compare it with state-of-the-art solutions.

This article is organized as follows. Section 2 presents state-of-the-art solutions for intrusion detection in resource-constrained WMN and enumerates some scenarios that are not considered in those solutions. Section 3 presents the system and security models considered in this article. Preliminaries and problem formulation are presented in Section 4. Section 5 presents the RAPID protocol and two implementations for it. We present the performance of our proposed IDS solution in Section 6 and conclude the article in Section 7.

2. BACKGROUND AND MOTIVATION

In this section, we first review state-of-the-art IDS solutions proposed for WMN and identify challenges they face, making them impractical for WMN. Next, we present our inspiring IDS solution, PRIDE [14], and describe a motivating scenario that highlights PRIDE's limitations in highly dynamic WMN. Finally, we present lessons learned from state-of-the-art solutions that helped the design of our proposed IDS.

2.1 State of the Art

The problem of intrusion detection in wireless mesh networks has received some attention from the research community. Some existing solutions address specific attacks (e.g., Man-in-the-Middle and Wormhole Attacks [16], Grayhole attack [17], message fabrication attack [13], and scheduling in WMN [18]). Other solutions are general IDS solutions for mesh networks, which consider memory, processing [11, 14, 19], and energy [10] constraints for performing off-the-shelf IDS tools on WMN devices. The later group of IDS solutions aim at addressing the challenges associated with monitoring mechanisms for intrusion detection in WMN, as opposed to the former group which addresses specific attacks in such networks. In this article, we do not propose a detection rule/mechanism for a specific attack. Instead, we investigate the problem of monitoring WMN traffic using off-the-shelf IDS that can detect both known and stealth attacks [20, 21].

Due to the infrastructureless nature of WMN, researchers, inspired by early research in ad hoc networks [22–25], have proposed *distributed* solutions for intrusion detection systems in WMN [9, 26–28]. In early research, mainly in the context of sensor networks and MANET, an intrusion detection agent was placed on each ad hoc node (completely decentralized) [22]. This approach is inefficient primarily because of redundant monitoring on multi-hop traffic that causes unnecessary resource consumption (e.g., resources that can be allocated to other networking services). The aforementioned inefficiency triggered major research on *optimal monitoring* for intrusion detection systems [8–10, 23–25]. In optimal monitoring solutions, a minimum subset of nodes is selected to perform IDS functions and monitor the entire network traffic. However, since the number of IDS functions on each node is limited by the amount of resources available to it, these solutions can only detect a limited number of attacks. In order to address the challenges posed by optimal monitoring solutions, i.e., high false negative rates due to performing few IDS functions, some researchers have proposed *lightweight IDS* for resource-constrained WMN devices [11, 13, 29]. These solutions, however, still employ attack-specific detection engines because they implement few IDS functions, resulting in high false negative detection rates.

As an alternative, to achieve high detection rates while conserving resources on resource-constrained WMN nodes, cooperative IDS solutions [12,13] have been proposed. In cooperative IDS solutions (e.g., hierarchical [12,30–32], group-based [33–35], zone-based [36], or neighbor-assisted [37,38]) every node is assigned a few IDS functions to detect attacks based on local observation. A cooperative IDS engine is then employed for detecting more attacks, based on neighbor information [13,38]. Cooperative mechanisms incur high communication overhead, caused by message exchange required for intrusion detection, and high detection latency [39], since some of decisions are made only after receiving other nodes' reports. Therefore, although cooperative IDS have proven viable for low-traffic networks, e.g., sensor networks, they are not practical (i.e., degrades the network performance and delays intrusion response) in WMN with significant traffic [6, 7, 27, 40, 41].

Recently, two traffic-aware IDS solutions, TRAM [26] and PRIDE [14,27], have been proposed for traffic monitoring of WMN routing paths. TRAM shows how to use mesh nodes in a multi-channel WMN to monitor network traffic while also contributing in WMN routing process. The proposed solution, however, does not specify the IDS tools and detection engine used in intrusion monitoring and the actual computational load imposed to WMN nodes. PRIDE proposes to use the security administrator's knowledge about WMN traffic to distribute IDS functions (i.e., Snort detection rules) to the nodes along WMN routing paths. Each node in a traffic path is assigned a distinct set of IDS functions such that the network traffic on that path is investigated by maximum, if not all, possible IDS functions in real-time (no detection latency caused by node cooperation). It is shown [14,27] that such a traffic-aware solution has a higher detection rate than optimal monitoring solutions when applied to real-world resource-constrained WMN.

2.2 Motivation

This research is motivated by the fact that, although PRIDE was shown [14,27] to be a practical intrusion detection mechanism for resource constrained WMN, it is based on a strong assumption about WMN traffic. In this section, we investigate some of the *challenges* that PRIDE faces in some real-world WMN applications. Additionally, we show some *features* that, if added to PRIDE, can significantly improve its performance.

2.2.1 Challenges

PRIDE considers static resource-constrained WMN where network topology does not change often (compared to other ad hoc networks). It assumes that network information periodically collected by the base station reflects the most recent network topology. However, research has shown [3, 42, 43] that even static WMN topology and routing paths are subject to change due to: a) link-quality variations caused by weather, noise and other radio signals, etc.; b) mobility of clients and their requested services that result in changes of WMN routing paths; c) node failure (e.g., running out of power) or node replacement (e.g., administrative reasons) during network lifetime. Hence, traffic awareness might be a strong assumption for many WMN applications. Motivated by this fact, we propose a traffic-agnostic IDS solution.

PRIDE is not a *scalable* solution because its execution time (i.e., to find optimal IDS function distribution for WMN nodes) significantly increases when network size, number of paths, and number of IDS functions increase, or when the memory threshold on the nodes decreases. The results shown in [14] are for a 10-node WMN for only 2 paths (for each given path length). When applied to a larger network (e.g., 30 nodes and 15 paths), however, it takes more than an hour to obtain the optimal IDS function distribution. Thus, a practical IDS solution must be able to quickly produce optimal results when used for large scale WMN. We note here that the traffic-agnostic solution, proposed here, has to solve a more complex problem because all WMN nodes perform IDS operations. Therefore, we need to develop an algorithm that can produce optimal IDS function distribution for a large WMN in a short period of time.

2.2.2 Improving Features

PRIDE only considers *multi-hop* attacks which means the attack traffic (malicious packets(s)) is routed across multiple nodes (i.e., at least one WMN backbone link). In addition, the experimental results [14] show that the longer the path is, the higher the detection rate will be. We aim to design an IDS that can detect both single-hop attacks (i.e., both attacker and target are clients connected to same router) and multi-hop attacks, routed through short paths (e.g., 2 hops).

PRIDE proposes a centralized algorithm that requires periodic data collection from WMN nodes and a computationally powerful base station to produce the optimal IDS function distribution. In this research, we propose an IDS solution that can be implemented in a distributed manner where WMN routers independently choose the optimal set of IDS function to perform. The distributed approach is based on random IDS function selection by the nodes that incur no communication overhead (caused by data exchanges between nodes and the base station). It also no longer requires a computationally powerful base station. We show that random IDS function selection surprisingly achieves near optimal network coverage ratios especially for high density WMN.

PRIDE uses a node-coverage approach, which means that only nodes along each routing path participate in traffic monitoring. However, it is shown [10] that link-coverage can achieve a higher link/path coverage ratio in infrastructure-less wireless networks. Hence, in addition to the nodes located on each routing path, other nodes can also participate in traffic monitoring if they can monitor at least one link of that path. We use a link-coverage approach in our proposed IDS and show how it increases the link/path coverage ratio.

3. SYSTEM AND SECURITY MODELS

System Model: The system we consider in this research is similar to the one considered in PRIDE, as specified by the IEEE 802.11s WLAN Mesh Standard [44]. The system consists of: i) *mesh access points (MAP)* that connect WMN clients (or "clients" for short) to the mesh network and external hosts (i.e., Internet); ii) a *wireless mesh backbone* consisting of relay nodes, also known as *mesh points (MP)*; and

iii) *gateways*, connecting the mesh network (internal hosts) to the Internet (external hosts). Some WMN routers/nodes (Note: from here on we use "node" and "router" interchangeably) are configured to work as both MP and MAP. Each WMN node has two wireless interfaces providing mesh connections on one interface and network access to the clients on the other interface.

