Computer Networks 139 (2018) 119–135

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs

Rui Wang, Zhiyong Zhang, Zhiwei Zhang, Zhiping Jia*

School of Computer Science and Technology, Shandong University, Qingdao, China

ARTICLE INFO

Article history:
Received 14 September 2017
Revised 11 April 2018
Accepted 14 April 2018
Available online 17 April 2018

Keywords:
SDWSNs
Malicious forwarding attacks
Trust management
Trust routing
Energy efficiency

ABSTRACT

Software-Defined Networking (SDN) has been introduced to Wireless Sensor Networks (WSNs) to achieve centralized control and flexible network management. In Software-Defined Wireless Sensor Networks (SDWSNs), security and energy are two critical issues. However, few studies address these two aspects simultaneously. In this paper, we propose an Energy-efficient Trust Management and Routing Mechanism (ETMRM) for SDWSNs to handle the malicious forwarding attacks, such as selective forwarding attack and new-flow attack. In the ETMRM, we firstly extend the SensorFlow tables to realize a light-weight trust monitoring and evaluation scheme at the node level, and propose a centralized trust management scheme at the controller level to detect and isolate the malicious nodes based on the trust information collected from sensor nodes. Secondly, we present an energy-efficient report message aggregating scheme for aggregation points' selection to save energy and ensure the transmission of control traffic. Thirdly, we present a trust routing mechanism jointly considering the node's residual energy and trust level to guarantee the transmission of data traffic. The experimental results show that the proposed architecture detects and responds to the internal network attacks, such as Greyhole, Blackhole, new-flow attacks, efficiently. Compared with the related work SDN-WISE, ETMRM improves the packet delivery ratio, reduces and balances the energy consumption, prolongs the network lifetime, and suffers lower control overhead.

1. Introduction

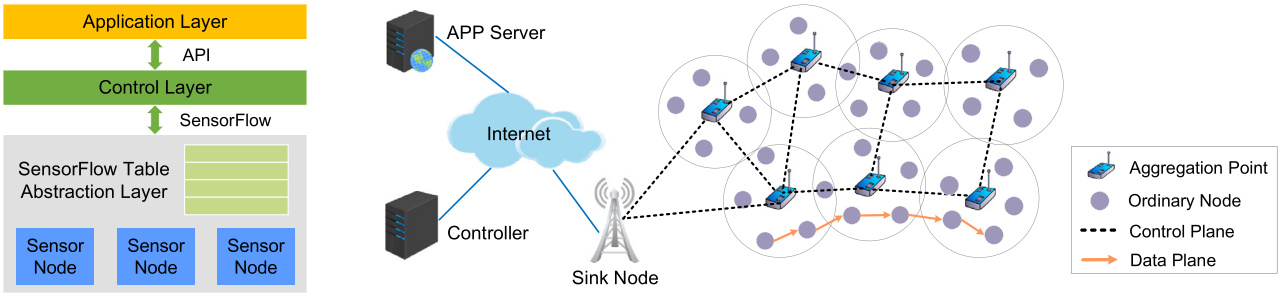
Wireless Sensor Networks (WSNs) [57] have gained wide attention from industry and academia in recent decades. WSNs have been deployed for many applications, such as remote environmental monitoring and target tracking. However, most previous researches only consider the case that a WSN is dedicated for a single sensing task. With the tightly coupled data and control planes, such an application specific WSN is prone to high deployment costs, low service reutilization, and difficult hardware recycling [15,34,60]. To tackle the above issues, researchers have introduced the concepts of Software-Defined Networking (SDN) [29] and OpenFlow [36] to the traditional WSNs, and proposed the Software-Defined Wireless Sensor Networks (SDWSNs) [16,60] and corresponding SensorFlow [34,35].

As shown in Fig. 1(a), the framework of SDWSNs has three layers [14]. The SensorFlow table abstraction layer is the underlying physical layer which consists of a set of SDWSN sensor nodes, and these nodes forward packets based on their SensorFlow tables. The control layer is separated from the forwarding layer and contains a logically centralized controller which controls the sensor nodes via the southbound interface SensorFlow. The controller also provides routing, security, and other services to the application layer via the northbound Application Program Interface (API). Based on the network status obtained from the control layer, the application layer realizes flexible management strategies, such as location services, condition services, load balancing, and topology management [15].

However, SDWSNs are not well designed yet and still face many challenges, such as the design of network operating system, standardization, security, and energy consumption [28]. In this article, we focus on the last two critical issues. For security, some security threats from the traditional WSNs are even enlarged in the SDWSNs due to the introduction of the logically centralized controller and the separation of data and control planes [19]. Taking selective forwarding attack for instance, once the malicious nodes receive packets, they may drop certain types of packets instead of forwarding all of them to the next hops [23,41]. In the SDWSNs, sensor nodes play both the roles of "hosts" and "switches" simultaneously. Therefore, if the selective forwarding nodes launch the internal attack, like dropping some control packets, the normal operation of SDWSNs will be severely affected. On the other hand, the SDWSNs inherit the issues of SDN architecture [39]. It is easy to conduct some new SDN attacks on SDWSNs due to the limited

* Corresponding author.
E-mail addresses: rw@mail.sdu.edu.cn (R. Wang), jzp@sdu.edu.cn (Z. Jia).

https://doi.org/10.1016/j.comnet.2018.04.009
1389-1286/



(a) The Framework of SDWSNs (b) A Deployment Scenario of SDWSN

Fig. 1. The network architecture of SDWSNs.

energy and capability of the deployed environments [28,39]. Newflow attack [56] is a typical representation. It leads to the controller single point failure by frequently sending new packets to the control plane. Both of the above attacks can be observed as the malicious forwarding behaviors.

