Measuring the Impact of Denial-of-Service Attacks on Wireless Sensor Networks

Michael Riecker, Daniel Thies and Matthias Hollick
Secure Mobile Networking Lab
Technische Universität Darmstadt
Mornewegstr. 32, 64293 Darmstadt, Germany
{michael.riecker, daniel.thies, matthias.hollick}@seemoo.tu-darmstadt.de

***Abstract*—Wireless sensor networks (WSNs) are especially susceptible to denial-of-service attacks due to the resource-constrained nature of motes. We follow a systematic approach to analyze the impacts of these attacks on the network behavior; therefore, we first identify a large number of metrics easily obtained and calculated without incurring too much overhead. Next, we statistically test these metrics to assess whether they exhibit significantly different values under attack when compared to those of the baseline operation. The metrics look into different aspects of the motes and the network, for example, MCU and radio activities, network traffic statistics, and routing related information. Then, to show the applicability of the metrics to different WSNs, we vary several parameters, such as traffic intensity and transmission power. We consider the most common topologies in wireless sensor networks such as central data collection and meshed multi-hop networks by using the collection tree and the mesh protocol. Finally, the metrics are grouped according to their capability of distinction into different classes. In this work, we focus on jamming and blackhole attacks. Our experiments reveal that certain metrics are able to detect a jamming attack on all motes in the testbed, irrespective of the parameter combination, and at the highest significance value. To illustrate these facts, we use a standard testbed consisting of the widely-employed TelosB motes.**

***Index Terms*—Wireless Sensor Networks, Denial-of-Service, Measurements**

I. Introduction

In the last years, the use of wireless sensor network (WSN) technology in several practical problems has grown in interest and relevance. From tracking industrial operations such as leakage detection and pipe pressure measurement [1] to monitoring critical infrastructures like the Golden Gate bridge [2], there is a wide range of real-life applications in which WSNs play an important role. Guaranteeing security in such prominent applications is an issue.

A standard way of detecting attacks is by making use of intrusion detection systems (IDSs). [3] explores the spatial correlation in the networking behavior of sensors in close proximity. In this approach, each sensor monitors its immediate neighbors and identifies outliers using Mahalanobis distances. Networking behavior is characterized here by packet dropping rate, packet sending rate, forwarding delay time, and the actual sensor readings. However, no reason is given why exactly those metrics are analyzed.

At the moment, the decision on which features are relevant for intrusion detection is rather arbitrary than scientifically justified. To identify the most pertinent features, we need to develop an understanding of the real-world effects of attacks on WSNs. In this work, we describe a systematic way to analyze these effects in a testbed consisting of TelosB motes. For this purpose, we collect a large number of local metrics under various combinations of parameters, such as topology and traffic intensity. The jamming and blackhole attacks we carry out are two benchmark denial-of-service attacks, which operate on the link and the network layer, respectively. By using statistical tests, we identify those metrics deviating significantly in an attacking scenario as compared to normal working conditions. The metrics are classified according to their distinction capabilities, naming the metrics with highest significance value on all motes throughout all parameter combinations as *Class A* metrics. The most promising *Class A* metrics can be integrated in detection mechanisms such as the one proposed in *Di-Sec* [4].

Our paper contributes to evaluate the effects of attacks against WSNs in a systematic way. Particularly, (1) in the case of denial-of-service attacks, we identify generally applicable metrics which are able to differentiate between attacking and non-attacking scenarios. (2) Our results lay the foundation for developing lightweight intrusion detection systems for WSNs, focusing on the most suitable metrics.

The remainder of this paper is organized as follows. In Section II, we present related work. Section III is devoted to describe our exhaustive scheme to analyze whether a metric is appropriate to detect an attack. Therefore, we describe our testbed, the topologies we have used, what type of attacks have been implemented and the set of metrics we have tested. In Section IV, we assess our procedure, and thus we classify metrics according to their response to attack detection. Finally, some concluding remarks are outlined in Section V.

II. Related Work

Systematic quantification of the effects of denial-of-service attacks on WSNs has been neglected in the field literature.

Xu et al. [5] analyzed several detection approaches for jamming attacks. They used three different metrics to distinguish between jamming and normal or congested traffic: (1) the averaged received signal strength indicator (RSSI), (2) the carrier sense time, and (3) the packet delivery ratio (PDR). The authors concluded that the combination of PDR

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

and RSSI is able to detect jamming attacks reliably. These same authors extended their work on jamming attacks in [6], and examined the impact of jamming attacks on the packet delivery ratio and implemented channel surfing techniques to cope with interference.

Also Zhao et al. [7] conducted a study on packet delivery performance in dense wireless sensor networks. Even though they take into account interfering transmissions at the MAC layer occuring during normal operation, an intended attack is not considered.

Another work dealing with packet delivery performance was presented in [8]. Hauer et al. evaluated the effects of WLAN interference on packet delivery performance in IEEE 802.15.4 body area networks.

Recently, Lu et al. [9] introduced a system to detect jamming attacks in time-critical networks. They proposed a new metric for performance quantification called the message invalidation ratio. Eventually, a message is regarded as invalid if the message delay is greater than a certain threshold. They studied the impact of jamming attacks on this particular metric.

Despite some of these works address the impact of jamming attacks on packet delivery performance, they lack a comprehensive analysis of other mote-level metrics, such as energy consumption or routing-related information. In addition, they are not concerned with different classes of denial-of-service attacks, such as blackhole attacks. To the best of our knowledge, we conducted the most comprehensive set of measurements in a real WSN testbed to study the effects of denial-of-service attacks on a large set of metrics. Our work focuses on individual and locally available metrics; thus, the normal operation of the WSN is not obstructed, and at the same time we keep the metric collection lightweight.

