Part IV

Security and energy harvesting
Chapter 12

Hide-and-detect: forwarding misbehaviors, attacks, and countermeasures in energy harvesting-motivated networks

Sunho Lim¹, Cong Pu², Jinseok Chae³, Manki Min⁴, and Yi Liu⁵

Abstract

Multi-scale, heterogeneous, and battery-powered Internet-of-Things (IoT) sensors and devices (later in short, nodes) have been widely deployed in diverse applications and networks. Due to the limited amount of battery energy, energy harvesting-motivated networks (EHNets) powered by immediate environmental resources are increasingly popular and rapidly emerging as the next generation of ubiquitous communication infrastructure. However, EHNets are admittedly vulnerable to a denial-of-service (DoS) attack because of the shared medium, centralized coordination, and limited computing and communicating capabilities. Because of inherent resource constraints, EHNets seldom deploying conventional heavy-weight cryptographic techniques and secure algorithms and protocols. In light of these, we first investigate energy harvesting-based networking operations and applications. Second, we analyze the different types of forwarding misbehavior and attack caused by malicious nodes and their corresponding detection strategies. We introduce a set of adversarial scenarios and visualize its communication activities to capture vulnerable scenarios and potential malicious nodes. Here, single and multiple malicious nodes colluding together are considered. Lastly, we comprehensively compare the detection strategies of forwarding misbehavior by considering six perspectives and provide future research directions with interdisciplinary points of view.

¹T2WISTOR: TTU Wireless Mobile Networking Laboratory, Department of Computer Science, Texas Tech University, Lubbock, USA
²Department of Computer Sciences and Electrical Engineering, College of Engineering and Computer Sciences, Marshall University, Huntington, USA
³Department of Computer Science and Engineering, Incheon National University, Incheon, South Korea
⁴Computer Science Program, Louisiana Tech University, Ruston, USA
⁵Department of Computer and Information Science, University of Massachusetts, Dartmouth, USA
12.1 Introduction

Recent advances in technology have fueled the development of a tiny and low-power node available to expedite fast deployment and improve portability, availability, and accessibility. Internet-of-Things (IoT) sensors and devices (later in short, nodes) have been used in diverse applications and networks, where nodes are often multi-scale, heterogeneous, and battery-powered. Nodes are seamlessly interconnected for actuation, sensing, and communication activities. IoT applications have been deployed in a variety of areas, such as smart homes, healthcare, infrastructure monitor, transportation and logistics, surveillance, and so on. As the demand for IoT applications is rapidly increasing globally, the IoT market is predicted to reach more than 2 trillion by 2023, which is three times higher than 2016 \[1\]. We envision that IoT-based networks will not only play an important role in realizing diverse applications ranging from civilian to military but also become the next generation of ubiquitous communication infrastructure.

Since nodes are primarily powered by batteries, it is unavoidable to replace or replenish batteries. This could be a critical issue if multiple nodes are deployed in a hard-reach area or a very wide area. It would be hard (if it is not impossible) to manually replace or replenish batteries. In light of these, we investigate energy harvesting-motivated networks (EHNets) to replenish or at least reduce the number of times in replacing batteries. In EHNets, each self-sustainable node is equipped with energy harvesting capabilities and powered by an immediate environment, e.g., solar, wind, or thermal. Nodes can communicate with others directly or indirectly through multi-hop relays.

Although a great research effort has been allocated to energy harvesting literature, we focus on a cybersecurity issue in the sense of forwarding misbehaviors, attacks, and countermeasures in EHNets. In this chapter, we summarize our contribution in threefold:

- First, we explore energy harvesting aided applications and research areas, consider system and adversarial models of energy harvesting capable nodes, and raise a denial-of-service (DoS) issue in EHNets.
- Second, we present a set of adversarial scenarios, analyze the forwarding interactions between legitimate and malicious nodes, and identify vulnerable scenarios and potential forwarding misbehavior.
- Third, we comprehensively compare and analyze the detection strategies of forwarding misbehavior with six major perspectives and provide future research directions with interdisciplinary insights.

The rest of this chapter is organized as follows. We review forwarding misbehavior in EHNets in Section 12.2. Both system and adversary models are introduced in Section 12.3. Energy harvesting-motivated attacks with adversarial scenarios and their detection strategies are discussed in Sections 12.4 and 12.5. Finally, we discuss future research directions with interdisciplinary aspects and insights and conclude the chapter in Sections 12.6 and 12.7, respectively.
12.2 Background and related work

We explore energy harvesting techniques and their applicable network operations and analyze forwarding detection strategies deployed in battery-powered networks.

**Energy harvesting-motivated networks:** Wireless communication could be responsible for more than half of total energy consumption in wireless and mobile networks [2]. In light of this, power-efficient and power-aware routing techniques have been developed [3–6]. However, it is hard to locate and replace low-power batteries because nodes often operate for a long period in an unattended environment. Researchers in academia and industry have been focusing on energy harvesting from various environmental sources [7–12] in which each node’s battery can be rechargeable (or renewable).

In particular, energy harvesting from photovoltaic cells has been intensively investigated in the last two decades [8,9,13–26]. A variety of issues have been identified including solar-based routing and scheduling policy [8,9,15,21,27,28], resource allocation [19,23,29,30], energy synchronization [18], bounding communication delay [31], duty cycle [16], and data extraction [17] in multi-hop wireless networks. For example, a solar-based energy harvesting model is applied to scheduling and routing protocols [8,15]. The proposed energy harvesting-aware routing can increase network lifetime compared to that of traditional battery-based routing schemes. Several threshold policies are to maximize the communication performance in the network, where each node is assumed to be randomly recharged and changes its state into one of three states, active, passive, or ready [13]. A solar-aware routing scheme forwards packets to the nodes powered by solar energy [9].