The WMN traffic is either *external*, i.e., between clients and external hosts, or *internal*, i.e., between two internal hosts. We note here that a host inside the mesh is either a client or a local server (e.g., a local FTP server) connected to the mesh routers. Our system also requires the presence of a base station (for centralized approach) — a computationally powerful node which periodically and securely collects, via a middleware, information about mesh nodes: *processing/memory* loads, traffic information, etc. Based on the collected information, the base station finds optimal IDS functions to be assigned to each node and then securely broadcasts [45] them to the nodes.

Attacker Model: We consider several types of attacks in this article. An *Insider* attack is launched by either a malicious client connected to a MAP or by a compromised router. An *Outsider* attack is launched by either an external host (connected to WMN through gateways) or by an unauthorized client not connected to WMN. For example, a malicious external host communicating with a mesh client or a local server in our WMN can launch an outsider attack. Furthermore, a malicious unauthorized wireless node physically located in the WMN coverage area, but not associated to a WMN MAP, is also considered as an outsider attacker.

Depending on the attack type, i.e., insider or outsider, the target can be either single-hop or multi-hop. For example, a malicious client connected to a MAP attacking the MAP or another host connected to that MAP is actually launching a single-hop attack. A compromised router attacking one of its neighbors (a WMN router) is also an example of a single-hop attack. The aforementioned single-hop attacks are insider attacks. An unauthorized node (physically located in WMN area) can launch a single-hop *node-based* attack (targeting a WMN router or host) or a single-hop *link-based* attack (targeting a communication link). A multi-hop attack, however, is always against routers or hosts. In an insider attack, a malicious client or a compromised router can launch attacks against multi-hop routers/hosts in the WMN. Moreover, outsider attacks performed by external hosts always target multi-hop routers/hosts in the WMN.

Intrusion Detection Engine: Similar to PRIDE, the IDS we consider in this article is Snort [21] because it is a mainstream off-the-shelf IDS and experimentally observed [11] to consume less resources than other IDS, e.g., Bro [20]. Moreover, unlike Bro, Snort is readily available for mesh hardware, as part of the OpenWrt development tree [46], i.e., a Linux distribution for embedded networking devices. Snort can be configured for different levels of intrusion detection. More complex actions performed by the detection engine (e.g., number of active rule sets) require more memory [14]. We use the "ac-bnfa-nq" search method as it is experimentally observed to consume minimum memory [14] among all low memory search methods in Snort.

For effective detection of both single-hop and multi-hop attacks, the intrusion detection system running on the mesh router should inspect network traffic at two different points: the local (i.e., MAP) and upstream (i.e., MP) network interfaces. For instance, a single-hop attack from a local client to another client in the same subnet (connected to the same MAP) will go through the local interface, while a response from a malicious external web site will go through both interfaces. It is absolutely infeasible for resource-constrained WMN hardware to run two Snort instances to monitor traffic on both interfaces. In addition, "interface bonding" proposed for multi-interface configuration is not applicable to mesh routers as it destroys routing configurations. Hence, as it will be presented in more detail in Section 6, in our proposed IDS, we develop a multi-interface Snort for OpenWrt platform that monitors traffic on multiple interfaces simultaneously. Our multi-interface Snort requires only ∼4% additional memory load when compared to the original Snort.

IDS Function vs. Detection Module: PRIDE proposes a *modularization mechanism* that groups small rule files and splits large rule files, resulting in a few sets of Snort detection rules of equal sizes (∼250 detection rules per each set) called *detection modules* [14]. As presented in [27], the entire set of Snort rule files (i.e., ∼70 files) and their corresponding detection rules are put in either 6 or 12 modules. This assignment trades off complexity of the IDS function distribution problem with accuracy in memory load estimation. In this article, we employ 6-module and 12-module configurations, as defined in PRIDE, and their corresponding memory loads. Thus, from here on, the terms "IDS function" and "detection module" mean the same thing and will be used interchangeably.

4. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we formulate the optimal distribution of IDS functions (detection modules) as an optimization problem. Although the problem formulation we present is based on Snort terminology (similarly to PRIDE), it can be generalized to other IDS, e.g., Bro or Di-Sec [47], with few minor changes based on their internals and functionality. For example, Di-Sec, a security framework proposed for resource-constrained sensor networks, consists of several detection and defense modules, similar to the Snort detection modules, and several sub-components that can be modeled as Snort preprocessors.

4.1 Preliminaries

Given a wireless mesh network, we denote the number of its nodes and number of its links by n and q, respectively. We model the wireless mesh network as a graph $G = \{V, E\}$, where V is the set of mesh nodes (routers) $\{v_1, v_2, \cdots, v_n\}$, and E is the set of backbone links $\{e_1, e_2, \cdots, e_q\}$. An example of such a graph, is shown in Figure 1 where $V = \{v_1, v_2, ..., v_{10}\}$ and $E = \{e_1, e_2, ..., e_{16}\}$. Figure 1, represents the network graph a real-world WMN deployed over the floor of a building. We denote by matrix $\mathbb{M}_{q\times n}$ the mapping between nodes and links, i.e., $m_{ij} = 1$ iff node v_j can monitor link e_i. Based on the *link-coverage* definition [10], v_j can monitor e_i if e_i is incident to v_j or v_j is connected to the two end points of e_i. The set of all links that can be monitored by node v_j is called *Covering Set* of node v_j represented by CS_j [10]. Accordingly, we denote by MS_i the set of all nodes that can monitor link e_i, i.e., *Monitoring Set* of link e_i. For the example shown in Figure 1, the matrix $\mathbb{M}$ is as follows:

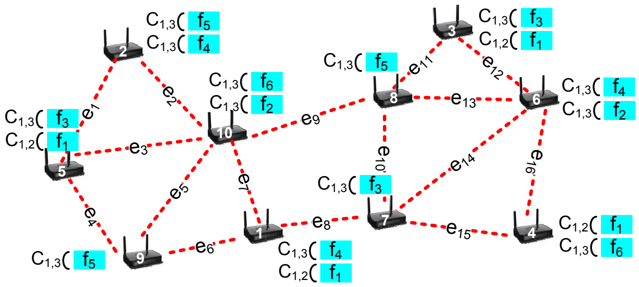


Figure 1: A WMN graph, consisting of 10 nodes and 16 links. As shown, a 6-module configuration is used in this WMN where Snort preprocessors are also grouped in three sets of preprocessors [27]. The nodes run different Snort configurations, e.g., node v_1 runs detection modules f_1 and f_4, which require preprocessors c_1, c_2 and c_3.

$$\mathbb{M}_{16\times 10} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \vdots & \vdots & & & & \cdots & & & & \vdots \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

We denote the set of all IDS functions (detection modules) by $\mathcal{F} = \{f_k \,|\, f_k \text{ is a set of detection rules}\}$ with size K (i.e., $|\mathcal{F}| = K$) where $K = 6$ in 6-module configuration and $K = 12$ in 12-module configuration. We also denote the set of IDS preprocessors (as in Snort) by $\mathcal{C} = \{c_r \,|\, \exists f_k \in \mathcal{F} \text{ that requires } c_r\}$ of size R (i.e., $|\mathcal{C}| = R$) where $R = 3$ in both 6-module and 12-module configurations. For the example presented in Figure 1, $\mathcal{F} = \{f_1, f_2, ..., f_6\}$, i.e., 6-Module configuration is used, and $\mathcal{C} = \{c_1, c_2, \text{and } c_3\}$. The dependency between IDS functions and preprocessors is stored in matrix $\mathbb{D}_{K\times R}$ where $d_{kr} = 1$ means that activation of module f_k requires the activation of preprocessor c_r. For the example shown in Figure 1, the matrix $\mathbb{D}^T$ is as follows:

$$\mathbb{D}^T_{3\times 6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Let $w : \{\mathcal{F}, \mathcal{C}\} \longrightarrow [0, 1]$ be a cost function that assigns memory load w^f_k and w^c_r to detection module f_k and preprocessor c_r, respectively. Consequently, vectors $W^f = [w^f_1, w^f_2, \cdots, w^f_K]$ and $W^c = [w^c_1, w^c_2, \cdots, w^c_R]$ represent memory loads for the detection modules in $\mathcal{F}$ and for the preprocessors in $\mathcal{C}$, respectively. Considering the 6-module configuration in PRIDE, for the configuration used in Figure 1, $W^f = [13.3\%, 14.6\%, 13\%, 17.4\%, 14.6\%, 17.3\%]$ and $W^c = [15.6\%, 1.1\%, 1\%]$. It is worth mentioning that $w^c_1 = 15.6\%$ is the total load caused by Snort base line, *stream5* (both static and dynamic loads as explained in PRIDE), and *frag3* - the most common and required Snort preprocessors for all detection modules [27]. We denote by $B = [b_1, b_2, ..., b_n]$ the base memory load (i.e., before performing IDS) of all nodes. Finally, the maximum allowable memory load (after detection modules and preprocessors are loaded) is represented by vector $\Lambda = [\lambda_1, \lambda_2, \cdots, \lambda_n]$, (*also called Memory Threshold*). Vector Λ depends on the memory space required by active services in WMN, and it is typically set by the security administrator.