III. Methodology, Metrics and Attacks

In this paper, we quantify the effects of denial-of-service attacks in WSNs on a variety of metrics. Our work aims at establishing a comprehensive scheme to find out if there are metrics more susceptible to exhibit an altered behavior under attack than others. Also, we want to rank a set of metrics fitting typical WSNs according to their response in case of attack. This could be further applied to develop IDSs.

A number of factors might influence a WSN under attack. For example, in a dense WSN the attack effects should propagate faster. In a systematic fashion, we control the topology, the intensity of the normal data traffic in a WSN, and the transmission power in order to understand the impact of these factors on the metrics, and thus on attack detection. We consider two DoS attacks, one prohibiting other communication (jamming) and one misdirecting the traffic (blackhole).

In what follows, we describe in details the testbed we use, the underlying protocols, attack implementations and the metrics we analyze.

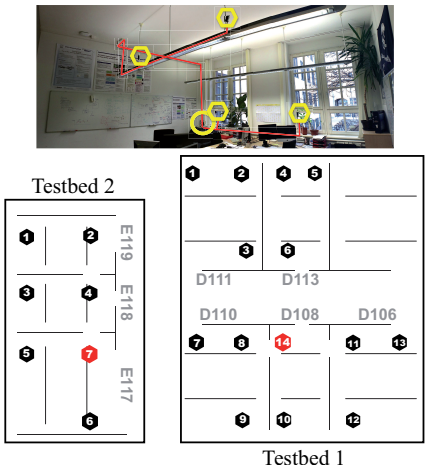

Fig. 1: Mote placement at the computer science building and sample deployment in one office (Image source: http://www.tudunet.tu-darmstadt.de).

A. Testbed

Our measurements are carried out in the TUDμNET[1], a federation of wireless sensor network testbeds deployed at various buildings of the Technische Universität Darmstadt. We have selected two different testbeds, i.e., subgroups of TelosB motes within the TUDμNET (Figure 1 is showing the mote placement in the corresponding offices; the attacker has node ID 14 in testbed 1 and node ID 7 in testbed 2). These motes provide a MSP430 MCU and a CC2420 radio chip. We run the operating system Contiki [10] using ContikiMAC. The first testbed contains 14 motes, located in neighboring office rooms of the computer science building. The second testbed is located at a different place of the TUDμNET to compensate for environmental influences or interferences, and to show the applicability of the analyzed metrics to other WSNs. This second testbed consists of 7 sensor nodes. The main difference between the two testbeds is the environment influencing the networks. While one testbed is located in office rooms with few people, the second is, among others, located in a pool room with frequently moving people and other interference resulting in a more challenging environment with respect to factors such as link quality. We also face common problems like unidirectional links and a constantly changing neighborhood. Hence, we address many challenges of real-world deployments. We vary a number of parameters such as:

- *Topology (mesh/collect)* - With the mesh and the collection tree protocol[2] [11], we consider two of the most widely-used protocols in WSNs.
- *Traffic (high/low)* - We want to analyze the impact of different traffic intensities on the detection capabilities of the metrics.
- *Transmission power (high/low)* - To vary the average node

[1] http://www.tudunet.tu-darmstadt.de

[2] In the remainder of this document we use the terms collect protocol as well as collection tree protocol interchangeably.

degree, we use two different transmission power settings.
- *Attack (jamming/blackhole/no attack)* - The influence of two different DoS attacks on the metrics is analyzed, and compared to an attack free scenario.

To generate artificial normal data traffic in the mesh network, messages containing a timestamp are exchanged between a random source and destination node on a regular basis. The nodes in the collect network periodically transmit messages to the base station. The intervals between the transmissions are set to 4 (for high traffic) and 10 seconds (for low traffic). Thus, the artificial traffic varies between 6 and 15 transmitted packets per minute in each node. Each packet has a message size of 6 bytes.

The varied transmission power leads to new topologies with changing network density. The CC2420 radio chip allows to set it to a value in the range of 1 (minimum) to 31 (maximum). For the first testbed, the transmission power is set to 10 for the "low power" setting, and to 16 for the "high power" setting. In the second testbed, the configuration is set to 8 and 16, because the second WSN is smaller and the distance between nodes is reduced.

During the series of measurements, each node periodically collects local metrics and makes them available through its serial port. The local collecting cycle time is set to 4 seconds. Note that all metric values are measured for the duration of the collecting cycle and then reset. The TUDμNET allows collecting all serial outputs in one centralized SQL database. This approach offers the advantage that all the metrics are collected over an out-of-band communication channel. Thus, the artificial traffic and our measurement collection do not interfere with each other.

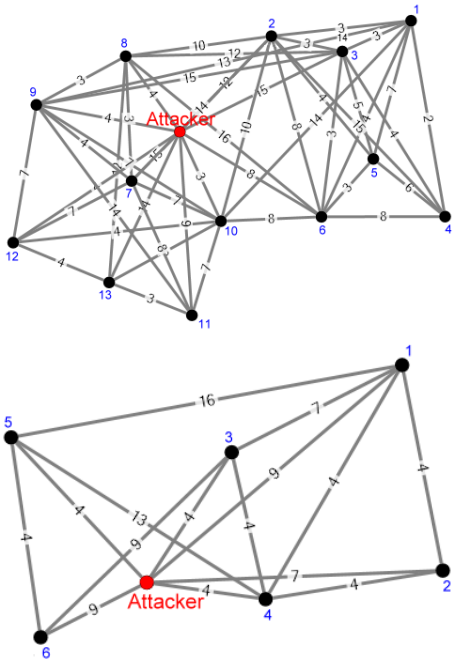

Fig. 2: Topologies of the two testbeds with high transmission power. Each arc connecting two nodes is labeled with the minimum transmission power value required to establish a path between both nodes.