Harvesting energy from ambient vibrations using a piezoelectric transducer has been investigated and applied to a wide range of civil and mechanical engineering applications for ease of battery replacement and energy replenishment [4,11,32–43]. Piezoelectric polymer patches are implanted into a living body to harvest energy from breathing [36]. Body heat, blood pressure, and even breath pressure have the potential to generate electric energy. Piezoelectric materials are used in the soles of shoes, where electrical power is generated through walking [4]. Researchers have also demonstrated the possibility of embedding a piezoelectric component in a textile [42]. Mechanical flow energy in oceans and rivers is utilized to convert electrical energy by using piezoelectric polymer actuators [39]. The piezo-based actuators can provide a large number of electrical power levels because of the vast size of the flowing water resource. Kinetic (motion)-based energy harvesting has received considerable attention [44–47].

**Detection of forwarding misbehavior:** A Watchdog technique and its variants [48,49] have been widely deployed to detect any communication misbehavior in infrastructure-based networks and infrastructure-less networks, such as mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs). This technique often relies on overhearing the packets transmitted around neighbor nodes and checks whether the packets are heading to the right receivers. Nodes continuously monitor and observe communication activities in the network, and thus, they are
required to stay in an active state for an extended period. Due to the non-negligible energy consumption, this technique cannot directly be applied to battery-powered networks.

Since nodes' communication activities can be spanned over multiple network layers, the algorithms and communication protocols embedded into the layers should not be conflicted. For example, IEEE 802.11 supports the power saving mechanism (PSM) in its medium access control (MAC) layer specification [50]. Each node can switch its state between active mode (AM) and power save (PS) mode. A node in AM stays awake all the time and conducts communication activities at any moment but wastes battery energy during idling. A node in PS periodically wakes up during the packet advertisement period and sees if there is any packet to receive. After staying awake and receiving any pending packet, the node puts itself to the low-power sleep state, PS, again for power saving. Thus, the Watchdog technique conflicts with the PSM embedded in the link layer. Although the Watchdog is not originally designed to work with the PSM, nodes are implicitly assumed to conduct communication activities in a resource-constrained environment.

We classify the detection strategies of forwarding misbehavior into three categories and briefly summarize their key ideas: (i) monitor, (ii) acknowledgment, and (iii) inducement. First, the basic idea of the monitor-based approach [51–55] is to check whether there is any forwarding misbehavior or network abnormality by observing the communication activities conducted among nodes, the amount of network traffic, or channel quality/condition. Second, the key operation of the acknowledgment-based approach [56–59] is that a set of designated nodes located between the source and the destination observes the forwarding operation of its very next node and sends an acknowledgment ($Ack$) packet to the source if an event or misbehavior is detected. Third, the basic idea of the inducement-based approach [60–62] is that nodes hide or fake their communication activities from malicious nodes to draw their forwarding misbehavior.

In summary, most detection approaches of forwarding misbehavior often require nodes not only to stay in an active state for an extended period but also to monitor/observe the communication activities via overhearing in a battery-supported network. Nodes are also supposed to generate a non-negligible number of control packets (i.e., $Ack$) to report any forwarding misbehavior that indeed consumes additional battery energy. Nevertheless, there is plenty of space to investigate EHNets with self-sustainable nodes that are under the charge-and-spend harvesting policy.

### 12.3 System and adversarial models

A system model mainly describes self-sustainable nodes, the energy harvesting process and policy, and initial network deployment. An adversarial model describes the potential misbehavior of malicious nodes in EHNets.

**System model:** First, a set of nodes is randomly distributed in a rectangle network, where each node can harvest energy. Nodes replenish their rechargeable
battery [63] periodically or non-periodically, such as an event driven. A piezo-based device is feasible to harvest energy from an immediate environment, such as disturbances or body movements. This piezo-based device can generate at least sufficient power for IoT nodes to transceive packets [44–46]. For example, the IEEE 802.15.4-compliant Texas Instrument Chipcon CC2420 radio can support a set of different transmission power levels from 3 $\mu$W to 1 mW [64]. The Cisco Aironet 340 and 350 series can also support four or six different transmission power levels [65]. In [66], both piezo devices and integrated self-charging power cells (SCPCs) can be combined to improve the efficiency of energy harvesting.

Second, a two-state Markov process is deployed to model an energy harvesting process: active and harvest states. Each node initially selects one of two states, spends a certain period, and changes the current state to the other. An average period spent in each state may vary depending on the deployed energy harvesting device and environmental resources. If a node changes the states in a short period frequently, both energy consumption and operational delay increase. To manage the energy efficiently and strategically, a charge-and-spend energy harvesting policy [22,52,62,67] is deployed in EHNets. Under this policy, a node in the harvest state cannot receive an incoming packet before it harvests a certain level of energy for communication. Nodes minimize the communication activities during the harvest state and replenish battery energy quickly. More importantly, each node in the harvest state periodically broadcasts a one-hop State packet to prevent its adjacent nodes from forwarding packets, resulting in a packet loss.