Most state-of-the-art researches focus on the architecture and application optimization, the security issue in the SDWSNs is still in infancy and has not got enough attention [28]. The existing security studies usually adopt some cryptography methods to concur security threats [19], architecture [8] or services [27]. However, these cryptography based mechanisms could not work well in the resistance of the internal attacks [20]. The effective traditional WSN trust management solutions [25], which are mostly based on the coupled architecture, also could not be directly adapted into SDWSNs without considering the decoupled architecture of SDWSNs [28].

Energy constraint is another critical issue existing in the WSNs [10,54,62]. Energy consumption in the SDWSNs should also be highly optimized, since the centralized control plane needs to frequently collect the topology information to reach a global view, which may lead to high energy costs [21,54]. Nowadays, the resource utilization of SDWSNs is gaining more and more attention. Many energy saving schemes in the routing, reliability, and network lifetime have been proposed [15,53,54]. However, they mainly work for safe network environments without attacks. Few SDWSN studies consider both the **security** and **energy** problems so far.

In this article, we consider an SDWSN deployment scenario as illustrated in Fig. 1(b). The scenario consists of an application server, a controller, a sink node, and a number of SDWSN sensor nodes. The controller selects certain sensor nodes as the Aggregation Points (APs) to aggregate the report messages [13]. The APs are used to save the energy and bandwidth resource and also avoid the attackers on the control path. Based on the above architecture, we propose an Energy-efficient Trust Management and Routing Mechanism (ETMRM) for SDWSNs. We aim to efficiently collect the network topology information and achieve a centralized network management considering the existence of malicious forwarding attacks. To the best of our knowledge, ETMRM is the first to study the trust management and routing mechanism for SDWSNs by considering both the security and energy aspects. The main contributions of this paper are summarized as follows:

1. We implement a light-weight local trust monitoring and evaluation scheme based on the extended SensorFlow tables, and propose a centralized trust management mechanism based on the extended report messages. The internal attack nodes are easily isolated from the network by inserting drop rules into their neighbor nodes' access control flow tables.
2. We present an energy-efficient report message aggregating scheme to reduce the control overhead, minimize the total reporting energy consumption, and guarantee the control traffic transmission in the SDWSNs. The selection of APs is formulated as an integer nonlinear programming (INLP) problem by considering the nodes' global trust level, residual energy level, and the transmission energy consumption.
3. We propose a centralized trust routing mechanism jointly considering the global trust value and residual energy of the nodes on the path to guarantee the data traffic transmission in the SDWSNs.
4. We have implemented a prototype of our scheme, and the experimental results show that ETMRM detects and responds to the internal network attacks efficiently. The proposed scheme also improves the packet delivery ratio, reduces and balances the energy consumption, prolongs the network lifetime, and suffers lower control overhead compared with the related work SDN-WISE [16].

The remainder of this paper is organized as follows. Section 2 provides a survey of the related works. Section 3 describes two kinds of malicious forwarding attacks and presents an overview of our network security architecture. The trust management and trust routing mechanisms of ETMRM at the node and the controller level are described in Sections 4 and 5, respectively. Section 6 presents the experimental results and the corresponding analysis. Finally, Section 7 concludes this work.

2. Related work

2.1. Software-Defined Wireless Sensor Networks

Many previous studies of SDWSNs have been done on the architecture, communication protocol, and network management [6,14,16,18,38]. Luo et al. [34] and De Gante et al. [12] propose the architectures of SDWSNs with SensorFlow and show the benefits of the SDN architecture in WSNs, such as energy saving and topology discovery. Galluccio et al. [16] introduce SDN-WISE, a stateful SDN solution for WSNs. The stateful SDN-WISE reduces the amount of information exchanged between sensors and controller, and makes sensor nodes programmable. The systems proposed by Miyazaki et al. [37], Cao et al. [6] and Bera et al. [4] prove the flexibility and simplicity of network management. The SDN-enabled architectures for WSNs have been applied to various wireless scenarios, such as smart home [55], smart urban sensing [32], smart Internet of Things (IoT) [40], Industrial Wireless Sensor Networks [33], etc.

Energy efficiency is another appealing character of SDWSNs. Utilizing the dynamic configurable property of SDWSN sensor nodes, Zeng et al. [60] present an energy-efficient sensor activation strategy in multi-task SDWSNs with guaranteed quality-of-sensing. Xiang et al. [54] propose an energy favored routing algorithm for SDWSNs. They select the control nodes with higher residual energy and better location as a cluster, and then make the control

nodes to coordinate the intra-cluster nodes to complete different tasks. To relieve the high energy cost of the information collection, Huang et al. [21] design a cognitive SDWSN prototype for environmental monitoring applications. This prototype uses reinforcement learning to perform value redundancy filtering and load-balancing routing and aims to improve the energy efficiency and adaptability. However, the above works mainly benefit the periodic data collection applications.