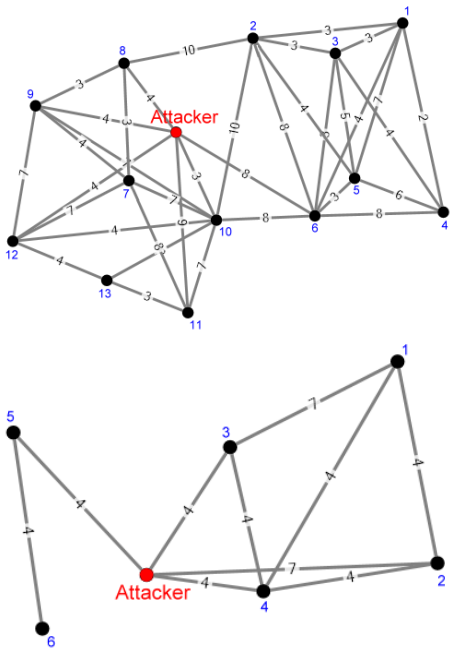

Fig. 3: Topologies of the two testbeds with low transmission power. Each arc connecting two nodes is labeled with the minimum transmission power value required to establish a path between both nodes.

Figures 2 and 3 depict the topologies of the two testbeds depending on the transmission power. Nodes are labeled with an ID number. Each arc connecting two nodes is labeled with the minimum transmission power value required to establish a path between both nodes. In the first collect testbed, node 13 is the base station. In the second collect testbed, the base station is located at node 1. When running the mesh protocol, base stations are not needed. The adversaries are placed close to the inner nodes of the two testbeds. To compensate for specific measurement errors or temporary anomalies, we perform three different test-runs at different daytimes for each combination of the testbed parameters *topology*, *traffic flow*, *transmission power* and *attack*. Each test-run has a duration of 15 minutes. We have chosen the test-run duration after assessing the time the network requires to reach steady state, which is about 1 minute. Overall, there are 144 measurement test-runs performed in both testbeds, providing about 3 million metric values.

B. Protocols Employed

We have chosen the mesh and collection tree protocol topologies, because they are among the most relevant protocols in WSNs. They are common for a lot of practical deployments [12], [13] and are used in scenarios such as central data collection and meshed multi-hop networks.

Collect Protocol: The collect protocol [11] is used in data

collection scenarios and provides mechanisms for building up a tree-based topology within the WSN. It also yields reliable hop-by-hop data packet forwarding to the base station at the root. Therefore, the collect protocol has desirable properties such as tracking network information and message delivery state, for example, packet successful reception. Furthermore, it has built-in methods for collecting statistical network data, which we use as protocol specific metrics in our analysis.

Mesh Protocol: The mesh protocol is implemented by Contiki's Rime stack and allows sending messages using multi-hop routing to specific destinations in the WSN. For finding the best route to the destination, each node manages an own routing table containing the next-hop neighbor for a specific receiver. However, the multi-hop forwarding mechanism cannot ensure packet delivery, since no acknowledgement packets are sent at the reception of data. Hence, the mesh protocol provides support for less metrics than the collect protocol.

C. Attack Implementations

An analysis of the literature reveals that the question of security has been mainly tackled from the cryptographic point of view, with a focus on data integrity and confidentiality. For instance, besides standard symmetric algorithms which run very efficiently on the resource-constrained nodes, also public-key cryptography has become feasible [14], [15]. Still, given the often unattended nature of sensor nodes, it is reasonable to assume an attacker physically compromising the nodes and gaining access to the cryptographic key material. Therefore, defense mechanisms against threats to the availability of the WSN, such as denial-of-service (DoS) attacks, and mechanisms to provide operational security are needed. Hence, we consider two (standard) denial-of-service attacks. First, we analyze a physical layer jamming as it is a simple but still powerful attack. Then, we evaluate a blackhole attack. We implement both attacks in the collection tree and mesh protocol, respectively.

Jamming: Jamming is a denial-of-service attack on the availability of the communication channel. In our implementation on the TelosB motes, we directly access the physical layer transmitting method provided via the radio chip. Using this method, the jamming node directly transmits the jamming packets to the environment. The implemented jammer is independent of the network protocols used in the WSN.

Blackhole: A blackhole node tries to attract all the neighborhood traffic. Instead of forwarding this traffic to the destination, it discards all incoming data packets. The implementations of this attack on the routing protocol need to be specifically adapted to mesh and collect networks. In the case of a mesh network, the blackhole node advertises route announcements with the best routing metric to all destinations and also replies to route requests by handling the incoming requests as the desired final destination and replying on behalf of it. The collect protocol also requires some modifications for attracting the traffic. The blackhole node periodically broadcasts announcement messages with routing information which are used for selecting the parent node. In particular, it sets its own announcement routing metric to one in order to trick the other nodes into selecting it as parent. Since the base station has announcement value zero, a blackhole setting this value to zero would be suspicious. The routing metric is not only announced directly, but is also piggybacked onto outgoing data or acknowledgement packets.

Taking into account these characteristics of jamming and blackhole attacks in both collect and mesh networks, it is possible to develop specific detection approaches. However, we are interested in the actual impact of these attacks on real wireless sensor networks.