Third, when a node senses an event or detects an abnormality, it generates and forwards a sensed data packet toward a sink. We deploy a simple broadcast-based forwarding scheme to quickly propagate the packet to the sink [68]. To initially conduct a network deployment process, a one-time Hello packet contained with a field (number of hops, initially set to zero) is broadcasted at the sink [68]. When a node receives the Hello packet, it rebroadcasts the packet after increasing the packet’s number of hops by one. If the received packet contains a smaller number of hops, the node remembers the hop and rebroadcasts the packet. If not, the node discards the packet immediately. This procedure is repeated until all nodes receive and broadcast the packet. Finally, each node can identify its one-hop apart node(s) and how many hops are away from the sink. Then, the packet can be forwarded to single or multiple neighbor nodes that are located to the sink closer.

Fourth, we assume a reasonably dense network, where there are at least single or multiple nodes that can forward a packet. If two separate networks are connected solely by a single node, this node can be a single point of failure or a malicious node that may conduct forwarding misbehavior. Then, the network can easily be divided into two isolated sub-networks. Note that this network partition significantly affects the network performance in an infrastructure-less network, such as an EHNet, MANET, or WSN.

**Adversarial model:** First, single or multiple adversaries are to interfere with ongoing communications, intercept on-flying packets, and disrupt network algorithms and protocols. An adversary may physically capture a legitimate node and compromise it to behave maliciously, e.g., forwarding misbehavior. A malicious
node is assumed to have no energy constraint and stay in an active state as long as it wants.

Second, we consider three types of misbehavior. (i) A single malicious node may blindly drop incoming packets (i.e., blackhole attack) or selectively/strategically drop/forward incoming packets (i.e., selective forwarding attack) to a sink. (ii) A single malicious node may overhear/eavesdrop on on-flying packets, inject fake information, or alter packet header information to lead network traffic to the wrong destination. If a sender applies an authentication technique to a packet, such as a lightweight digital signature [69], then a receiver can detect whether the packet has been modified during the transmission. In this chapter, we focus on a DoS attack under diverse forwarding misbehavior scenarios that cannot be detected by cryptographic techniques. Thus, cryptographic primitives are out of scope. (iii) Multiple malicious nodes may collude together to hide their forwarding misbehavior.

### 12.4 Energy harvesting-motivated adversarial scenarios and attacks

We first observe and analyze a set of forwarding misbehaviors and adversarial scenarios and then briefly introduce corresponding detection schemes in EHNets. Single or multiple malicious nodes with control packet exchanges are investigated using a simple network topology to clearly see the forwarding misbehaviors and attacks.

#### 12.4.1 Single malicious node

**Adversarial scenarios, AS1:** These four adversarial scenarios are based on the overhearing of implicit acknowledgment in the network, where four energy harvesting enabled nodes interact as shown in Figure 12.1. Unlike an explicit acknowledgment by receiving an Ack packet, an implicit acknowledgment implies that a sender overhears if one-hop apart adjacent nodes have forwarded the received packet.

![Figure 12.1](image)

*Figure 12.1 A single malicious node (n_m), shaded as red, in a network, where solid and dashed-arrow lines mark a packet forwarding and overhearing, respectively*
First, suppose an active state sender \((n_a)\) transmits a packet to one-hop apart nodes, \(n_b\) or \(n_m\). If the sender is currently in a harvest state, it does not forward the packet until it becomes an active state. Whenever the state is changed, the sender broadcasts a one-hop State packet. Suppose \(n_a\) sends a packet to \(n_m\) in Figure 12.1 (a). \(n_b\) can overhear the packet and temporarily cache it in the local storage. If \(n_m\) is in the harvest state, \(n_a\) would send the packet to \(n_b\). If \(n_m\) is in the active state and forwards the packet to \(n_c\), both \(n_a\) and \(n_b\) can overhear the packet as an implicit acknowledgment. If \(n_m\) simply holds or discards the packet, both \(n_a\) and \(n_c\) cannot overhear the packet. Since \(n_b\) overheard the packet before, it can forward its cached packet to \(n_c\) directly after a timeout period, as depicted in Figure 12.1(b). \(n_a\) can now overhear the packet from \(n_b\) and may suspect the forwarding behavior of \(n_m\).

Second, suppose \(n_m\) receives the packet from \(n_a\) and forwards it to \(n_c\), while \(n_c\) is in the harvest state. If \(n_b\) is in the harvest state, \(n_m\) would not be suspected because \(n_a\) can still overhear the packet from \(n_m\). Third, if \(n_b\) is in the active state, it suspects \(n_m\) because \(n_b\) knows that \(n_c\) is in the harvest state. Thus, \(n_b\) forwards its cached packet to \(n_c\) after the timeout period. If \(n_a\) overhears the packet forwarded from \(n_b\) rather than the original forwarder \(n_m\), it may suspect the forwarding behavior of \(n_m\). Fourth, suppose \(n_m\) receives a packet from \(n_a\), changes its state to harvest from active, and does not broadcast the State packet. \(n_b\) can forward its cached packet to \(n_c\) because it is in the active state. If \(n_b\) is in the harvest state, but \(n_a\) cannot overhear the packet forwarded from \(n_m\), \(n_a\) considers \(n_m\) as a failure node and tries to find other forwarding candidate nodes.

In Figure 12.2, we highlight the vulnerable cases in which a malicious node can show forwarding misbehavior. The first misbehavior case is shown in Figure 12.2(a). When \(n_c\) is in the harvest state, \(n_m\) tries to forward the packet received from \(n_a\) to \(n_c\). Similarly, \(n_m\) tries to forward the packet to \(n_c\) when \(n_b\) and \(n_c\) are in the harvest state concurrently, as shown in Figure 12.2(b). This is the second misbehavior case because \(n_a\) overhears the forwarded packet from \(n_m\) and considers the forwarding operation as valid.