Wang et al. [52] present an SDN based sleep scheduling algorithm to manage the energy consumption of the WSNs. Liao et al. [53] present an energy-efficient algorithm based on multi-dimensional energy space to reduce the energy cost of WSNs. Wang et al. [50] present an SDN routing in the wireless multi-hop network, which generates the shortest path based on the nodes' residual energy and global topology view. The simulation results show that this centralized routing mechanism extends the network lifetime compared with the existing algorithms (OLSR, AODV) when the traffic load reaches a certain value. However, these energy-efficient mechanisms assume that network is entirely safe, which cannot be promised in the current severe network situation.

2.2. Security in SDWSNs

Nowadays, the SDWSNs are not well designed and still face many challenges [28], especially the security problem [39]. On the one hand, the inherent character of the SDN paradigm brings many advantages to SDWSNs. The centralized security measures can be implemented on the control and application layers without the constraints of resource and can easily identify the malicious activities utilizing the global view [39]. On the other hand, some new threats will be introduced due to the decoupled architecture and the introduction of the logically centralized controller, such as the attacks on the control plane communications and forwarding devices [19]. Furthermore, considering the inherent problems of WSNs, such as computation and resource limitations, these new security threats may cause even critical problems. However, most researchers focus more on the architecture or application aspects of SDWSNs, security in SDWSNs has just received little attention [28,39].

The security issue can be reflected by studying the mechanisms designed for Software-Defined Internet of Things (SDIoT) [47]. Kalkan and Zeadally [26] present a comprehensive survey on how the SDN technology can provide security for IoT environment. They classify the literature as crypto based, network based, and traffic based solutions. The crypto based solutions focus on the cryptographic properties of the environment in order to guarantee security. Chakrabarty and Engels [8] and Choi and Kwak [9] propose the SDN based IoT security architectures that can provide authentication, access control, integrity, confidentiality, and other system security services. Although these cryptography and authentication security mechanisms can defend against the external attackers, they could not resist the internal attacks from the compromised nodes [20]. The network based solutions use cluster heads in the hierarchical structures. Gonzalez et al. [17] build a new secured network architecture based on SDN and clusters. In each domain, the cluster head can distribute routing functions and security rules to each edge controller. However, it stays at the architecture level, the performance benefit needs to be evaluated. For the traffic based solutions, the key approach is to detect and prevent the attacks based on the information collected from the traffic flows. To defend against the new-flow attack in the SDIoT, Xu et al. [56] propose a smart security mechanism with a low-cost monitoring method by reusing the asynchronous messages on the control plane. Similarly, Bull et al. [5] propose an adaptive flow based security mechanism for IoT devices using an SDN gateway. This mechanism performs dynamic analysis of the traffic patterns, and detects TCP and ICMP flood based attacks efficiently. However, the network scales of SDIoT are usually limited to one-hop which do not need to worry about the attacks from the intermediate nodes.

Especially, as a crypto based solution, Khan and Hameed [27] propose an SDN based IoT framework for provisioning security services. The trust module in the SDN controller calculates the trust value of the service requester based on the opinions received from its neighbor nodes. This scheme shows the theoretical feasibility of the centralized trust computation. However, they do not consider that the trust collection process may be damaged by the underlying attackers. Its performance evaluation also needs to be completed.

2.3. Trust schemes in SDN and WSNs

The trust and security measures in SDN, WSNs, and traditional networks have been well studied [31,43,44,58], such as intrusion detection, secure routing, and secure data. The feedback control of SDN security mechanisms usually includes three phases: collecting network statistics, detecting anomalies or intrusions in the network, and inserting flow rules to protect the network [43]. However, compared with SDN, the energy and computation resource of the SDWSN sensor nodes are limited. So the reliability of the control plane, computation and communication overhead must be taken into account when designing the security structure [19]. As an essential complement to the cryptography based security mechanisms, the trust management and trust routing schemes of the traditional WSNs can efficiently defend against the internal attackers and enhance the security, reliability, and impartiality of the system [20,25]. However, they are usually based on the coupled control and data planes [28]. Each node makes the trust decision and routes in a distributed manner [23]. Each node is the "center of its own world" [48]. This is obliviously opposite to the centralized management of SDWSNs. So not all solutions of SDN and WSNs can be combined and adapted to the SDWSN model [28,39]. In order to promise true secure SDWSNs, security must be inherently built into the SDWSN architecture [39].

In [38], some centralized network management approaches of the traditional WSNs have been discussed. One of them, Hunkeler et al. [22] propose an Intelligent, Manageable, Power-Efficient and Reliable Internet-working Architecture (IMPERIA), a centrally managed architecture for large-scale WSN. IMPERIA demonstrates that the centralized approach for the management of WSNs is an efficient alternative to the distributed protocols. To reduce the energy consumption and increase the reliability of WSNs, Sharma and Jena [45] design an energy-efficient routing strategy using the clustering and multi-path techniques. This strategy is based on the principle of reducing the load on the sensor nodes by moving more responsibility to the sink. This is similar to the concept of SDWSNs.