4.2 Problem Formulation

The main objective of our proposed IDS is to monitor all WMN links using the maximum allowable number of detection modules that can be performed on WMN nodes (i.e., activated and executed by Snort on nodes). A higher number of detection modules executed by node v_j means more attack traffic can be detected on the links in CS_j. Thus, our IDS solution aims at assigning Snort detection modules on the WMN nodes, such that all of WMN links are monitored by the maximum number of modules and none of the nodes is overloaded. In order to mathematically formulate this problem, we first introduce several definitions.

DEFINITION 1. ***IDS Function Distribution****, represented by $T = \{(v_j, \mathcal{F}_j, \mathcal{C}_j) |\, v_j \in V,\ \mathcal{F}_j \subseteq \mathcal{F}, \text{and } \mathcal{C}_j \subseteq \mathcal{C}\}$, is a distribution of detection modules and preprocessors in the WMN, such that modules $\mathcal{F}_j$ and their corresponding preprocessors $\mathcal{C}_j$ are assigned to node v_j (i.e., they will be activated on the customized Snort executed on v_j).*

After the *IDS Function Distribution*, the set of detection modules and preprocessors assigned to WMN nodes are represented by binary matrices $\mathbb{X}_{n\times K}$ and $\mathbb{Z}_{n\times R}$, respectively. Accordingly, $x_{jk} = 1$ means module f_k is activated on node v_j and $z_{jr} = 1$ implies that preprocessor c_r is activated on node v_j (i.e., there is at least one module assigned to node v_j that requires preprocessor c_r). For example, the *IDS Function Distribution*, and matrices $\mathbb{X}$ and $\mathbb{Z}$ for the example given in Figure 1 are:

$$T = \{(v_1, \{f_1, f_4\}, \{c_1, c_2, c_3\}), (v_2, \{f_4, f_5\}, \{c_1, c_3\}), ...$$
$$..., (v_{10}, \{f_2, f_6\}, \{c_1, c_3\})\},$$

$$\mathbb{X}_{10\times 6} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & & \cdots & & \vdots \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{Z}_{10\times 3} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ & \vdots & \\ 1 & 0 & 1 \end{bmatrix}.$$

The total memory load of node v_j, after the *IDS Function Distribution*, becomes $L_j = b_j + \Sigma_{c_r \in \mathcal{C}_j} w^c_r + \Sigma_{f_k \in \mathcal{F}_j} w^f_k$. Obviously, an *IDS Function Distribution* in which there is at least one v_j such that $L_j > \lambda_j$ is deemed infeasible because the load L_j is not allowed to exceed the threshold λ_j.

DEFINITION 2. *For a given link e_i and its corresponding monitoring set MS_i,* ***Link Coverage Ratio (LCR)*** *is defined as $LCR_i = |U_i|/K$, where $U_i = \bigcup_{v_j \in MS_i} \mathcal{F}_j$ is the set of detection modules assigned to nodes that can monitor the link.*

DEFINITION 3. *Link e_i is called* ***Fully Covered*** *if $LCR_i = 1$ $(U_i = \mathcal{F})$, i.e., for $\forall f_k \in \mathcal{F}$, $\exists v_j \in MS_i$ assigned with $\mathcal{F}_j$ such that $f_k \in \mathcal{F}_j$.*

DEFINITION 4. ***Link Coverage Problem (LCP)*** *Given $G = \{V, E\}$, vectors W^f and W^c, and matrix $\mathbb{D}$, find a distribution $T = \{(v_j, \mathcal{F}_j, C_j) |\, v_j \in V \text{and } \mathcal{F}_j \subseteq \mathcal{F} \text{and } C_j \subseteq C\}$, such that $\frac{1}{q}\sum_{e_i \in E} LCR_i$ is maximized and $L_j \leq \lambda_j$, $\forall v_j \in V$.*

LCP aims at maximizing the average link coverage ratio while ensuring that memory loads on nodes are below their memory thresholds.

Given matrices $\mathbb{M}$ and $\mathbb{X}$, we denote by matrix $\mathbb{Y} = \mathbb{M} \cdot \mathbb{X}$ the mapping between links and the modules activated on the monitoring set of the links, i.e., y_{ik} is in the range $[0, n]$. For example, $y_{ik} = 0$ means that module k is not activated on any of nodes in MS_i while $y_{ik} > 0$ implies that there is at least one node in MS_i running module k. According to the LCR (union of all $\mathcal{F}_j$ for $\forall v_j \in MS_i$), $y_{ik} > 0$ is equivalent to $y_{ik} = 1$ since both of them mean link e_i is monitored by detection module f_k (redundant modules do not count). Thus, we define function $BN : \{\mathbb{Y}\} \longrightarrow \{0, 1\}$ that converts y_{ik} to a binary value, i.e., if $y_{ik} = 0$, $BN(y_{ik}) = 0$, otherwise $BN(y_{ik}) = 1$. For the example shown in Figure 1, matrices $\mathbb{Y}_{16\times6}$ and $BN(\mathbb{Y}_{16\times6})$ are as follows:

$$\mathbb{Y}_{16\times6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \mathbf{2} & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ \vdots & \vdots & & \cdots & & \vdots \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix},$$

$$BN(\mathbb{Y}_{16\times6}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \mathbf{0} & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ \vdots & \vdots & & \cdots & & \vdots \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

The objective function of *LCP* is non-linear. This is because the link-coverage requires the non-linear function *BN*. Thus, unlike PRIDE, *LCP* cannot be formulated as an ILP. In addition to non-linearity, *LCP* is more complex than path coverage problem (as defined in PRIDE) for a given network. This is because the number of paths to be covered is usually less than the number of communication links in WMN [14]. Moreover, as mentioned in Section 2.2, we aim for a scalable IDS solution that can be applied to large WMN (i.e., more links have to be monitored). Thus, we need to develop a technique to reduce the complexity of link coverage problem when compared to path coverage problem.

One can observe that matrix $\mathbb{D}$, for both 6-module and 12-module configurations [27], can be summarized as: i) every detection module requires the first group of preprocessors of size 15.6%; ii) every detection module requires either the second group of preprocessors (1.1% load) or the third group of preprocessors (1% load). We propose a *dependency relaxation* to run all three groups of preprocessors on every single node at the price of at most 1.1% extra load. Accordingly, the total memory load of node v_j, after the *IDS Function Distribution*, becomes $L_j = b_j + 17.7\% + \Sigma_{f_k \in \mathcal{F}_j} w_k^f$. However, it reduces the complexity of *LCP* when compared to path coverage problem in PRIDE.

Thus, *LCP* can be formulated as a non-linear optimization problem with integer (binary) variables as follows:

$$\text{maximize} \qquad \frac{1}{q}(\mathbf{1}^T \cdot \mathrm{BN}(\mathbb{M} \cdot \mathbb{X}) \cdot \mathbf{1}) \qquad (1)$$

$$\text{subject to:} \qquad B^T + (17.7)\mathbf{1}^T + \mathbb{X} \cdot {W^f}^T \leq \Lambda^T \qquad (2)$$

$$x_{jk} \in \{0, 1\} \qquad ,\forall j, k \qquad (3)$$

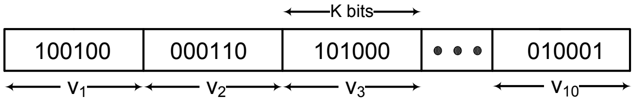


Figure 2: The matrix $\mathbb{X}$ for the 10-node mesh network shown in Figure 1 is encoded as a chromosome.

where the objective function is to maximize the average link coverage ratio in the network; constraint 2 limits the memory load on every node v_j to be less than its memory threshold λ_j; and constraint 3 forces x_{jk} to be either 1 or 0 meaning node v_j is either running module f_k or not.