**Cooperative detection:** A hop-by-hop cooperative detection (HCD) scheme [52] is proposed to discourage forwarding misbehavior by reducing the forwarding probability of malicious nodes in EHNETs. The basic idea is that each node overhears the communication activities conducted around its neighbor nodes and

---

**Figure 12.2** Vulnerable cases in adversarial scenarios, where a malicious node shows forwarding misbehavior in a network
records the trace of forwarding operations. This is different from the prior approach, in which each node reports a suspected malicious node either to a source node [48,56,57] or to a centralized server (i.e., credit clearance service) [70]. Then, the source node or server decides how to assign a credit/penalty and whether to isolate the malicious node from participating in the communication activities in the network, accordingly.

12.4.2 Single malicious node with an additional control packet

**Adversarial scenarios, AS2:** We enhance the adversarial scenarios of AS1 by including an additional control packet, *Wait*, in the network, as depicted in Figure 12.3.

The first scenario depicted in Figure 12.3(a) is the same as the first scenario shown in AS1. Here, since \( n_a \) and \( n_b \) are in the active state, \( n_m \) does not hold or drop the packet on purpose. This is because the forwarding misbehavior of \( n_m \) can be detected by either \( n_a \) or \( n_b \). Thus, \( n_m \) forwards the packet just like a legitimate node. Second, as shown in Figure 12.3(b), suppose the harvest state \( n_c \) periodically broadcasts a *State* packet to its one-hop adjacent nodes. To avoid any forwarding misbehavior suspect, \( n_m \) simply holds the packet and waits until \( n_c \) changes the state back to active and broadcasts another *State* packet. Then, \( n_m \) sends a *Wait* packet back to \( n_a \) and behaves like a legitimate node. After receiving the *Wait* packet, \( n_a \) can select other forwarding candidate nodes, e.g., \( n_b \). Third, suppose the harvest state \( n_b \) periodically broadcasts a *State* packet to one-hop neighbor nodes as shown in Figure 12.3(c). This is similar to the case when \( n_c \) is in the harvest state and broadcasts a *State* packet. \( n_m \) may send a *Wait* packet to \( n_a \) to intentionally

**Figure 12.3** Adversarial scenarios with an additional control packet are depicted. Here, a malicious node (\( n_m \)) is marked as red, and nodes in the harvest state (\( n_b \) or \( n_c \)) are marked as shade. Forwarding, overhearing, and broadcasting operations are marked as solid, dotted, and dashed–dotted lines, respectively.
delay the packet transmission. However, \( n_c \) can overhear the Wait packet and may suspect \( n_m \) of forwarding misbehavior. To avoid a forwarding misbehavior suspect, \( n_m \) behaves just like a legitimate node and forwards the packet to \( n_c \) rather than holding or dropping the packet on purpose. Fourth, when both \( n_b \) and \( n_c \) are in the harvest state, they periodically broadcast a State packet, as shown in Figure 12.3(d).

Since the neighbor nodes of \( n_m \) (\( n_b \) and \( n_c \)) become blind, \( n_m \) can forward the packet to \( n_c \) and incur a packet loss intentionally. This is a vulnerable case, in which the forwarding misbehavior of malicious nodes cannot be detected even though \( n_a \) overhears the packet from \( n_m \).

**Camouflage-based active detection:** A camouflage-based active detection (CAM) scheme [62] is proposed to detect the forwarding misbehavior of malicious nodes in EHNetworks. In the CAM, each node hides its current state and does not broadcast a State packet. A harvest state node pretends to conduct energy harvesting but, in fact, observes the communication activities of neighbor nodes to detect a lurking malicious node. The CAM is different from the prior approach [48,49,52,71,72], where each node passively observes the routing operations in the network.

### 12.4.3 Multiple malicious nodes

**Adversarial scenarios, AS3:** We further investigate eight adversarial scenarios, in which five energy harvesting enabled nodes are deployed in the network, as shown in Figure 12.4. In AS3, two malicious nodes (\( n_{m1} \) and \( n_{m2} \)) located along with the forwarding path observe communication activities and collude together for a selective forwarding attack. Suppose a sender (\( n_a \)) forwards a Data packet to \( n_c \) via intermediate nodes, \( n_b \), \( n_{m1} \), and \( n_{m2} \).

The first scenario depicted in Figure 12.4(a) is similar to the aforementioned scenarios in AS1 and AS2 except for two malicious nodes in the network. When a sender (e.g., \( n_a \), \( n_{m1} \), or \( n_{m2} \)) forwards the received packet, a set of adjacent nodes (e.g., \( n_a \), \( n_b \), or \( n_{m1} \)) can overhear and cache the packet in their local storage. If both \( n_a \) and \( n_b \) are in the active state, \( n_{m1} \) and \( n_{m2} \) do not conduct any forwarding misbehavior by holding or dropping the packet on purpose. This is because \( n_a \) and \( n_b \) can overhear any forwarded packet. Thus, \( n_{m1} \) and \( n_{m2} \) forward the packet just like a legitimate node.