D. Metrics

We identify an exhaustive list of metrics that have been selected based on two main criteria. First, we focus on metrics that are already provided by the node or the used protocols. Second, the metrics should be calculatable in a lightweight manner. This choice stems from efficiency reasons, as we envision the metrics to be used in lightweight IDSs. In our experiments, metrics can be divided into three categories: elementary metrics, collection tree specific metrics, and mesh network specific metrics. Due to the lack of acknowledgments in the mesh protocol, metrics such as the packet delivery rate are missing for this type of network. In the following, we describe our metrics that were directly provided by Rimestats/Energest in detail.

Elementary Metrics All sensor nodes can obtain elementary metrics, independently of the underlying protocol.

- *Received Signal Strength Indicator (RSSI):* RSSI represents the current radio signal power measured at the receiver and is typically expressed in dBm.
- *Transmit/Listen time:* It represents the amount of time the radio chip is in transmitting/listening mode during a specific time period. We measure the time with a timer of 8192 Hz.
- *Transmit/Listen duty cycle:* It depicts the usage of the transmit and listen time as percentage values with respect to the entire measurement period.
- *Transmitted/Received packets on network layer:* The packet rate metrics at the network layer contain counters for all outgoing and incoming packets.
- *Transmitted/Received packets on MAC layer:* It counts the outgoing and incoming packets on the MAC layer.
- *Packets with invalid CRC checksum:* It counts, how often a received packet is discarded because of an invalid CRC checksum.
- *Energy consumption by radio activities:* It is calculated as[3]

$$energy_r = (l \cdot 18.8 + t \cdot 17.4 + i \cdot 0.426) \cdot v$$

where $energy_r$ is the current energy consumption of the radio activities, l is the listen time, t is the transmit time, i is the idle time, and v is the current operating voltage.

[3]The specific values are taken from the CC2420/MSP430 manual

- *Radio load percentage:* This metric also represents the uptime of the radio. It is given as the percentage value of the uptime with respect to the entire measurement period.
- *Energy consumption by MCU activities:* It is calculated as[3]

$$power_{MCU} = m \cdot 0.5 \cdot v$$

where $power_{MCU}$ is the current energy consumption of the MCU, m is the MCU uptime, and v is the current operating voltage.
- *MCU load percentage:* This metric represents the percentage value of the MCU uptime with respect to the entire measurement period.
- *Contention drop:* This metric counts the number of times the node fails to send a packet due to a busy channel.
- *Pending packets:* This is a boolean metric indicating whether the node has unprocessed packets in the incoming packet buffer.
- *Too short packets:* When received packets are shorter than the footer plus the checksum, this counter is increased.

Metrics of the Collection Tree Protocol The collection tree protocol provides additional routing statistic measurements which can be used as possible detection metrics:

- *Transmitted/Received data packets:* These counters represent the amount of transmitted/received data packets using the collection tree protocol.
- *Transmitted/Received acknowledgement packets:* It counts the amount of transmitted and received ACK packets.
- *Received duplicate packets:* It describes the amount of received duplicate data packets.
- *Dropped packets by queue overload:* If the queue buffer for incoming data packets is overloaded, the next arriving packets will be discarded. This metric counts the occurrences of this event.
- *Packet delivery rates (PDR):* The PDR describes the fraction of successfully transmitted data packets.
- *Changing parent node:* If a sensor node is not able to communicate directly with the base station, it will connect to a parent node which then forwards its data traffic. The amount of changing events is counted by this metric.
- *Link estimation of best neighbor:* A node defines its parent node by choosing the neighbor with the lowest routing costs. The routing costs are calculated by either the link estimation or by the header information of incoming packets. This metric estimates the link quality to the best neighboring node.
- *Number of neighboring nodes:* This value represents the number of reachable nodes in the neighborhood.

Metrics of the Mesh Protocol The mesh protocol used in the testbed is based on a simple broadcasting of normal data messages. The following metrics are derived:

- *Number of direct neighbors in the routing table:* It represents the number of reachable nodes in the neighborhood. It is identified by those routing table entries having the same value for the next hop and the destination.
- *Number of entries in the routing table:* This metric contains the number of all entries in the routing table, and not only the direct neighbors as in the previous metric.

Having detailed the metrics we study under two different denial-of-service attacks, we will now illustrate a systematic way to assess the metric behavior. This assessment can also be applied for evaluating additional metrics in arbitrary protocols.

IV. Evaluation

In this section we describe the analysis of the collected metric data from the testbed and present the evaluation results. We particularly investigate the influence of the network density and the traffic intensity on the metric behavior.

A. Methodology

To determine whether the metric measured values under attack deviate significantly from those in an attack-free scenario, we perform a statistical test. We inspect the cumulative distribution function and perform the Kolmogorov-Smirnov test at a significance level of $\alpha = 0.05$ to check for normality. Both analyses show that there is evidence enough to assume the data not to be normally distributed and, hence, we carry out a non-parametric test. We have chosen the so-called Wilcoxon-Mann-Whitney test [16] to contrast whether there are statistically significant differences between *attack* and *attack-free* scenarios. The Wilcoxon-Mann-Whitney test is the analogue of the t-test without the assumption of normality. We use the open-source statistic tool R[4] for all tests.