5. RAPID PROTOCOL

In this section, we propose RAPID, a protocol to solve LCP in centralized and distributed manners. The centralized approach requires a base station that periodically collects nodes' information (e.g., network connectivity and memory utilization), solves LCP, and finally broadcasts IDS function distributions to the nodes. The distributed solution does not need the base station (i.e., nodes locally decide which detection modules they should run).

5.1 Centralized Solution

Given a modularization chosen by the security administrator for the IDS configuration (e.g., the 12-module configuration imposes higher execution time to the solver but is suitable for low memory thresholds [14]), the centralized RAPID periodically collects the local information from nodes, decides on an optimal set of detection modules to be executed by each node, and distributes them to the nodes. Since LCP has a non-linear objective function, linear constraints, and integer variables, we cannot use integer linear programming. Thus, we propose a Genetic Algorithm (GA), a popular and effective type of evolutionary algorithms.

GA starts with a set of *random* solutions and then derives better solutions using the Darwinian process of "survival of the fittest." The survival of the fittest process is iterative, and uses genetic operations, such as Selection, Crossover, and Mutation on the current set of solutions (from here on we will use "set of solutions" and "population" interchangeably). Selection gives the most fit solutions the chance to survive. Crossover combines solutions in each generation to produce offsprings (i.e., new solutions) of the next generation, and mutation is used to maintain genetic diversity in two consecutive generations. GA solutions are encoded as bitstrings (i.e., chromosomes) of specific length and tested for fitness. In our formulation, matrix $\mathbb{X}$ is a solution that can be encoded as a chromosome of length $n \times K$. Figure 2 depicts the chromosome corresponding to the IDS function distribution (i.e., the solution represented by matrix $\mathbb{X}$) of the WMN shown in Figure 1. The fitness (objective) value of each solution is the average LCR in the network. The genetic operations used in this article are based on operations explained in [39] that their details are omitted here.

The centralized RAPID protocol is presented in Algorithm 1 as performed on the base station. Given the set of WMN nodes, the base station first collects information from nodes and then produces matrix $\mathbb{M}$ (Lines 1 and 2). Moreover, matrix $\mathbb{X}$, number of initial solutions (*POP_SIZE*) and number of generations (*GEN_SIZE*) are initialized in Line 2. Next, the base station generates a set of *POP_SIZE random*

Algorithm 1 Centralized RAPID

```
 1: $Data\_Collection(V, E, n, q)$
 2: $Initialization(\mathbb{M}, \mathbb{X}, POP\_SIZE, GEN\_SIZE)$
 3: $Initial\_Solutions(POP\_SIZE, S_{\mathbb{X}})$
 4: $g = 1$
 5: while $g \leq GEN\_SIZE$ do
 6:     $Elitism(POP\_SIZE, S_{\mathbb{X}})$
 7:     $Selection(POP\_SIZE, S_{\mathbb{X}})$
 8:     $Crossover(POP\_SIZE, S_{\mathbb{X}})$
 9:     $Mutation(POP\_SIZE, S_{\mathbb{X}})$
10:     if $Stopping\_holds(\alpha)$ then
11:         $break$
12:     end if
13:     $g{+}{+}$
14: end while
15: $\mathbb{X} = Best\_Sol(S_{\mathbb{X}})$
16: $Sec\_BRDCST(\mathbb{X})$
```

solutions called $S_{\mathbb{X}}$ (Line 3). Starting from the first population, the Algorithm then iteratively performs genetic operations and creates another population for the next generation (Lines 4-9). The Algorithm stops generating a new population if either the number of generations exceeds *GEN_SIZE* (Line 5) or the stopping criteria holds (Lines 10-12), i.e., no improvement in the recent α optimal values has been observed, where α is set by the network administrator. Algorithm 1 then extracts matrix $\mathbb{X}$ from the best solution in $S_{\mathbb{X}}$ of last generation (Line 15) and securely broadcasts the IDS functions to the WMN nodes (Line 16).

5.2 Distributed Approach

The main purpose of a distributed approach for RAPID is to remove the communication overhead caused by message exchange between nodes/base station and the computation overhead of running GA for large networks. Additionally, this approach is adaptive to frequent path and topology changes where the base station might not have the most recent routing information (unless the nodes' information is collected frequently, which might incur very high communication and computation overhead). Hence, in the distributed RAPID (presented in Algorithm 2), each node, depending on its memory threshold, chooses a set of *random* detection modules to perform.

As shown in Algorithm 2, Line 1, each node requires some preliminary information such as the set of modules and their corresponding memory weights, set of preprocessors, and memory threshold λ which are assumed to be already set on the device by the security administrator. The base memory load b is obtained from system logs (Line 1) and added to the total memory load imposed by all preprocessors (Line 2). The algorithm then creates a new set of detection modules in a random order denoted by $\mathcal{F}'$ in Line 3. Next, detection modules in $\mathcal{F}'$ are iteratively checked (Lines 4-6) if they can be activated on the Snort configuration (depending on their memory weight and threshold λ). If so, the module will be activated and the total memory load L will be updated (Lines 7-8).

We will show that this approach works very well (produces near optimal solutions) and its performance surprisingly increases (i.e., achieves the centralized performance) in high memory thresholds or high network density. It is worth emphasizing that such a good performance is achieved without any communication overhead and with a very simple algorithm when compared to the centralized RAPID.

Algorithm 2 Distributed RAPID

```
 1: $Mod\_Setting(\mathcal{F}, \mathcal{C}, K, R, W^f, W^c, b, \lambda)$
 2: $L = b + \sum_{r=1}^{R} w_r^c$                    // $\sum_{r=1}^{R} w_r^c = 17.7\%$
 3: $Rand\_Perm(\mathcal{F}', \mathcal{F})$
 4: for $f = 1$ to $K$ do
 5:     $Mod = \mathcal{F}'(f)$
 6:     if $L + w_{Mod}^f \leq \lambda$ then
 7:         $Activate(\mathcal{F}, Mod)$
 8:         $L = L + w_{Mod}^f$
 9:     end if
10: end for
```

6. PERFORMANCE EVALUATION

In this section, we first demonstrate, through a proof of concept experiment using WMN hardware, that the ideas of link-coverage and multi-interface Snort are practical. Next, through extensive simulations, based on real data obtained from real-world WMN deployment and memory measurements, we evaluate the performance of our proposed centralized and distributed RAPID solutions. The main reason we use simulation is to be able to evaluate RAPID's performance for large networks and for different network densities, which are extremely difficult to be evaluated in a real testbed.

6.1 Proof of Concept Experiment

This section shows that link-coverage approach practically works in WMN. It also evaluates the performance of multi-interface Snort (as discussed in Section 3) and its extra memory load, when compared to the original Snort.

6.1.1 Multi-interface Snort

A common way for running Snort and other similar passive network monitoring applications on multiple network interfaces is to *bridge* all interfaces into a single *virtual* network interface (a process also known as "bonding"), and run a single instance of the IDS on that virtual interface. On a mesh router, however, this solution is not possible because in Linux the bonded interfaces cannot be configured with routable IP addresses, and consequently the router cannot perform its main task of routing packets. Another option would be to run two Snort instances, one for each interface. Snort includes support for running multiple instances, but due to its single-threaded design, each instance is a different process, with separate copies of all buffers and data structures. Although this approach works well for typical multi-core IDS sensors with ample RAM, it is not practical for a mesh router with very limited CPU and memory resources [14].

To be able to run a single Snort instance that receives traffic from both network interfaces without altering the network configuration of each interface, we followed an alternative approach and modified Snort to capture packets concurrently through two Libpcap handles. This is possible by opening two Libpcap packet capture handles, one for each interface, and then asynchronously retrieving packets from either handle through `select()`, whenever packets are available. A Libpcap handle can be put into "nonblocking" mode using `pcap_setnonblock()`, and then a file descriptor that can be monitored using `select()` can be obtained through `pcap_get_selectable_fd()`. This design allows us to: i) avoid the overhead of running a second Snort instance (context switches, duplicate data structures); ii) capture traffic from both local and upstream network interfaces concurrently; and iii) preserve the routing configuration of both interfaces. We experimentally observed that our multi-interface Snort imposes only $\sim 4\%$ extra memory load (compared to the original Snort) when running on a Netgear WNDR3700 router used in the PRIDE testbed.