Second, suppose \( n_{m2} \) intentionally changes to the harvest state and broadcasts a State packet, as depicted in Figure 12.4(b). If \( n_b \) is in the active state, \( n_{m1} \) may forward the received packet to \( n_{m2} \) on purpose, resulting in a packet loss. Since the active state \( n_b \) can overhear the forwarded packet, this forwarding operation may be suspected as misbehavior. Thus, \( n_{m2} \) replies a Wait packet to the sender, \( n_a \), for delaying the packet transmission intentionally. Third, suppose harvest state \( n_c \) broadcasts a State packet as depicted in Figure 12.4(c). Both \( n_a \) and \( n_b \) are in active state and can overhear the communication activities. Both \( n_{m1} \) and \( n_{m2} \) behave as legitimate nodes and forward the received packet to the next-hop neighbor nodes. Note that \( n_{m2} \) may conduct forwarding misbehavior without being detected by simply forwarding the packet to \( n_c \), resulting in a packet loss. Fourth, if \( n_c \) is in the harvest state, as depicted in Figure 12.4(d), \( n_{m1} \) should play as a legitimate node not to gain any suspect of forwarding misbehavior.
Figure 12.4 Adversarial scenarios with two colluding malicious nodes in a network. Here, a malicious node is marked as red and a harvest state node is shades, respectively. Forwarding, overhearing, and broadcast operations are marked as solid, dotted, and dashed–dotted lines, respectively.
Fifth, suppose harvest state $n_b$ broadcasts a *State* packet in Figure 12.4(e), where $n_{m_2}$ may forward the received packet to $n_{m_4}$. Then, $n_{m_6}$ may not forward the packet to the one-hop adjacent nodes but instead maliciously hold or drop it. It is hard for $n_b$ and $n_c$ to detect this forwarding misbehavior of $n_{m_2}$ and $n_{m_4}$, because they cannot overhear any forwarded packet. Here, $n_b$ is in the harvest state and $n_c$ is far away from $n_{m_4}$.

Sixth, as depicted in Figure 12.4(f), both legitimate and malicious nodes (i.e., $n_b$ and $n_{m_6}$) are in the harvest state and broadcast a *State* packet, respectively. In this vulnerable case, $n_a$ cannot detect forwarding misbehavior of $n_{m_2}$ because $n_b$ is in the harvest state and even cannot forward its cached packet after the timeout period. Thus, $n_{m_2}$ may forward the received packet to $n_{m_4}$ on purpose, resulting in a packet loss. Since the forwarder node, $n_{m_4}$, is also a malicious node, both $n_{m_4}$ and $n_{m_6}$ may collude together for the forwarding attack. Seventh, suppose both harvest state $n_b$ and $n_c$ broadcast a *State* packet as depicted in Figure 12.4(g). This is another vulnerable case because both $n_{m_2}$ and $n_{m_6}$ can collude together for forwarding misbehavior without being detected. Since both $n_b$ and $n_c$ are blind, they cannot overhear any communication activity in the network. $n_{m_4}$ may keep quiet when it receives the forwarded packet from $n_{m_6}$, ultimately resulting in a packet loss. Lastly, suppose $n_c$ is in the harvest state as depicted in Figure 12.4(h). $n_{m_4}$ may still forward the received packet to $n_{m_6}$ on purpose without being detected.

According to the analysis of adversarial scenarios, if more than one malicious node is located consecutively in a sparse network, it would be hard to detect their collusion of forwarding misbehavior. As aforementioned, the network should be dense enough not only to prevent network partition but also to discourage forwarding misbehavior.

**Inducement- and monitor-based detection:** The proposed countermeasure, called EYES, is to detect and discourage the forwarding misbehavior of colluding malicious nodes in EHNets [73]. The EYES is different from the prior approach [48, 49, 56–58, 71, 72, 74–78], in which nodes passively monitor any forwarding misbehavior in the battery-powered networks. The EYES consists of inducement- and monitor-based sub-schemes, called SlyDog and LazyDog, respectively. The SlyDog is extended from the CAM [62]. Each node pretends to harvest energy without monitoring, but, in fact, it observes the forwarding operations conducted in one-hop neighbor nodes to efficiently detect shy malicious nodes and collusion of malicious nodes. In the LazyDog, each node counts the number of received/overheard packets and requests this information to its one-hop neighbor nodes. Then, each node can receive the information from its two-hop neighbor nodes and analyze the information to detect forwarding misbehavior.

### 12.5 Comparison and analysis of detection strategies

In Table 12.1, we summarize and categorize the detection strategies of forwarding misbehavior deployed in diverse networks. We analyze the strategies using six key perspectives.

- **Collusive attack:** We investigate whether multiple malicious nodes collude together to conduct a forwarding attack and achieve their attack goal(s) in the
Table 12.1  The comparison of detection strategies of forwarding misbehavior extended from [73]