Tajeddine et al. [46] design CENTERA, a centralized trust based efficient routing protocol with an appropriate authentication scheme for WSNs. Each node in this system periodically sends the list of its neighbors and its packet forwarding experiences to the Base Station (BS). With the cryptography mechanism, each node on the path decrypts and verifies the signature from the source node. Then BS evaluates the trustiness of every node based on their packet forwarding ratio. CENTERA prevents replay, modification and impersonating attacks. However, as analyzed in [1], CENTERA cannot completely defend against internal attacks launched by compromised legitimate sensor nodes which already hold the key and authentication information. The nodes in CENTERA are also not intelligent enough to make decisions to isolate the misbehaving nodes. The encryption and decryption at each intermediate node also cause too much computation and energy overhead during the route discovery. Besides, CENTERA is not specially de-

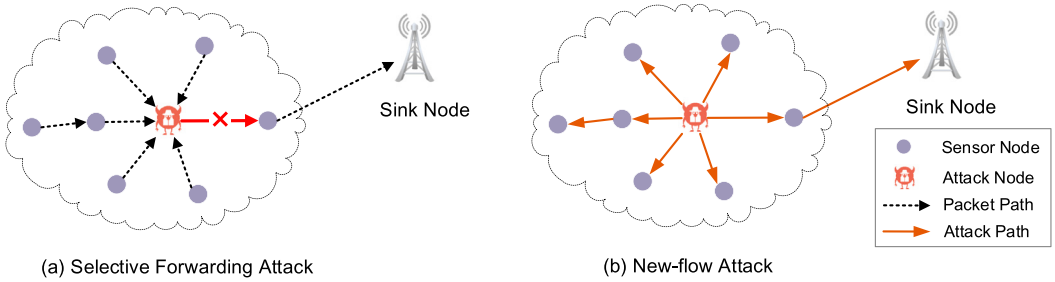


Fig. 2. Malicious forwarding attacks on SDWSNs.

signed for SDWSNs. The network flexibility is limited due to the fixed uplink and downlink paths. It fails to consider both the security [28,39] and energy efficiency [54] problems of the neighbor information collection process when there exists malicious forwarding attackers.

3. ETMRM overview

In this section, we first discuss two kinds of the malicious forwarding attacks. Then, the design goals of our proposed architecture are presented. Finally, we give an overview of the proposed ETMRM scheme.

3.1. Malicious forwarding attacks on SDWSNs

3.1.1. Selective forwarding attack

Sensor nodes have limited communication capability and low computation resource. So every node is a weak entity that can be easily compromised by the adversary [49]. As shown in Fig. 2(a), in a multi-hop wireless network, selective forwarding attack is the threat launched by the compromised node by maliciously dropping a subset of forwarding packets to deteriorate the packet delivery ratio of the network [41]. It is called as the **Greyhole** attack if the attacker drops certain types of packets, or the **Blackhole** attack if the attacker drops all packets [23,58]. Compared with WSNs, the compromised node is more harmful to the normal operation of SDWSNs. As shown in Fig. 2(a), in the centralized control SDWSNs,

- if the attacker drops the data packets it received, it would damage the transmission of data traffic.
- if the attacker drops the report packets it received, the controller could not receive the latest statuses of those nodes and verify whether they are still alive or not. Consequently, the controller will be miss-leaded to update the wrong network topology view and then make wrong routing decisions.
- if the attacker abandons the rule request or response packets, the requester would not set up the flow rules and handle the incoming packets successfully. The operation of the network will be paralyzed when a certain number of nodes are compromised.

In the traditional WSN trust mechanisms, based on the watchdog, MAC layer retransmission rate, and ACK methods [23], these selective forwarding attacks could be solved to a certain degree by eliminating distrusted nodes via the corresponding rules of trust ranking [58]. However, as discussed in Section 2.3, these trust mechanisms mainly rely on the distributed trust decisions from every single node and could not be adapted to the SDWSNs directly due to the decoupled characteristic and centralized management.

3.1.2. New-flow attack

As shown in Fig. 2(b), since the unmatched packets will be sent to the controller, the attacker could launch the new-flow attack [56] by injecting new packets to its neighbor nodes. The neighbor nodes could not distinguish the legal and the illegal traffic and then would encapsulate the new packets in *Packet – In* (rule request) format to the controller. In this way, the attack can easily fabricate much malicious traffic into the network and consume much bandwidth and energy of the whole network.

Unfortunately, compared with WSNs, the authentication and anti-replay protection schemes could not completely prevent this kind of attack in the SDWSNs. This is because the compromised node can easily launch the attack by flooding new packets with its legal identity. Besides, the control plane of SDWSNs is less robust than that of SDN. When the control link of SDWSNs is congested, the normal network operation will be destroyed and most services will be terminated.

This attack also can be seen as a malicious forwarding attack via deliberately forwarding much malicious traffic to the neighbor nodes. The corresponding measures for SDWSNs need to take both the limited capability and resource of sensor nodes into consideration.

3.2. Design goals

Our ETMRM aims to achieve the following goals simultaneously:

High security: The malicious forwarding attack nodes should be detected and isolated correctly. The routing path should be provided with high trust and reliability.

Energy efficiency: To eliminate the side effect and guarantee the safety of the periodical topology information collection, an energy-efficient report message secure aggregation scheme needs to be proposed to reduce the control overhead and protect the transmission of control traffic.

3.3. System overview

To make a clear understanding of trust, we adopt the trust definition as below [24].