6.1.2 *Experimental Verification*

We performed an experiment in a small-size indoor WMN to validate link-coverage monitoring and multi-interface monitoring. We note here that the idea of intrusion detection using a set of detection modules distributed on multiple WMN nodes was previously demonstrated and evaluated in PRIDE.

In our experiment, we used three Netgear WNDR 3700 routers (e.g., nodes A, B, and C) connected to each other creating a triangle WMN topology. Each router was configured to run a multi-interface Snort instance, monitoring network traffic on both 2.4 GHz (local traffic among its clients) and 5 GHz (WMN backbone traffic) wireless interfaces. Each of routers A and B had one client, while two clients (laptops) were connected to router C. Using the Rule to Attack (R2A) tool [27], we launched two different types of attacks: i) A's client targeting B's client (multi-hop attack); ii) a C's client targeting another C's client (single-hop attack). The corresponding detection modules for each attack were activated on multi-interface Snort running on node C. The alerts generated by the multi-interface Snort on router C proved the detection of both single-hop and multi-hop attacks simultaneously. Therefore, our proposed link-coverage (i.e., monitoring WMN backbone traffic on A-B link) and multi-interface Snort (i.e., monitoring both local and upstream interfaces concurrently) was shown to be practical for WMN.

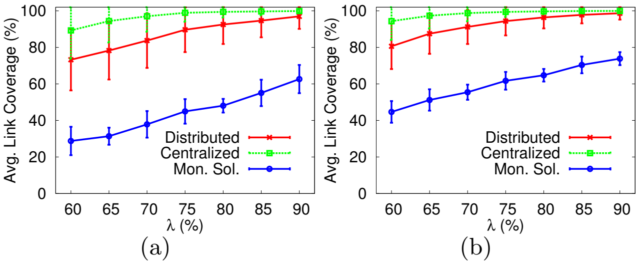
6.2 Simulation Results

We performed a thorough set of simulations to evaluate the performance of centralized and distributed RAPID in covering WMN links and detecting different types of attack. We compare our simulation results with PRIDE and monitoring node solutions as two state-of-the-art solutions. We implemented a monitoring node solution (*Mon. Sol.*) based on the formulation presented in [10]. The objective function, however, was changed to select nodes with higher total memory so that more detection modules can be run on monitoring nodes, thus having a fair comparison with RAPID.

All algorithms are implemented in MATLAB and run for different network sizes and densities. More precisely, our evaluation metrics are Average LCR in WMN, Average Memory Load on WMN nodes, and Average Intrusion Detection Rates for different types of attack with respect to two tuning parameters, Memory Threshold λ and Network Density. The average base line memory of the nodes (vector B) was 20%. Our simulation results are based on 6-module and 12-module configurations [27].

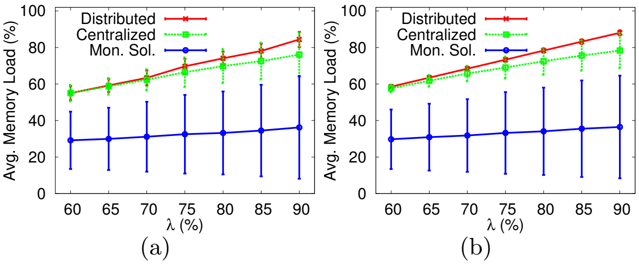
6.2.1 *Average LCR and Memory Consumption*

To evaluate the average LCR and average memory load, we created 100 random networks of size 30 (Note: PRIDE performance is evaluated on a 10-node WMN.) The average LCR and its standard deviation obtained from centralized RAPID, distributed RAPID and monitoring node solution are depicted in Figure 3. Figure 3(a) shows the average LCR for 6-module configuration while Figure 3(b) depicts the average LCR for 12-module configuration.



(a) (b)

Figure 3: The effect of λ on the average link coverage in: (a) 6-Module configuration; (b) 12-Module configuration.



(a) (b)

Figure 4: The effect of λ on the average memory load in: (a) 6-Module configuration; (b) 12-Module configuration.

As shown, the average LCR increases as λ increases which means more detection modules are executed on the nodes. The monitoring solution (consistent to the results shown for path coverage in [14]) has the minimum coverage ratio since the selected monitoring nodes are resource-constrained and cannot perform all detection modules and do not help each other to achieve higher coverage ratios. Obviously, the average LCR in centralized RAPID is higher than that of distributed RAPID as the centralized approach uses global information and produces optimal IDS distribution. The distributed RAPID, however, achieves an almost similar LCR to the centralized RAPID for large λ.

These results are comparable to the path coverage ratio obtained for 2-hop paths in PRIDE [14]. We note here that the execution time for the centralized RAPID is at most $\sim$ 5 seconds (for 30-node WMN) while it was more than 1 minute for the 10-node WMN in PRIDE (using ILP solver) and more than 1 hour for 30-node WMN. Moreover, when considering the average LCR, both centralized and distributed RAPID outperform PRIDE because RAPID uses the link-coverage approach, which allows more nodes to participate in traffic monitoring. As expected, the average LCR is slightly higher in 12-module configuration especially for small λ. This is because the size of detection modules are smaller than those in 6-module configuration, which allows more modules to fit in the small free memory spaces. It is worth mentioning that such a better performance obtained from 12-module configuration is at the price of *slightly longer execution time* in RAPID, PRIDE, and Mon.Sol.

The average memory load on WMN nodes and its standard deviation of all three IDS solutions for the 6-module and 12-module configurations are depicted in Figures 4(a) and 4(b), respectively. It is important to note that the average memory load for the RAPID solution (both centralized and distributed) is always higher than that of monitoring node solution, i.e., consistent with results shown in PRIDE. This is because in monitoring node solution, only monitoring nodes are assigned with detection modules and the non-monitoring nodes are not loaded with any detection modules. Therefore, only few selected nodes will have high memory load as opposed to RAPID where all WMN nodes are loaded with the maximum number of detection modules that can fit. The large standard deviation of average memory load in Mon.Sol. indicates the difference between total memory load on monitoring nodes and non-monitoring nodes.

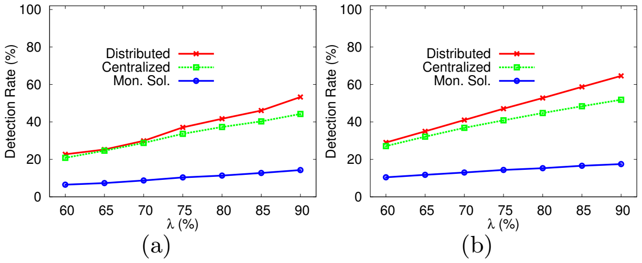
6.2.2 *Average Detection Rates for Different Attacks*

As mentioned in Section 3, we consider both single-hop (local) and multi-hop attacks in WMN. For a given WMN of size n, we simulated $10 \times n$ single-hop attacks and $2 \times n$ multi-hop attacks of random types (i.e., detectable by random detection modules as listed in [27]) and measured the detection rates based on the activated detection modules on the nodes.

Figure 5 shows the average detection rate of all $10 \times n$ single-hop attacks obtained from different IDS solutions. The results are produced for 100 random WMN of size n=30. Figure 5(a) depicts the average detection rates of single-hop attacks in all IDS solutions when the 6-module configuration is used. As shown, the larger the λ, the higher the detection rate is. This is because a larger memory threshold allows nodes to load and execute more detection modules and detect more local attacks, since the neighbors cannot help the node in detecting local attacks. As depicted in Figure 5(b), the average detection rate for 12-module configuration is slightly higher than those of 6-module configuration in all three IDS solutions. It is worth mentioning that, although the detection rates for both centralized and distributed RAPID are at most $\sim$ 60%, they are much better than for the monitoring node solution (i.e., at most $\sim$ 20%) and for PRIDE (i.e., 0% for local attacks).