Before starting the evaluation process, we first have to preprocess the collected data. We introduce a binary label (attack/no attack) to distinguish between values in an attack/normal scenario. Furthermore, we have to remove the first minute of each test-run, since the WSN is unbalanced during start-up, leading to wrong metric values. For instance, the PDR is always *-1* in the collect protocol, as no packets have been sent yet and thus the PDR cannot be calculated. Next, we group the obtained information during the three test-runs according to every combination of parameters separately for each metric and for each node. With a significance level of $\alpha = 0.05$ we test the null hypothesis that the attack values and the normal values have identical data distributions, and note the corresponding p-values. Therefore, if the p-value is less than the significance value, we reject the null hypothesis. The lower the p-values are, the more differ attack values from normal values. We also calculate for each run and for each node the arithmetic mean and standard deviation of the different metrics.

B. Metric Assessment

In order to assess the quality of a metric for distinguishing between attack and no attack, we classify them into four categories, namely A, B, C, and D metrics. This is done independently for the collect and mesh protocol, as they provide different metrics. As explained in Section IV-A, the

[4]http://www.r-project.org

TABLE I: Classification of the analyzed metrics for the different scenarios. Sc. 1: Jamming (Collect), Sc. 2: Jamming (Mesh), Sc. 3: Blackhole (Collect), Sc. 4: Blackhole (Mesh)

Metric	Class			
	Sc. 1	Sc. 2	Sc. 3	Sc. 4
RSSI	B	B	C	C
Transmit time	B	B	C	C
Transmit percentage	B	B	C	C
Listen time	A	A	C	D
Listen percentage	B	B	B	C
NET Sent pkts	B	B	B	C
MAC Sent pkts	B	B	B	D
Sent data pkts	B	N/A	B	N/A
Received data pkts	B	N/A	B	N/A
Sent ACK pkts	B	N/A	B	N/A
Received ACK pkts	B	N/A	-	N/A
Received duplicates	C	N/A	C	N/A
Dropped pkts	D	N/A	D	N/A
Packet delivery rate	A	N/A	B	N/A
Changing parent	D	N/A	D	N/A
Link estimation	B	N/A	B	N/A
No. of neighbors	A	A	B	C
No. of routing entries	N/A	A	N/A	C
NET Received pkts	B	B	B	C
MAC Received pkts	B	B	B	C
Invalid CRC	C	C	C	D
Radio energy	B	B	B	C
Radio load	B	B	B	C
MCU energy	B	B	B	C
MCU load	B	B	B	C
Contention drop	D	D	D	D
Pending pkts	C	D	-	-
Too short pkts	D	-	-	-

p-values are determined by performing the Wilcoxon-Mann-Whitney test. The classification is performed for each attack in the following way:

- Class A - These metrics are able to detect the attack in both traffic intensities, both transmit power settings, on all nodes in both testbeds, and with highest significance value (minimum and maximum p-values are lower than $2.2 \cdot 10^{-16}$, which indicates that the null hypothesis is rejected at all possible significant values $\alpha = 0.1, 0.05, 0.01, ...$).
- Class B - Metrics which can detect the attack in both traffic intensities, both transmit power settings, and having a minimum significance level lower than $2.2 \cdot 10^{-16}$ in both testbeds.
- Class C - Metrics that identify an attack in both traffic intensities, and both transmit power settings.
- Class D - All remaining metrics that are capable to disclose the attack.

This classification allows us to identify generally applicable metrics for attack detection (Class A), while others are only suited for specific scenarios or specific nodes (Classes B, C, and D). We admit that our classification is biased towards globally effective attacks and is dependent on the network size as well as on the strength of the attack. Thus, in a larger network there might be no Class A metrics at all. Still, it gives us a more fine-grained view on the impact of the implemented attacks on our testbeds.

C. Results

We find that several metrics are well-suited to detect the implemented attacks. From Table I we observe that the metrics in the collect topology constantly perform as good as the metrics in the mesh topology, and in some cases better (an entry in the table marked with "-" indicates there is not enough evidence to reject the null hypothesis; an entry marked with "N/A" signifies that the corresponding metric is not available in this protocol). In the collection tree protocol setting, the implemented traffic is more deterministic because all traffic is destined to the sink, whereas in the mesh protocol we use broadcast messages to different destinations. Thus, metrics related to network traffic statistics perform better in the collection tree protocol.

The main finding is that jamming attacks have a more significant global influence on the metrics than blackhole attacks, which tend to be locally restricted. Class A metrics are only available for jamming attacks. For example, the number of neighbors is significantly reduced. The most affected nodes are in the direct neighborhood of the jammer, having a neighbor count of zero whenever the WSN is jammed, as shown in Figure 4 (in all subsequent figures the error bars show the standard deviation of the metric values). While this metric can be obtained in both collect and mesh networks, another

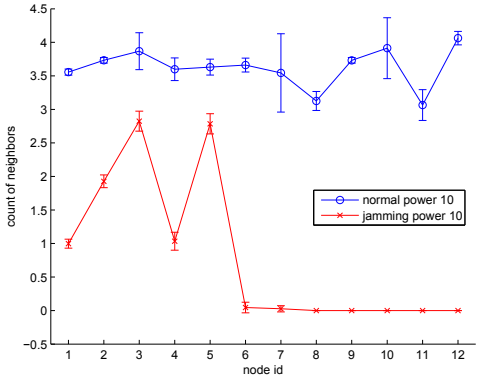

Fig. 4: Jamming attack in a mesh WSN with low traffic and low transmission power. The neighbor count is shown for every node.