Definition 1. (Trust). Trust is defined as a belief level that one sensor node puts on another node for a specific action according to the previous observation of behaviors. The **trust value** is used to reflect whether a sensor node is willing and able to act normally in the WSNs.

As shown in Fig. 3, in the ETMRM, the trust can be classified in **local trust** and **global trust**. Local trust is based on the observation of direct communication. It reflects the trust relationship between two neighbor nodes. Global trust is calculated by the centralized trust system based on the local trust information collected from the whole network. It reflects the trust assessment to each node in the view of the controller.

Generally, ETMRM exploits the network capability at both the node and the controller level. At the node level, we treat the SDWSN sensor nodes not only as a simple SDN forwarding switch but also as an individual host which has its own security capability. Each node easily monitors and records its neighbor nodes'

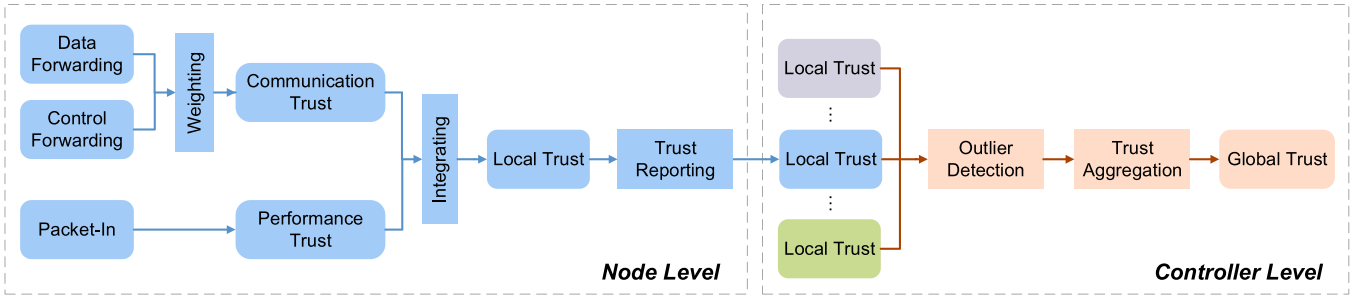


Fig. 3. The trust components of ETMRM.

forwarding and performance behaviors utilizing the extended SensorFlow tables. So based on the past successful and unsuccessful interactions, each node can get the communication trust values of its neighbor nodes via a light-weight Bayesian trust model. Then the local trust value of each neighbor node can be obtained by integrating the communication trust value and the performance trust value. To eliminate the inaccuracy in the local direct trust [24], the local trust value will be sent to the controller as a segment of network topology report message [13,16], which enables the controller to build a secure view of the network.

At the controller level, the controller first calculates each node's global trust value with outlier detection based on the collected trust information. The nodes whose global trust values smaller than the threshold will be treated as untrusted nodes (tagged as either malicious dropping attackers or new-flow attackers). They will be isolated from the network by inserting the drop rules into their neighbors' access control flow tables. To save the energy and bandwidth resource and also protect the transmission of control traffic, certain sensor nodes with large trust values and high residual energy will be selected as APs to aggregate the topology report packets. Finally, the controller could establish the trusted routing jointly considering the residual energy and trust level of sensor nodes.

4. ETMRM design: node level

In this and following sections, we will describe the trust management procedures of ETMRM at the node level and controller level, respectively. The assumptions and notations used in ETMRM are listed firstly. Then the trust monitoring, evaluation, and reporting processes are presented.

4.1. Assumptions and notations

For ETMRM architecture, we make the following assumptions:

- The network is static and is randomly deployed. Each sensor node has a unique identifier. All the sensor nodes are homogeneous.
- The sink node and controller are always trusted and also have unlimited energy and computation capability.
- The messages or flow rules from the controller can be authenticated by the cryptography technology when needed. Each node can share symmetric keys with the controller to avoid forged messages.
- The links are symmetric and the network is dense enough so that each sensor node can have several one-hop neighbor nodes within its radio range.
- Good nodes always submit the true feedbacks. There is no attack at the network initial stage.

Notations used in this section are listed in Table 1, where the columns "Notation" contains the names of variables, and their definitions are presented in the columns "Definition".

4.2. Trust recording

A topology example used in this article is displayed in Fig. 4(a). There are nine nodes in the network where node E and D are APs and node A is the attacker. The controller is located in the sink node. We will explain the main components of flow tables in the view of node E.

As shown in Fig. 5, we extend the flow tables of SDWSNs with behavior records. The Table 0 is used to achieve access control. The packets of the attackers will be dropped immediately. We split the flow tables of the SensorFlow into **data** flow table and **control** flow table in order to process the data and control traffic separately. The *Total* counter represents the total number of matched packets from the rule established time till now. We further extend the counters both in the data and control flow tables with the new *Interval* counters. For each flow entry, the *Interval* counter stands for the number of the corresponding matched packets during the Δt. The *Data_S*, *Ctrl_S*, *Data_U*, and *Ctrl_U* counters represent the numbers of successfully or unsuccessfully forwarded packets of each flow entry, respectively.