<table>
<thead>
<tr>
<th>Approach</th>
<th>Collusive attack</th>
<th>Computation overhead</th>
<th>Communication overhead</th>
<th>Detection latency</th>
<th>Punishment</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watchdog [48]</td>
<td>N</td>
<td>Low</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Stand-alone</td>
</tr>
<tr>
<td>CDS [51]</td>
<td>N</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>HED [54]</td>
<td>Y</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Y</td>
<td>Distributed</td>
</tr>
<tr>
<td>HCD [52]</td>
<td>N</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Y</td>
<td>Distributed</td>
</tr>
<tr>
<td>CHEMAS [57]</td>
<td>N</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>CAD [79]</td>
<td>N</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>SCAD [80]</td>
<td>Y</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>APS [59]</td>
<td>N</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>Y</td>
<td>Distributed</td>
</tr>
<tr>
<td>CBDS [60]</td>
<td>Y</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>N</td>
<td>Distributed</td>
</tr>
<tr>
<td>SNBDS [61]</td>
<td>Y</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>N</td>
<td>Distributed</td>
</tr>
<tr>
<td>CAM [62]</td>
<td>N</td>
<td>Low</td>
<td>N</td>
<td>Medium</td>
<td>Y</td>
<td>Stand-alone</td>
</tr>
<tr>
<td>ACIDS [81]</td>
<td>N</td>
<td>Medium</td>
<td>N</td>
<td>Medium</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>SCM [75]</td>
<td>N</td>
<td>Low</td>
<td>N</td>
<td>Medium</td>
<td>N</td>
<td>Stand-alone</td>
</tr>
<tr>
<td>EAAACK [76]</td>
<td>N</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>FADE [77]</td>
<td>Y</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>N</td>
<td>Centralized</td>
</tr>
<tr>
<td>CRS [78]</td>
<td>Y</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Y</td>
<td>Distributed</td>
</tr>
<tr>
<td>SlyDog</td>
<td>Y</td>
<td>Low</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Stand-alone</td>
</tr>
<tr>
<td>LazyDog</td>
<td>N</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Y</td>
<td>Distributed</td>
</tr>
</tbody>
</table>
network. A single malicious node can selectively cooperate with a legitimate node or an infrastructure network component.

- **Computation overhead**: To detect forwarding behavior, either a sender, receiver, or intermediate nodes may record the history of communication activities that occurred in their adjacent nodes for a certain period. They can share and cross-check the history to detect any forwarding misbehavior. To realize this, each resource-constrained node is required a certain level of computing power to process.

- **Communication overhead**: In EH Nets, whenever a node is harvest state, it broadcasts a *State* packet. The original intention of this periodic packet is for one-hop neighbor nodes not to mistakenly send a packet to the energy harvesting node. Depending on the detection purposes, one or multiple nodes have to generate and send a series of control packets to the packer sender or designated nodes, resulting in a communication overhead in such a resource-limited network.

- **Detection latency**: We consider how quickly each node suspects its one-hop neighbor nodes, detect their forwarding misbehavior, and isolate them from the network by excluding their participation in the communication activities in the network.

- **Punishment**: We also investigate whether there is a mechanism to discourage single or multiple malicious nodes for their forwarding misbehavior. For example, if a node suspects one of its neighbor nodes, it reduces the forwarding probability to the neighbor node. As more forwarding misbehaviors are suspected, the neighbor node gradually loses a chance to receive a packet to forward in the network. This is the same effect of network isolation. Legitimate nodes will not involve a malicious node for communication.

- **Architecture**: In this perspective, we consider three types of network operation in conducting detection strategies, *Centralized*, *Distributed*, or *Stand-alone* [58]. In *Centralized*, a set of designated nodes conduct most major detection operations, but the rest of the nodes have relatively simple operations, such as monitoring communication activities, and report any event or abnormality to the designated nodes. In *Distributed*, every node has an equal responsibility to monitor and detect forwarding misbehavior in the network. Nodes frequently exchange control packets or the history of communication activities for detection. *Stand-alone* is the same as *Distributed*, but each node does not exchange or share any information with others.

We also analyze major ideas and operations of the detection strategies. A centralized detection system (CDS) [51] is proposed to detect packet-dropping attacks in clustered IoT networks. The basic idea is that an uplink packet drop probability of IoT devices is calculated to monitor the behavior of the gateway, which is associated with IoT devices. A detection rule is provided by conducting a generalized likelihood ratio test, in which attack probabilities are approximated based on the maximum likelihood estimation. In a heuristic-based detection (HED) scheme [53,54], a suppression attack is discouraged in a multicast protocol for low power and lossy networks (LLNs). A malicious node multicasts a series of spoof
data packets with continuous sequence numbers to prevent legitimate nodes from accepting valid data packets in the network. In the HCD [52], each node monitors its adjacent nodes’ forwarding misbehavior by tracing a limited amount of forwarding history in EHNets. Each node also gradually reduces the forwarding probability of suspected malicious nodes to exclude them from participating in the routing operation. A monitor-based approach (CMD) [55] is to mitigate forwarding misbehavior in LLNs. Each node monitors the preferred parent node to observe its packet loss rate, compares this rate with the collected packet loss rates from one-hop neighbor nodes, and detects forwarding misbehavior.

In [56] and its extended approach, a proposed checkpoint-based multi-hop acknowledgment scheme (CHEMAS) [57] randomly selects checkpoint nodes to monitor ongoing forwarding operations and replies an Ack packet to the original packet sender in WSNs. Each intermediate node located along the forwarding path counts the number of received Ack packets corresponding to the number of Data transmissions. If the node receives the less number of Ack packets, it may suspect the next located neighbor node for forwarding misbehavior, e.g., dropping either Data or Ack packet. Then, the node generates an Alarm packet and transmits it to the original packet sender for reporting a malicious node that is potentially involved in the forwarding operation. In the CHEMAS, intermediate nodes often receive and forward many Ack and Alarm packets, resulting in high battery energy consumption. To efficiently detect a selective forwarding attack, a single checkpoint-based countermeasure (SCAD) [58] is proposed in resource-constrained WSNs. Unlike the CHEMAS, a single checkpoint node is randomly selected in the network and detect forwarding misbehavior. This approach can be combined with the timeout technique and hop-by-hop retransmission operation to mitigate the unexpected packet losses that are primarily caused by forwarding attacks or fluctuating channel qualities. To discourage a malicious node dropping data packet, an acknowledgment-based punishment and stimulation scheme (APS) [59] is proposed in MANETs. In the APS, each node estimates the reputation of neighbor nodes based on routing reliability and shares its recommendation to identify a malicious node.