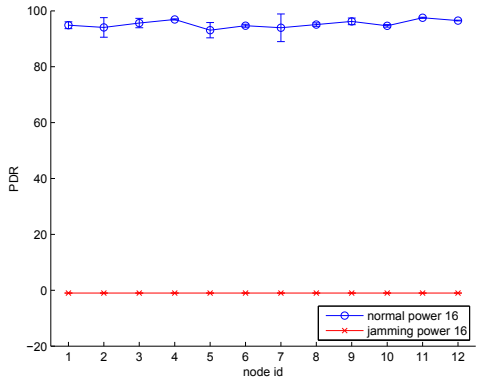

Fig. 5: Jamming attack in a collect WSN with low traffic and high transmission power. The PDR is shown for every node.

TABLE II: Influence of the Network Density.

Row	Metric	Topology	Traffic	Attack	p for Sparse Network	p for Dense Network
d1	Changing parent	Collect	Low	Jamming	$p < 1.6 \cdot 10^{-10}$	$p > 0.05$
d2	Contention drop	Collect	Low	Jamming	$p > 0.05$	$p < 3.9 \cdot 10^{-6}$
d3	Too short pkts	Collect	High	Jamming	$p < 6 \cdot 10^{-4}$	$p > 0.05$
d4	Changing parent	Collect	High	Jamming	$p < 7.3 \cdot 10^{-15}$	$p > 0.05$
d5	Dropped pkts	Collect	High	Jamming	$p < 2.2 \cdot 10^{-16}$	$p > 0.05$
d6	Changing parent	Collect	Low	Blackhole	$p < 1.2 \cdot 10^{-6}$	$p > 0.05$
d7	Changing parent	Collect	High	Blackhole	$p < 5.1 \cdot 10^{-7}$	$p > 0.05$
d8	Contention drop	Collect	High	Blackhole	$p > 0.05$	$p < 1 \cdot 10^{-6}$
d9	Dropped pkts	Collect	High	Blackhole	$p < 2.2 \cdot 10^{-16}$	$p > 0.05$
d10	Pending pkts	Mesh	Low	Jamming	$p > 0.05$	$p < 2.2 \cdot 10^{-16}$
d11	Contention drop	Mesh	Low	Jamming	$p > 0.05$	$p < 6.4 \cdot 10^{-5}$
d12	Contention drop	Mesh	Low	Blackhole	$p > 0.05$	$p < 6 \cdot 10^{-5}$
d13	Listen time	Mesh	Low	Blackhole	$p < 5.4 \cdot 10^{-5}$	$p > 0.05$
d14	MAC Sent pkts	Mesh	Low	Blackhole	$p < 7.7 \cdot 10^{-5}$	$p > 0.05$
d15	Contention drop	Mesh	High	Blackhole	$p < 3.6 \cdot 10^{-12}$	$p > 0.05$

metric which is only available for collect networks also reaches class A quality, namely the packet delivery rate. As shown in Figure 5, in an attack-free scenario the PDR is almost 100 for all nodes. In contrast, under a jamming attack, the PDR drops to zero, i.e., no packets can be transferred successfully. In a wireless sensor network with higher traffic, the average PDR in a normal scenario is reduced due to the higher amount of collisions. Besides, the PDR is also able to detect blackhole attacks, but in this case the effects of the attack are more local. The PDR is especially reduced for nodes that are not the direct neighbors of the base station and hence have to route their data via other nodes. In such a situation, the blackhole is effectively causing denial-of-service by dropping packets.

There is also a large number of Class B metrics that are heavily influenced by the attacks. Unsurprisingly, metrics covering traffic related information such as the number of sent/received packets are helpful in detecting the attack. However, not all nodes in the testbeds are affected in the same significant way, as some are more distant from the attack. Next, we want to give insights on the impact of selected parameters on the metrics' attack detection capability.

Influence of the Network Density We now investigate the influence of the network density on the metric distinction capabilities between an attacking scenario and the normal operation. Therefore, we describe the behavior of those metrics that are able to identify the attack in one network density setting, but fail to do so in the other density setting. We perform this analysis separately for the different network protocols and the different attacks. From now on, we call a network using the high transmission power a *dense* network. A network operating with the low transmision power setting is called a *sparse* network. An overview of the results is presented in Table II, in which we list the minimum *p-value* we calculated across all nodes in both testbeds, if we were able to reject the null hypothesis that the attack and the normal values of this metric have identical data distributions. Otherwise, p is greater than 0.05, which means that the metric in this density setting cannot differentiate between attack and normal operation.

Jamming: We start with analyzing the influence of the network density under a jamming attack on the metrics in the collect topology. If we compare the dense to the sparse WSN, we notice four differences. First, the jamming attack affects the sparse WSN stronger and thus causes parent changing events (rows d1 and d4). The effects of the jamming on the routing metric are more severe and influence the reachability of nodes. Second, the dense WSN is subject to a greater message dropping due to contention, because there is higher traffic than in a sparse network (row d2). Third, in a sparse network the count of invalid packets because of short packet size is higher (row d3). The jamming attack has a greater chance to corrupt messages because a sparse network does not suffer as much from contention as a dense WSN. The same reasoning explains the last difference. In a sparse wireless sensor network, there are no dropped packets due to queue overload in an attack-free scenario, since the overall traffic is lower. Consequently, packet dropping indicates the presence of jamming attacks (row d5).

Regarding the mesh topology, the results are similar: in a dense network, the number of messages dropped due to contention (row d11) and the number of pending packets (row d10) is higher.