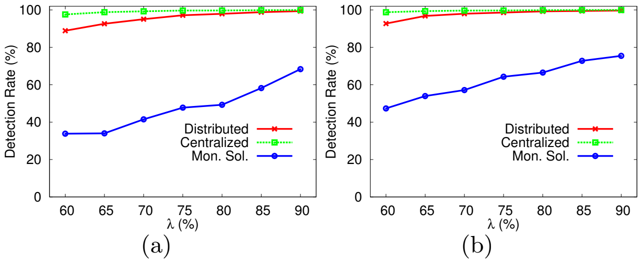
To evaluate the performance of IDS solutions in detecting multi-hop attacks, we considered 100 random networks of 30 nodes and 60 random paths. The path length of each attack is randomly chosen between 2 and 5 hops. Figures 6(a) and 6(b) depict the average detection rates of multi-hop attacks in all three solutions for 6-module configuration and 12-module configuration, respectively. As shown, the detection rates for multi-hop attack in RAPID and Mon.Sol. are much higher than those for single-hop attacks. This is because as traffic packets go through more IDS nodes, they will be more likely inspected by more distinct detection modules. Moreover, as previously observed, the larger the λ, the higher the detection rate will be. Also, the 12-module configuration again outperforms the 6-module configuration (at the price of slightly larger time complexity).

Figures 7(a) and 7(b) show the simulation results for average detection rates of compromised node attacks in 6-module and 12-module configurations, respectively, in all IDS solutions. The results are obtained from 100 random networks of 30 nodes where 60 random attacks are considered for each network. The compromised node is randomly chosen among WMN nodes to run either a single-hop (targeting a neighbor WMN node) or multi-hop attack. As shown, the results are slightly worse than multi-hop attacks because the compromised node itself is considered unable to detect the attack, which results in inspecting attack traffic with less detection modules.



(a) (b)

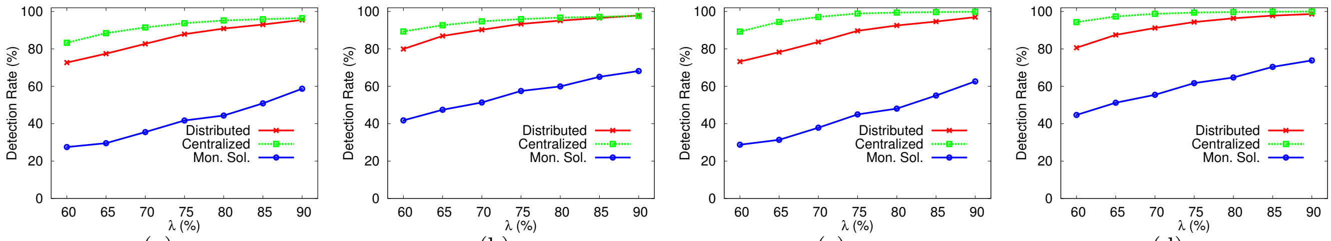
Figure 5: The effect of λ on the detection rate of single-hop (local) attacks in: (a) 6-Module configuration; (b) 12-Module configuration.



(a) (b)

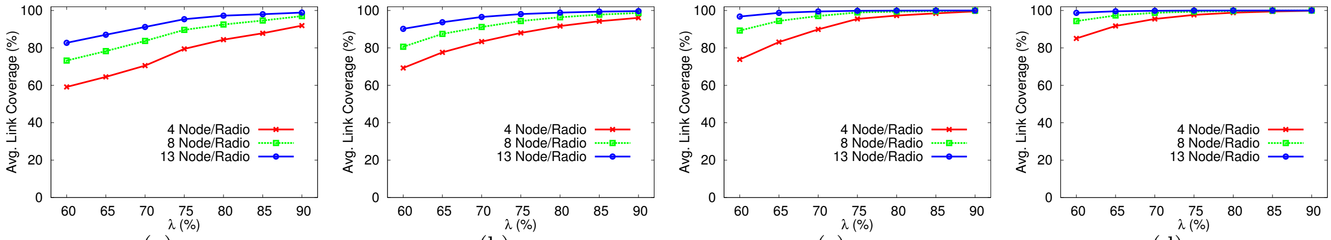
Figure 6: The effect of λ on the detection rate of multi-hop attacks in: (a) 6-Module configuration; (b) 12-Module configuration.

The last type of attack we consider for intrusion detection evaluation is unauthorized client attack. An unauthorized client is assumed to be physically located in WMN area but not associated with any of MAPs (i.e., outsider). The attacker can launch attacks against WMN nodes (e.g., DoS, battery depletion, spoofed de-authentication, etc.) or WMN links (e.g., jamming, blackhole/grayhole, etc.). We assume that in the attack against a WMN node, the target is unable to participate in the intrusion detection process. Figures 8(a) and 8(b) show the average detection rate of unauthorized client attacks targeting WMN *nodes* for 6-module and 12-module configurations, respectively. The results are obtained from 100 random networks of 30 nodes where 300 random attacker locations and targets are considered. The results show that these attacks are highly detectable by RAPID algorithms as opposed to Mon.Sol. solution that can achieve at most $\sim$ 60% detection rate. We note here that PRIDE cannot detect such attacks since the attack traffic is not routed through WMN nodes. Figures 8(c) and 8(d) show the average detection rate of unauthorized client attacks targeting WMN *links*, when using 6-module and 12-module configurations, respectively. As depicted, the results are slightly better than those targeting WMN nodes because



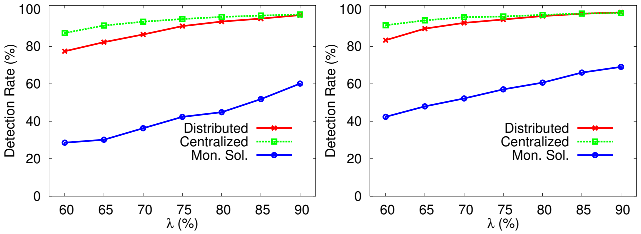
(a) (b) (c) (d)

Figure 8: The effect of λ on the detection rate of unauthorized client (outsider) attacks: (a) against nodes in 6-Module configuration; (b) against nodes in 12-Module configuration; (c) against links in 6-Module configuration; (d) against links in 12-Module configuration.



(a) (b) (c) (d)

Figure 9: The effect of λ and network density on the average link coverage in: (a) 6-Module configuration of distributed RAPID; (b) 12-Module configuration of distributed RAPID; (c) 6-Module configuration of centralized RAPID; (d) 12-Module configuration of centralized RAPID.



(a) (b)

Figure 7: The effect of λ on the detection rate of compromised node attacks in: (a) 6-Module configuration; (b) 12-Module configuration.

more nodes participate in monitoring the target WMN link.

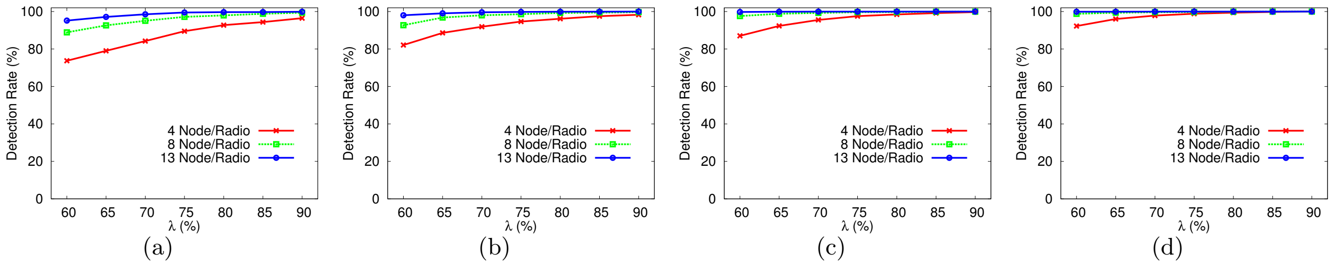
6.2.3 The Effect of Network Density on RAPID Performance

In order to show the effect of network density on performance of RAPID, we repeated all previous simulations (i.e., network density was 8 nodes per radio range) for two more network densities, 4 and 13 nodes per radio range. Intuitively, the higher the network density should result in participating more neighbors in traffic monitoring that would increase the average link coverage ratio and consequently the intrusion detection rate. In this section, we show the simulation results for average LCR and average detection rates of different attacks as functions of λ and network density.