To detect both selective forwarding and blackhole attacks, a cooperative bait detection scheme (CBDS) [60] is proposed based on the dynamic source routing (DSR) in MANETs. In the CBDS, a source node virtually creates a destination address to monitor the reaction of a potential malicious node. Since the malicious node does not know whether the destination address is real, it may reply a fake route reply (RREP) packet to the source. Upon receiving the RREP packet, the source can trace back the route and identify the malicious node. Based on the ad hoc on-demand distance vector routing (AODV), a sequence number-based bait detection scheme (SNBDS) [61] is proposed in MANETs. A series of sequence numbers piggybacked in the RREP packet is used to see if there is any packet drop over the transmission. Each node examines whether there is any gap between sequence numbers in the receiving packets. The next-hop neighbor node may be suspected of forwarding misbehavior if the gap is greater than a predefined threshold value. A CAM scheme [62] is deployed in EHNets, and its operational summary is presented in Section 12.4.2.
An accurate and cognitive intrusion detection system (ACIDS) [81] is proposed to defend against blackhole attacks in MANETs. In the ACIDS, each node monitors key parameters (e.g., destination sequence number and route reply) and checks the amount of deviation from the normal to detect an intruder. An attack detection framework [82] is proposed to defend IoT cyber-attacks using a deep learning technique, in which an attack detector is implemented and embedded into fog nodes. In [83], a dependence estimator-based scheme is proposed in IoT sensor networks, where a deep analysis of network traffic is conducted. This scheme can identify key IoT network traffic parameters and help in detecting any malicious network activity and traffic accurately.

12.6 Discussion and future research directions

We envision that energy harvesting-motivated computing and networking under security awareness are essential to support future IoT networks. To see the full potential of research introduced in this chapter, we discuss promising research issues and directions with the interdisciplinary points of view.

**Vibration sensitive medium access control:** In vibration-motivated energy harvesting, a disturbance event initiates the direct piezoelectric effect actively or passively. A passive event can be caused by surrounding environmental resources (e.g., ground disturbance or wind) in a static EHNet, where the nodes located nearby the event sense and transform it into mechanical vibration energy for communication. Since multiple nodes can respond to the same event, they may initiate the transmission simultaneously that may result in packet contention, collision, and retransmission. On the other hand, an active event can be caused by immediate environmental resources (e.g., the kinetic motion of walking or running) in a mobile EHNet, where each node responds to the event. Note that each node must maximize the utilization of harvested energy for communication.

The prior energy harvesting-aware MAC protocols have concentrated on solar-[22,84] or thermal-based [85] energy harvesting. However, there is plenty of space to extend by deploying vibration-motivated energy harvesting from intermittent kinetic movements and their integration with the IEEE 802.11 MAC protocol. This research approach newly considers underlying properties of ambient vibrations and practical obstacles in terms of the medium access technique that will significantly affect the design of algorithms and communication protocols embedded in upper layers, such as the network and application layers.

**Energy harvesting-motivated lower power and lossy networks:** IoT-based networks equipped with smart sensors and objects are expected to play an important role in building a future communication paradigm, such as minimizing or without human intervention for communication activities [86]. In the realm of IoT, IPv6-based LLNs consisting of a myriad of resource-constrained devices endowed with the capabilities of sensing, computing, and wireless communicating represent a key enabler for IoT applications. To overcome limited battery power, energy harvesting-motivated LLNs (EH-LLNs) are rapidly emerging and will be a major part of IoT-based
networks, where energy harvesting-motivated nodes use a routing protocol for LLNs (RPL) [87]. Since the RPL was not originally designed for the energy harvesting features, we plan to develop an energy harvesting module to seamlessly integrate with RPL and conduct different simulation scenarios by using Contiki Cooja network simulation [88]. In addition, we plan to investigate the dissipation of harvested energy and traffic load, and design a traffic load and energy balancing RPL to further extend the network lifetime and improve the network performance.

**Authentication with lightweight cryptography:** We can find the presented scheme to apply to IoT device authentication, especially collaborative authentication on a group of IoT nodes using threshold cryptography [89–91]. The proposed countermeasure will allow us to further filter out the honest behaving nodes or generating the honestness weights, so when it is combined with some other security measures, we can further strengthen the threshold and improve the security level of the IoT network nodes. We can imagine that this level of work may happen at a much more powerful node such as a base station which possibly possesses the entire (or at least majority of) the IoT node topology, and the honesty of each node can aid the more accurate computation of group security measures.

Another research direction related to the proposed scheme is the authentication itself of each IoT device using very lightweight cryptographic functions such as cryptographic hash functions. Traditionally hash chain was found to be useful to balance off the computational overhead and the security level [92–95], but it does not work well to handle more complicated hierarchical structures. Hierarchically structured authentication has extensively been studied [96–99], and we can use more complicated hash structures such as hash Merkle tree, multidimensional hash chains, and hash vine. By designing a lightweight hash function to sacrifice the collision attack security, we can make it work with low computing-powered IoT devices for lower security but shorter lifetime protection of broadcast communications. This is a plausible direction in the sense that the lifetime of each broadcast is very short, so each hash computation needs to be safe against collision attacks for the short period that can be achieved with the lightweight hash design.