Blackhole: Concerning the collect protocol, the blackhole causes a higher number of parent changing events in the sparse network due to the lower number of possible parents (rows d6 and d7). As Figure 6 shows, nodes that are not direct neighbors of the blackhole (node IDs 1-6) exchange their parent ID with the attacker. We also find that in an attack-free sparse network there is a low number of dropped messages caused by queue overload. A blackhole increases this count in a sparse network by actively advertising routes very often (row d9). In addition, a blackhole attack in a dense network provokes more message dropping at certain nodes due to contention (row d8).

For the mesh network we make the following observations. The number of contention drops is in general higher in a dense network with high traffic and is therefore not a significant metric for detecting blackhole attacks. However, an increase in this contention dropping rate is significant in a sparse network

TABLE III: Influence of the Traffic Intensity.

Row	Metric	Topology	Network Density	Attack	p for Low Traffic	p for High Traffic
i1	Too short pkts	Collect	Sparse	Jamming	$p > 0.05$	$p < 6.7 \cdot 10^{-4}$
i2	Contention drop	Collect	Sparse	Jamming	$p > 0.05$	$p < 1.3 \cdot 10^{-11}$
i3	Dropped pkts	Collect	Sparse	Jamming	$p > 0.05$	$p < 2.2 \cdot 10^{-16}$
i4	Contention drop	Collect	Sparse	Blackhole	$p < 5.6 \cdot 10^{-7}$	$p > 0.05$
i5	Dropped pkts	Collect	Sparse	Blackhole	$p > 0.05$	$p < 2.2 \cdot 10^{-16}$
i6	Contention drop	Mesh	Sparse	Jamming	$p > 0.05$	$p < 6.9 \cdot 10^{-5}$
i7	Pending pkts	Mesh	Dense	Jamming	$p < 2.2 \cdot 10^{-16}$	$p > 0.05$
i8	Contention drop	Mesh	Sparse	Blackhole	$p > 0.05$	$p < 3.6 \cdot 10^{-12}$
i9	Invalid CRC	Mesh	Sparse	Blackhole	$p > 0.05$	$p < 4.7 \cdot 10^{-5}$
i10	Listen time	Mesh	Dense	Blackhole	$p > 0.05$	$p < 7.1 \cdot 10^{-11}$
i11	MAC Sent pkts	Mesh	Dense	Blackhole	$p > 0.05$	$p < 2.2 \cdot 10^{-14}$
i12	Invalid CRC	Mesh	Dense	Blackhole	$p > 0.05$	$p < 1.3 \cdot 10^{-5}$
i13	Contention drop	Mesh	Dense	Blackhole	$p < 6 \cdot 10^{-5}$	$p > 0.05$

(row d15). In particular, we notice that the direct neighbor of the attacker in the second testbed (node ID 6) has a highly increased number of packets dropped due to contention. In a sparse network, traffic is flowing to the blackhole over fewer nodes, provoking an increased number of messages dropped due to contention at the direct neighbors of the blackhole. This behavior cannot be observed in the tree-structured collect protocol and is weakened in a low traffic scenario, where the corresponding metric is not significant in a sparse network, as opposed to a dense network (row d12). Further, we note that when a blackhole is active, the listen time for nodes on the route to the blackhole is increased in a sparse network, as more messages have to be transferred over those nodes (row d13). Similarly, the number of sent packets on the MAC layer is significantly reduced in a sparse network because messages are not forwarded by the blackhole (row d14).

Influence of the Traffic Intensity Equal to the analysis of the network density, in what follows we evaluate the impact of the traffic intensity on the metrics. For an overview of the results, please refer to Table III. Again, for significant metrics we give the minimum *p-value* we calculated across all nodes in both testbeds; otherwise p is greater than 0.05.

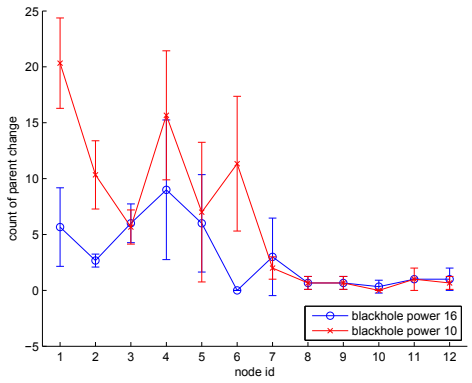

Fig. 6: Comparison of the effects of a blackhole attack in a low traffic collect WSN, depending on the density. The count of parent changing events is shown for every node.

Jamming: Investigating the metric behavior in the collect protocol, we remark three observations in a high traffic setting: (1) there is higher packet dropping due to contention (row i2) and (2) due to queue overload (row i3), and (3) the number of too short messages is higher (row i1). Thus, the jamming attack has a more severe negative effect on these three metrics, since more messages flow through the network.

Correspondingly, also the mesh protocol exhibits a higher packet dropping rate due to contention under high traffic (row i6). Besides, with low traffic and under normal operation, no pending packets are observed. A jamming attack increases the number of pending messages (row i7). In a high traffic wireless sensor network, this metric is not significant, as we also have pending packets without attack.

Blackhole: Focussing on the collect protocol, a high traffic results in more messages dropped due to queue overload (row i5), since a higher count of messages has to be transferred over fewer links. When the traffic is low, the number of messages dropped due to contention is significantly increased during a blackhole attack because of the malicious node blocking the channel with its route announcements (row i4).