Figures 9(a) and 9(b) show the average LCR obtained from distributed RAPID for 6-module and 12-modules configurations, respectively. The results confirm that the average LCR increases asλ or network density increase. Figures 9(c) and 9(d) depict the average LCR obtained from centralized RAPID for 6-module and 12-modules configurations, respectively. The results obtained from centralized RAPID are better than those obtained from distributed RAPID, at the price of some communication and computation overheads. We note here that the network density has no effect on the average LCR of PRIDE (because of using node-coverage instead of link-coverage approach) and Mon.Sol (because it only affects the number of monitoring nodes and not the number of detection modules they perform).

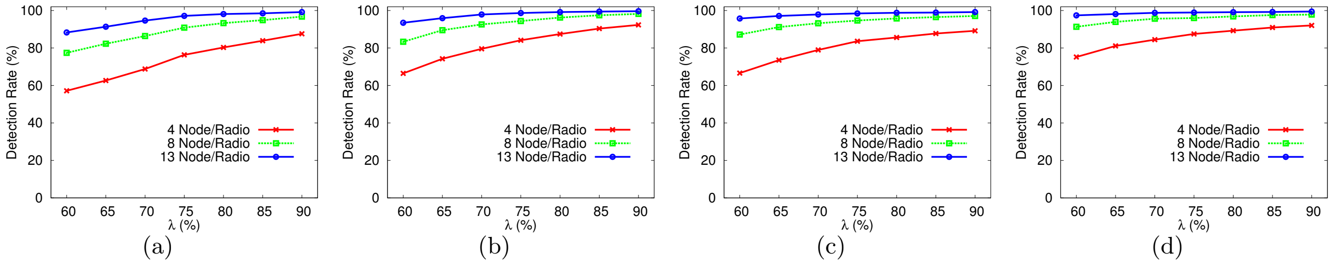
Figures 10(a) and 10(b) show the effect of λ and network density on the detection rate of multi-hop attacks in the distributed RAPID for 6-module and 12-modules configurations, respectively. Surprisingly, the multi-hop attacks are almost always detectable for $\lambda \geq 70\%$ and network density larger than 8 nodes per radio range in both 6-and-12-module configurations. Figures 10(c) and 10(d) show the results for the centralized RAPID which are above 90% even for the lowest network density and memory threshold. We note here that network density has no effect on single-hop attack detection as only one node (the local router) is responsible for intrusion detection and other WMN nodes do not participate in the intrusion detection process.

Figures 11(a) and 11(b) show the effect of λ and network density on the detection rate of compromised node attacks in the distributed RAPID for 6-module and 12-modules configurations, respectively. As depicted, the detection rate increases as network density and λ increase which means more nodes with more detection modules inspect the attack traffic generated by the compromised nodes. Figures 11(c) and 11(d) show the same results for the centralized RAPID when using 6-module and 12-modules configurations, respec-



(a) (b) (c) (d)

Figure 10: The effect of λ and network density on the detection rate of multi-hop attacks in: (a) 6-Module configuration of distributed RAPID; (b) 12-Module configuration of distributed RAPID; (c) 6-Module configuration of centralized RAPID; (d) 12-Module configuration of centralized RAPID.



(a) (b) (c) (d)

Figure 11: The effect of λ and network density on the detection rate of compromised node attacks in: (a) 6-Module configuration of distributed RAPID; (b) 12-Module configuration of distributed RAPID; (c) 6-Module configuration of centralized RAPID; (d) 12-Module configuration of centralized RAPID.

tively. The results show that centralized RAPID outperforms distributed RAPID, however, at the price of higher computation and communication overheads.

The effect of λ and network density on the detection rate of unauthorized client (outsider) attacks against WMN nodes in the distributed RAPID are shown in Figures 12(a) and 12(b) for 6-module and 12-modules configurations, respectively. Also, Figures 12(c) and 12(d) show the same results for the centralized RAPID when using 6-module and 12-modules configurations, respectively. The results confirm that the larger the λ and network density, the higher the detection rate will be. Moreover, centralized approach works better than distributed approach as 12-module configuration also works better than 6 module configuration.

Finally, we show the effect of λ and network density on the detection rate of unauthorized client (outsider) attacks against WMN links in both distributed and centralized RAPID. Figures 13(a) and 13(b) show the detection rates in the distributed RAPID for 6-module and 12-module configurations, respectively. The results are slightly better than those obtained from attacks against WMN nodes since more nodes participate in traffic monitoring. Figures 13(c) and 13(d) show the same results for centralized RAPID when using 6-module and 12-module configurations, respectively.

7. CONCLUSIONS

In this article, we showed that traffic-aware IDS solutions proposed for resource-constrained WMN are based on a strong assumption that might not hold for some WMN applications. We then investigated the scalability of state-of-the-art traffic-aware IDS solutions proposed for WMN and showed that their communication and computation overheads make them impractical for large scale WMN. Next, inspired by traffic-aware solutions and their limitations, we proposed a traffic-agnostic intrusion detection mechanism for resource-constrained WMN that is scalable and can be implemented in both centralized and distributed manners with lower computation and communication loads than traffic-aware solutions. Our proposed solution is based on a link-coverage approach in traffic monitoring and, unlike traffic-aware solutions that only detect multi-hop attacks, can detect both multi-hop and single-hop attacks. Through real-world experiments and extensive simulations, we showed that our IDS solution outperforms state-of-the-art IDS solutions proposed for resource-constrained WMN.

Acknowledgement

This work was funded in part by NABSSUP Fleet Logistics Center San Diego grant #N00244-12-1-0036 and NSF grants #1127449, #1145858, and #0923203.

8. REFERENCES

[1] H. Chenji, W. Zhang, M. Won, R. Stoleru, and C. Arnett, “A wireless system for reducing response time in urban search and rescue,” in *Proceedings of 31st IEEE International Performance Computing and Communications Conference (IPCCC)*, 2012.

[2] A. G. Fragkiadakis, I. G. Askoxylakis, E. Z. Tragos, and C. V. Verikoukis, “Ubiquitous robust communications for emergency response using multi-operator heterogeneous networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, p. 13, 2011.

[3] H. Chenji, A. Hassanzadeh, M. Won, Y. Li, W. Zhang, X. Yang, R. Stoleru, and G. Zhou, “A wireless sensor,

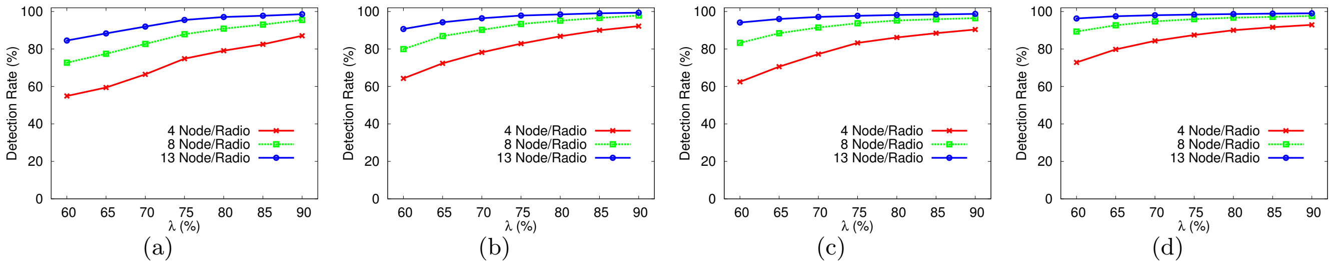


Figure 12: The effect of λ and network density on the detection rate of unauthorized client (outsider) attacks against nodes in: (a) 6-Module configuration of distributed approach; (b) 12-Module configuration of distributed approach; (c) 6-Module configuration of centralized approach; (d) 12-Module configuration of centralized approach.