**IoT software architecture:** IoT, as a rapidly growing field, has applications in various domains such as healthcare, automated home services, smart energy and smart grid, food and water tracking, and transportation [100,101]. “Software engineering for the IoT poses challenges in light of new applications, devices, and services” [102]. The IoT adds additional complexity to software development as its nature of distribution and inclusion of heterogeneous devices, such as sensors and actuators [102]. One of the areas of research in the IoT from the Software Engineering perspective is software architecture.

Several reference architectures have been proposed to standardize the design of IoT systems, in which some reference architectures are more generic on industry scale implementation [102] while some are more specific [103] to the resources or environment, such as cloud computing. Some research targets specific software architecture for the IoT applications in different domains, for example, [104] presents a service-oriented software architecture for a data-driven smart city utility application and [105] did a mapping study on using microservice architecture as the building blocks for IoT
systems and cloud computing solutions. The research work [106,107] have done mapping studies on exploiting software architecture models to develop IoT systems.

Although reference architectures give the software developers a general guide and the specific software architectures proposed for different domains and resources allow the developers to adapt the methodologies while developing the IoT systems similar settings, there are still scenarios in which these architectures are not applicable. The energy harvesting-based wireless sensor networks systems [108] involve the special requirements that need to be addressed in the software architectural design. The existing proposed IoT software architectures may need to be extended and expanded with the unique aspects of wireless sensor networks and energy harvesting-based computing involved. Thus, as one of the future works, we plan to exploit the software architectural styles that work as the best practice in EHNNets powered by harvesting environmental resources.

12.7 Concluding remarks

Seamlessly interconnected IoT sensors and devices have been deployed in diverse applications and networks ranging from civil to military. Since IoT nodes are powered by batteries, they should be replaced or replenished ultimately but often hard, if it is not impossible. Due to the limited battery energy, energy harvesting from immediate environmental resources would be the best candidate to efficiently replenish or significantly reduce the frequency of replacing batteries.

This chapter introduces a DoS attack that must be considered in rapidly emerging EHNNets. Depending on single or multiple malicious nodes, three sets of adversarial scenarios and their corresponding forwarding misbehaviors are observed and analyzed to find vulnerable scenarios and malicious nodes. Detection strategies of forwarding misbehavior are also compared and analyzed comprehensively in terms of six perspectives. In addition, potential future research directions for IoT and its variants are provided, including energy harvesting-aware MAC, energy harvesting-motivated LLNs, IoT lightweight authentication, and IoT software architecture.

We envision that this chapter will open many interesting research directions to pursue and enable the research community to quickly follow up the proposed energy harvesting-motivated networking research.

List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACIDS</td>
<td>Accurate and cognitive intrusion detection system</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc on-demand distance vector routing</td>
</tr>
<tr>
<td>APS</td>
<td>Acknowledgment-based punishment and stimulation scheme</td>
</tr>
<tr>
<td>CAD</td>
<td>Channel-aware detection</td>
</tr>
<tr>
<td>CAM</td>
<td>Camouflage-based active detection scheme</td>
</tr>
</tbody>
</table>

(Continues)
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBDS</td>
<td>Cooperative bait detection scheme</td>
</tr>
<tr>
<td>CDS</td>
<td>Centralized detection system</td>
</tr>
<tr>
<td>CHEMAS</td>
<td>Checkpoint-based multi-hop acknowledgment scheme</td>
</tr>
<tr>
<td>CMD</td>
<td>Monitor-based approach</td>
</tr>
<tr>
<td>CRS</td>
<td>Channel-aware reputation system</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic source routing</td>
</tr>
<tr>
<td>EAACK</td>
<td>Enhanced adaptive acknowledgment</td>
</tr>
<tr>
<td>EHNet</td>
<td>Energy harvesting-motivated network</td>
</tr>
<tr>
<td>EH-LLN</td>
<td>Energy harvesting-motivated lower power and lossy network</td>
</tr>
<tr>
<td>FADE</td>
<td>Forwarding assessment based detection</td>
</tr>
<tr>
<td>HCD</td>
<td>Hop-by-hop cooperative detection</td>
</tr>
<tr>
<td>HED</td>
<td>Heuristic-based detection</td>
</tr>
<tr>
<td>IoTSN</td>
<td>Internet-of-Things sensor network</td>
</tr>
<tr>
<td>LazyDog</td>
<td>Proposed monitor-based detection scheme</td>
</tr>
<tr>
<td>LLN</td>
<td>Low power and lossy network</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium access control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile ad hoc network</td>
</tr>
<tr>
<td>PFCB</td>
<td>Fiber composite bi-morph</td>
</tr>
<tr>
<td>SCAD</td>
<td>Single checkpoint-based countermeasure</td>
</tr>
<tr>
<td>SCM</td>
<td>Side channel monitoring</td>
</tr>
<tr>
<td>SCPC</td>
<td>Integrated self-charging power cell</td>
</tr>
<tr>
<td>SlyDog</td>
<td>Proposed inducement-based detection scheme</td>
</tr>
<tr>
<td>SNBDS</td>
<td>Sequence number based bait detection scheme</td>
</tr>
<tr>
<td>WatchDog</td>
<td>Observation-based detection scheme</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless sensor network</td>
</tr>
</tbody>
</table>

References


Symposium on Information Processing in Sensor Networks (IPSN); 2007. p. 254–263.


[95] Pinto A, and Costa RF. Hash-Chain-Based Authentication for IoT; 2016.