4.2.1. Communication recording

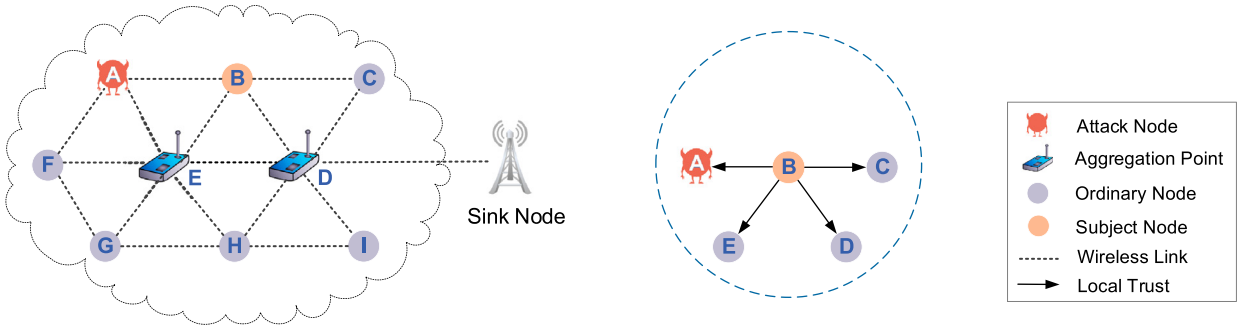
During the communication between two sensor nodes, the corresponding counters of matched flow entry will be updated when each packet of this flow is processed. If the *next_hop* neighbor node is not the destination node of the flow, the node will listen to the forwarding behavior of each packet. We take the second flow entry of data flow table (Table 1) for instance. As shown in Fig. 4 (a), when the node N_E has sent a packet of N_G to N_C via N_B, the trust system of N_E will monitor the forwarding behavior of N_B. As assumed, each node can communicate via a shared bidirectional wireless channel and operate in the promiscuous mode [30]. If N_E hears that N_B has forwarded this packet correctly, the *Data_S* will be added by 1. If N_E does not hear the retransmission of this packet within a pre-fixed time from node N_B, or without a packet-in message for this packet due to the rule miss, or if the overheard packet is illegally fabricated (by comparing the payload that is attached to the packet) [30], it will consider this interaction is unsuccessful and add the *Data_U* by 1. We can see that this achieves the forwarding recording during every Δt. At the end of each interval, the number of *Interval* counters will be added to the *Total* counters and then will be cleared to 0. The control flow table works in the same way except for some control packets types, such as *Beacon, Broadcast* whose forwarding counters are unnecessary.

4.2.2. Performance recording

For the new-flow attack, as discussed in Sections 2.2 and 2.3, those packets whose identities are forged can be detected and filtered by authentication or anti-replay protection mechanisms. However, they could not resist these malicious packets having the legal identities. So we design a packet-in recording table to defend against the new-flow attack from these compromised nodes.

Table 1
Notations used in this section.

Notation	Definition	Notation	Definition
Δt	The monitoring interval	t	The current time
LT_{ij}	The local trust value of neighbor node j in the view of node i	CT_{ij}	The communication trust value of neighbor node j
CT_{ij}^{D}	The data traffic forwarding trust value of neighbor node j	CT_{ij}^{C}	The control traffic forwarding trust value of neighbor node j
PT_{ij}	The performance trust value of neighbor node j	e	A flow entry of data or control flow table
FE_j	A flow entry set about node j	$invl_j$	The packet-in counter value of node j
F_τ	The maximum number of new flows during Δt	N	The number of sensor nodes



(a) A Topology Example of SDWSN (b) Local Trust Evaluation of Node B

Fig. 4. The network topology example and local trust evaluation.

Table 0 (Access Control Flow Table)

Matching Rule			Action
Src_Addr	Dst_Addr	Pkt_Type	Type
A	*	*	Drop
*	*	Data	Goto T1
*	*	Control	Goto T2

Table 1 (Data Flow Table)

Matching Rule			Action		Counters		
						Interval	
Src_Addr	Dst_Addr	Type	Type	Next_Hop	Total	Data_S	Data_U
F	Controller	Data	Forward	D	37	21	0
G	C	Data	Forward	B	16	10	1
B	H	Data	Forward	H	20	*	*

Table 3 (Packet-In Table)

Matching rule	Counters	
Src_Addr	Total	Interval
A	60	33
F	3	1
G	7	6

Table 2 (Control Flow Table)

Matching Rule			Action		Counters		
						Interval	
Src_Addr	Dst_Addr	Type	Type	Next_Hop	Total	Ctrl_S	Ctrl_U
*	Controller	Request	Forward	D	29	6	0
Controller	*	Broadcast	Broadcast	*	*	*	*
{F, G}	Controller	Report	Aggregation	D	68	23	0

Fig. 5. The extended flow tables with trust recording.

When the node N_E receives a new flow from its neighbor N_A and the source address of the new packet is N_A, this will lead N_E to generate a packet-in request to the controller. Then the node N_E will add the corresponding *Interval* counter in Table 3 with 1, as shown in Fig. 5. This achieves the statistics of neighbor nodes' packet-in sending behaviors. Similarly, at the end of each period Δt, this value is also added to the *Total* counter and then cleared to 0. Considering some nodes may have heavy monitoring and transmission missions, the controller could distribute the whitelists to their neighbor nodes to avoid false positives.