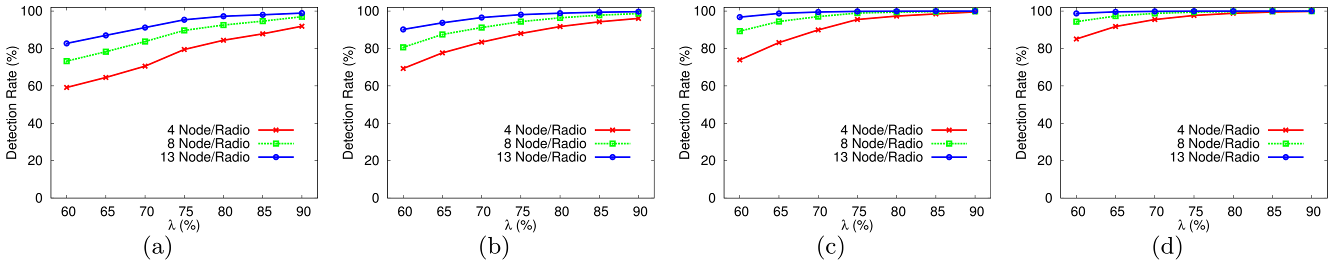


Figure 13: The effect of λ and network density on the detection rate of unauthorized client (outsider) attacks against links in: (a) 6-Module configuration of distributed approach; (b) 12-Module configuration of distributed approach; (c) 6-Module configuration of centralized approach; (d) 12-Module configuration of centralized approach.

adhoc and delay tolerant network system for disaster response," LENSS-09-02, Tech. Rep., 2011.

[4] D. Wu, D. Gupta, and P. Mohapatra, "QuRiNet: A wide-area wireless mesh testbed for research and experimental evaluations," *Computer Networks*, vol. 9, no. 7, pp. 1221–1237, 2011.

[5] J. Backens, G. Mweemba, and G. Van Stam, "A rural implementation of a 52 node mixed wireless mesh network in Macha, Zambia," *EInfrastructures and EServices on Developing Countries*, pp. 32 – 39, 2010.

[6] M. Adeyeye and P. Gardner-Stephen, "The Village Telco project: a reliable and practical wireless mesh telephony infrastructure," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, p. 78, 2011.

[7] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks and ISDN Systems*, vol. 47, no. 4, pp. 445–487, 2005.

[8] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A framework for misuse detection in ad hoc networks-part I," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 274 – 289, Feb. 2006.

[9] D.-H. Shin and S. Bagchi, "Optimal monitoring in multi-channel multi-radio wireless mesh networks," in *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2009.

[10] A. Hassanzadeh, R. Stoleru, and B. Shihada, "Energy efficient monitoring for intrusion detection in battery-powered wireless mesh networks," in *Proceedings of the 10th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW)*, 2011.

[11] F. Hugelshofer, P. Smith, D. Hutchison, and N. J. Race, "OpenLIDS: a lightweight intrusion detection system for wireless mesh networks," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2009.

[12] N. Saxena, M. Denko, and D. Banerji, "A hierarchical architecture for detecting selfish behaviour in community wireless mesh networks," *Computer Communications*, vol. 34, no. 4, pp. 548 – 555, 2011.

[13] A. Morais and A. Cavalli, "A distributed and collaborative intrusion detection architecture for wireless mesh networks," *Mobile Networks and Applications - Springer*, 2013.

[14] A. Hassanzadeh, Z. Xu, R. Stoleru, G. Gu, and M. Polychronakis, "PRIDE: Practical intrusion detection in resource constrained wireless mesh networks," in *Proceedings of 15th International Conference on Information and Communications Security (ICICS)*, 2013.

[15] V. Sekar, R. Krishnaswamy, A. Gupta, and M. K. Reiter, "Network-wide deployment of intrusion detection and prevention systems," in *Proceedings of the 6th International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT)*, 2010.

[16] S. Glass, V. Muthukkumarasamy, and M. Portmann, "Detecting man-in-the-middle and wormhole attacks in wireless mesh networks," in *Proceedings of the IEEE 23rd International Conference on Advanced*

Information Networking and Applications (AINA), 2009.

[17] D. Shila and T. Anjali, "A game theoretic approach to gray hole attacks in wireless mesh networks," in *Proceedings of the 27th IEEE Military Communications Conference (MILCOM)*, 2008.

[18] M. Kim, V. K. S. Iyer, and P. Ning, "Mrfair: Misbehavior-resistant fair scheduling in wireless mesh networks," *Ad Hoc Networks*, pp. 299 – 316, 2012.

[19] D. Makaroff, P. Smith, N. Race, and D. Hutchison, "Intrusion detection systems for community wireless mesh networks," in *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2008.

[20] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *Proceedings of the 7th Conference on USENIX Security Symposium (SSYM)*, 1998.

[21] *SNORT Users Manual v2.9.2*, The Snort Project, September 2011.

[22] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[23] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS)*, 2003.

[24] H. Kim, D. Kim, and S. Kim, "Lifetime-enhancing selection of monitoring nodes for intrusion detection in mobile ad hoc networks," *AEU - International Journal of Electronics and Communications*, vol. 60, no. 3, pp. 248–250, Mar. 2006.

[25] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, no. 5, pp. 545–556, 2003.

[26] Q. Gu, W. Zang, M. Yu, and P. Liu, "Collaborative traffic-aware intrusion monitoring in multi-channel mesh networks," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012.

[27] A. Hassanzadeh, Z. Xu, R. Stoleru, and G. Gu, "Practical intrusion detection in resource constrained wireless mesh networks," Texas A&M University 2012-7-1, Tech. Rep., 2012.

[28] B. Wang, S. Soltani, J. K. Shapiro, P. ning Tan, and M. Mutka, "Distributed detection of selfish routing in wireless mesh networks," Michigan State University, MSU-CSE-06-19, Tech. Rep., 2006.

[29] R. do Carmo and M. Hollick, "DogoIDS: a mobile and active intrusion detection system for IEEE 802.11s wireless mesh networks," in *Proceedings of the 2nd ACM workshop on Hot Topics on Wireless Network Security and Privacy (HotWiSec)*, 2013.

[30] A. Hassanzadeh and R. Stoleru, "Towards optimal monitoring in cooperative ids for resource constrained wireless networks," in *Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011.

[31] C. Liu and G. Cao, "Distributed monitoring and aggregation in wireless sensor networks," in *Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM)*, 2010.

[32] B. Sun, K. Wu, and U. W. Pooch, "Alert aggregation in mobile ad hoc networks," in *Proceedings of the 2nd ACM workshop on Wireless Security (WiSe)*, 2003.

[33] A. P. Lauf, R. A. Peters, and W. H. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 253–266, 2010.

[34] G. Li, J. He, and Y. Fu, "A distributed intrusion detection scheme for wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS)*, 2008.

[35] I. Krontiris, Z. Benenson, T. Giannetsos, F. C. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," in *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN)*, 2009, pp. 263–278.

[36] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C. Tseng, T. Bowen, K. Levitt, and J. Rowe, "A general cooperative intrusion detection architecture for MANETs," in *Workshop on Information Assurance*, 2005.

[37] S. Razak, S. Furnell, N. Clarke, and P. Brooke, "Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1151 – 1167, 2008.

[38] H. Yang, J. Shu, X. Meng, and S. Lu, "Scan: self-organized network-layer security in mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 261–273, 2006.

[39] A. Hassanzadeh and R. Stoleru, "On the optimality of cooperative intrusion detection for resource constrained wireless networks," *Computers & Security*, vol. 34, pp. 16 – 35, 2013.

[40] J. Eriksson, S. Agarwal, P. Bahl, and J. Padhye, "Feasibility study of mesh networks for all-wireless offices," in *Proceedings of the 4th International Conference on Mobile Systems, Applications, And Services (MobiSys)*, 2006.

[41] D. Manikantan Shila and T. Anjali, "Load aware traffic engineering for mesh networks," *Computer Communications*, vol. 31, no. 7, pp. 1460–1469, 2008.

[42] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Saade, and M. Rubinstein, "Routing metrics and protocols for wireless mesh networks," *Network, IEEE*, vol. 22, no. 1, pp. 6–12, 2008.

[43] G. Parissidis, M. Karaliopoulos, R. Baumann, T. Spyropoulos, and B. Plattner, "Routing metrics for wireless mesh networks," in *Guide to Wireless Mesh Networks*, ser. Computer Communications and Networks. Springer London, 2009, pp. 199–230.

[44] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: the WLAN mesh standard," *IEEE Wireless Communications*, pp. 104–111, Feb 2010.

[45] A. Hassanzadeh, R. Stoleru, and J. Chen, "Efficient flooding in wireless sensor networks secured with neighborhood keys," in *Proceedings of the 7th IEEE*

International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2011.

[46] “OpenWrt Wireless Freedom,” http://www.openwrt.org.

[47] M. Valero, S. S. Jung, A. S. Uluagac, Y. Li, and R. A. Beyah, “Di-Sec: A distributed security framework for heterogeneous wireless sensor networks.” in *Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM)*, 2012.