Again, we observe similar results in the mesh protocol. In a high traffic setting, the number of packets dropped due to contention (row i8) and the number of packets with bad CRC checksum (row i9) is higher. This holds for nodes on the route to the blackhole, which experience an increased traffic flow. In contrast to the sparse network, the number of packets dropped due to contention is not significant in a dense high traffic WSN. Also without attack this number is relatively high, as opposed to the low traffic WSN experiencing a significant increase under a blackhole attack (row i13). Given high traffic, the number of packets with bad CRC checksum (row i12) and the count of sent packets on the MAC layer (row i11) is higher when compared to the low traffic WSN. Besides, in a high traffic WSN the listen time is reduced for nodes that are exposed to the blackhole dropping packets, while with low traffic there is no significant difference (row i10).

D. Discussion

Except for the metric dealing with too short packets, all other metrics we tested are at least able to detect the jamming

attack. The distinction capabilities in the case of a blackhole attack are a little worse, as the metric classification exhibits. The reason for these results is the different types of attacks. Jamming targets the physical or low layer communication medium, while the blackhole is an attack on the routing algorithms at higher layers. Therefore, it is difficult to find indications for a blackhole at the lower layers. One of the best blackhole metrics in the collect network is the link estimation, which measures the link quality to the neighbors with expected transmissions to the sink as cost metric. Routing algorithms make use of these values on higher layers. Therefore, the link estimation provides information about network anomalies with regard to manipulated values. For this reason, the metric might also have detection capabilities for sinkholes and wormholes, being also based on the same traffic attracting scheme.

The received packet rate on the MAC layer, counting all received regular packets using the radio chip, is another metric that can detect jamming attacks. The implemented attacker uses regular packets for the jamming, which is the reason for the highly increased values in the jamming scenarios. If a jammer uses a random signal without any packet structure, this metric might be unsuitable for jamming detection.

V. Conclusion and Future Work

In this work, the effects of DoS attacks on 28 distinct performance and network metrics in WSNs are studied in a systematic way. We identify widely applicable metrics and verify that they show a significantly different behavior under attack when compared to the baseline operation. The local metrics are able to detect jamming and blackhole attacks in a lightweight and practical way, since they are easily obtained without incurring too much overhead. Most of the metrics are already calculated by the lower network layers. Hence, it is possible to directly implement the intrusion detection mechanisms in the operating system of the sensor nodes to locally detect network anomalies.

We identify the packet delivery rate as a decisive metric to distinguish between attacking and normal scenario. Other highly significant metrics for jamming detection are the listen time and the number of neighbors; both can detect the attack on all nodes in the testbeds, and across all combinations of parameters. The effects of the blackhole attack are more locally restricted, yet we find several highly significant metrics that are able to detect the attack on selected nodes. Examples include the number of sent/received data packets, the link estimation of the best neighbor, and the radio energy consumption.

The analysis of the relationship between the analyzed metrics and important IDS metrics such as false-positive rates and detection time is part of our future work. We also plan to examine the combination of metrics. For example, the detection capability of the transmit duty cycle metric is higher in wireless sensor networks with high traffic. In contrast to that, the listen duty cycle metric has a better distinction capability in low traffic WSNs. Thus, the combination of these two metrics might be a relevant metric as well. A logistic regression can be applied to study such effects.

We focused on lightweight detection of two denial-of-service attacks, however, our approach to identify the quality of the metrics will also work to identify attacks other than DoS, thus paving the way to practical lightweight IDS in wireless sensor networks.

Acknowledgments

The work presented in this paper was performed in the context of the Software-Cluster project SINNODIUM (www.software-cluster.org). It was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant no. ”01IC10S01” and was supported by LOEWE CASED (www.cased.de). The authors assume responsibility for the content.

References

[1] P. Suriyachai, J. Brown, and U. Roedig, “Time-critical data delivery in wireless sensor networks,” in *DCOSS*, 2010.

[2] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, “Health monitoring of civil infrastructures using wireless sensor networks,” in *IPSN*, 2007.

[3] F. Liu, X. Cheng, and D. Chen, “Insider attacker detection in wireless sensor networks,” in *INFOCOM*, 2007.

[4] M. Valero, S. Jung, A. Uluagac, Y. Li, and R. Beyah, “Di-sec: a distributed security framework for heterogeneous wireless sensor networks,” in *INFOCOM*, 2012.

[5] W. Xu, W. Trappe, Y. Zhang, and T. Wood, “The feasibility of launching and detecting jamming attacks in wireless networks,” in *MobiHoc*, 2005.

[6] W. Xu, W. Trappe, and Y. Zhang, “Channel surfing: defending wireless sensor networks from interference,” in *IPSN*, 2007.

[7] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *SenSys*, 2003.

[8] J.-H. Hauer, V. Handzinski, and A. Wolisz, “Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks,” in *EWSN*, 2009.

[9] Z. Lu, W. Wang, and C. Wang, “From jammer to gambler: modeling and detection of jamming attacks against time-critical traffic,” in *INFOCOM*, 2011.

[10] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *LCN*, 2004.

[11] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *SenSys*, 2009.

[12] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, L. Mo, W. Dong, Z. Yang, M. Xi, J. Zhao, and X.-Y. Li, “Does wireless sensor network scale? A measurement study on greenorbs,” in *INFOCOM*, 2011.

[13] M. Ceriotti, M. Corra, L. D’Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. Jesi, R. Lo Cigno, L. Mottola, A. Murphy, M. Pescalli, G. Picco, D. Pregnolato, and C. Torghele, “Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels,” in *IPSN*, 2011.

[14] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in *PerCom*, 2005.

[15] A. Liu and P. Ning, “Tinyecc: a configurable library for elliptic curve cryptography in wireless sensor networks,” in *IPSN*, 2008.

[16] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, “How does your password measure up? The effect of strength meters on password creation,” in *USENIX Security*, 2012.