4.3. Local trust evaluation

As shown in Fig. 3, the **Local Trust** (LT_{ij}) consists of **Communication Trust** (CT_{ij}) and **Performance Trust** (PT_{ij}). Considering the data traffic and control traffic play different roles in the network operation, the communication trust is divided into the corresponding data and control traffic forwarding trust, as CT_{ij}^{D} and CT_{ij}^{C}. As illustrated in Fig. 4(b), each node calculates the local trust values of its neighbor nodes at the end of each Δt as follows.

4.3.1. Communication trust

To the node N_i, $CT_{ij}(\Delta t)$ reflects its neighbor N_j's forwarding honesty in the last Δt. To calculate this trust value and consider the resource-constrained SDWSN nodes, we use a popular light-weight trust management mechanism – Bayesian trust model [58]. As shown in Eq. (4.1), Bayesian theory uses the expectation $E(beta(p|x, y))$ of the beta probability destiny function $beta(p|x, y)$ to obtain the reputation score. p denotes the posteriori probabilities of binary events (x, y) and $0 \le p \le 1$. x and y represent the positive and negative interactions, respectively. This posterior reputation value $E(beta(p|x, y))$ is interpreted as the utmost possibility in the future.

$$E(beta(p|x,y)) = \frac{x+1}{(x+1)+(y+1)} \tag{4.1}$$

In the following part, we take the calculation of the control traffic forwarding trust for instance. For node N_i's neighbor node N_j, the number of the successful interactions $sc_{ij}(\Delta t)$ during each Δt can be obtained by searching the control flow table where the flows N_j needs to forward. This includes the rules whose *next_hop* is N_j but the N_j is not the destination address. We use a flow entry set $FE_j = \{e | e.next\_hop = N_j \ and \ e.dst\_addr \neq N_j\}$ to represent them. By utilizing the extended *Interval* counter, $sc_{ij}(\Delta t)$ can be gained by Eq. (4.2). The number of unsuccessful interactions $uc_{ij}(\Delta t)$ can be obtained in the same way.

$$sc_{ij}(\Delta t) = \sum_{e \in FE_j} e.interval.ctrl\_s \tag{4.2}$$

So based on the successful and unsuccessful interactions, the $CT_{ij}^{C}(\Delta t)$ of neighbor node j during the last Δt can be derived by Eq. (4.3).

$$CT_{ij}^{C}(\Delta t) = \frac{sc_{ij}(\Delta t) + 1}{(sc_{ij}(\Delta t) + 1) + (uc_{ij}(\Delta t) + 1)} \tag{4.3}$$

The $CT_{ij}^{D}(\Delta t)$ can be calculated similarly. Finally, the communication trust $CT_{ij}(\Delta t)$ during the last Δt can be obtained by Eq. (4.4), where α and β are weighted factors and $\alpha + \beta = 1$. We set α larger than β because the damage of dropping the control packets is more serious than that of data packets.

$$CT_{ij}(\Delta t) = \alpha \cdot CT_{ij}^{C}(\Delta t) + \beta \cdot CT_{ij}^{D}(\Delta t), 0 < \beta < \alpha < 1 \tag{4.4}$$

4.3.2. Performance trust

As discussed in Section 3.1, some compromised nodes may not drop the packets of their neighbors but send too many malicious packets to them (new-flow attack). So the $PT_{ij}(t)$ is used to measure this forwarding performance during the last Δt. To the node N_i's neighbor N_j, making use of the packet-in recording table as shown in Fig. 5, the performance trust $PT_{ij}(t)$ is calculated by Eq. (4.5). F_τ is the maximum number of the new flows that a node can receive from its one neighbor during a Δt. It is adaptive to the actual network environment. $invl_j$ is the value of the corresponding *Interval* counter and its default value is 1.

The more excessive sending new flows, the lower the evaluation will be. When the $invl_j$ reaches $F_\tau/2$ and N_j is not in the whitelist, the security system of node N_i will send a warning packet to its neighbor node N_j. If the node N_j still floods the new flows and $PT_{ij}(t) < 1$, then the node will send an attack report to the controller.

$$PT_{ij}(t) = \frac{1}{\left\lceil \frac{invl_j}{F_\tau} \right\rceil} \tag{4.5}$$

4.3.3. Local trust

By integrating the two kinds of forwarding trusts - communication trust and performance trust, we calculate the local trust value $LT_{ij}(t)$ by Eq. (4.6). In the existing works, the trust value can be a real number between 0 and 1 (4 bytes) or an integer between 0 and 10 (0.5 bytes). As discussed in [30], this is critical for WSNs because of the limited memory as well as transmission and reception power. To reduce the memory and transmission overhead, we set it as an unsigned integer between 0 and 100. It only needs 1 bytes of memory space and will not lose too much precision.

$$LT_{ij}(t) = \left\lceil 100 \cdot CT_{ij}(t) \cdot PT_{ij}(t) \right\rceil \tag{4.6}$$

$LT_{ij}(t)$ reflects the trust evaluation on the node N_j from node N_i's opinion and can be used to predict the node N_j's behavior in the future. By repeating these calculations, N_i can get the local trust values of all its neighbor nodes. Other nodes can also execute in the same manner to obtain their own evaluations.

Byte	Bit 0-7	Bit 0-7
0	Packet Length	Network ID
2	Source Address	
4	Destination Address	
6	Packet Type	Time To Live
8	Next Hop Address	
10	No. Hop	Battery Level
12	Congestion Level	***N***
14	*Neighbor Address*$_1$	
16	***Local Trust Value***$_1$	*RSSI*$_1$
18	*Neighbor Address*$_2$	
20	***Local Trust Value***$_2$	*RSSI*$_2$
...	.....	

Fig. 6. New report message.

4.4. Trust reporting

4.4.1. New report message

The local trust is calculated based on the observations of the forwarding behaviors. If there is not enough direct communication between two nodes, the local trust could not be calculated normally. Besides, if it is affected by the wireless interference or malicious attacks, it may have a deviation to the real value [24]. In the traditional WSNs, some reputation based trust management models take the recommendations from other neighbor nodes into the trust evaluation [3,20]. Although these ways may correct the local trust to a degree, they will be easily influenced by the surrounding malicious bad-mouthing nodes. Furthermore, collecting and managing many recommendations need extra energy consumption.

To improve the resource efficiency and reduce the overhead of information exchange, we cancel the feedback between each node as proposed in [30]. From the point of a centralized management policy, these local trust values will be reported to the controller to get more accurate global trust values. As shown in Fig. 6, we add fields for neighbor trust information in the SDN-WISE report message [13,16] sent by the node to the controller. This minor modification to the report message will avoid introducing additional transmission overhead.

4.4.2. Simple reporting

In the centralized control of SDWSNs, the controller faces a critical challenge of efficiently collecting the report messages from the network with the existence of malicious forwarding attacks. A simple way is to make all the nodes broadcast the report messages to the whole network. We call it **Simple Broadcasting**.

Another way is proposed in the SDN-WISE [16] which uses a network Topology Discovery (TD) protocol. The controller periodically broadcasts a TD packet via the sink node. This packet contains the controller's identity. Its current distance to the sink is initialized to 0. When a node N_i receives a TD packet from its neighbor node N_j with a close distance to the sink, it will update N_j as the next hop node towards the controller and broadcast this TD packet with its information. Completing the topology discovery, every node can report its current list of neighbors to the controller via its next hop. We name this way as **Simple Reporting**. The list of neighbors of SDN-WISE is periodically cleared. So this procedure needs to be executed periodically.

4.5. Overhead analysis

Through the above processes of trust recording, trust evaluation, and trust reporting, the sensor node completes the trust ob-

Table 2
Notations used in this section.

Notation	Definition	Notation	Definition
$M(t)$	The local trust matrix at current time t	G	The network graph
T_i	The current global trust value of node i	S_i	The local trust value set to node i
$E_{ON}(i)$	The energy consumption of ordinary node i	$E_{AP}(i)$	The energy consumption of AP node i
$L(i)$	The bit length of a report message	E_{tx}	The energy consumption for sending 1-bit data
E_{rx}	The energy consumption for receiving 1-bit data	E_{da}	The energy consumption for aggregating 1-bit data
λ	The compression ratio of the AP node	$E_{\tau_1}(i)$	The residual energy threshold for AP node i at time t
$Nb(i)$	The current neighbor set of node i	Adj_{ij}	The edge adjacency between node i and node j
T_τ	The trust threshold	$E_{Avg}(i)$	The average residual energy of node i's neighbor nodes at time t
$E_i(t)$	The residual energy of node i at time t	$E_{\tau_2}(i)$	The residual energy threshold for path node i at time t
$C_{SD}(P)$	The routing cost of path P from node N_S to N_D	H_{ij}	One-hop combined metric function

servation, calculation of its neighbor nodes, and finally reports the trust information to the controller.

In the trust recording process, the data and control flow tables may require extra memory space to store the extended counters. As shown in Fig. 5, it only needs two additional *Interval* counters (*Data_S*, *Data_U*) or (*Ctrl_S*, *Ctrl_U*) for each data or control flow entry. In view of the low-rate network environment of SDWSNs, each *Interval* counter could be at least 8 bits (1 bytes). Thus, taking into account that the general flow tables have at least $O(n \times 20)$ bytes [16] where n is the number of flow entries, this additional $O(n \times 2)$ bytes will not consume too much memory.

In the trust evaluation procedure, it does not need to do the complex mathematical calculations. So it suits the resource and capability limited SDWSN environment. However, in the trust reporting stage, these reporting ways face two main problems as follows.

1. **High Energy Consumption.** It's obvious that the simple broadcasting way will lead to a broadcast storm at the reporting stage. During each topology discovery period, the simple reporting scheme also has to broadcast to the entire network. To deliver the report messages, it will consume average $O(N \times h)$ transmission times where h is the average hops to the controller. Their communication overhead and energy cost are quite high [21,48,54].
2. **High Packet Loss Ratio.** These two ways could not guarantee that all the report messages and other control traffic will be arrived at the controller when there exists malicious forwarding attacks. They may work well at the beginning or when the compromised nodes are few. However, with the increment of attackers, the transmission procedures become uncontrollable and the delivery ratio of network traffic will drop sharply.

So to overcome the above problems, we will propose a solution to reduce the communication and energy cost and also guarantee the delivery of control traffic.

5. ETMRM design: controller level

In this section, we will describe the trust management and trust routing procedures of ETMRM at the controller level. We first list the notations used in this section. Then we introduce a centralized trust evaluation model and propose an energy-efficient report message aggregating mechanism to solve the above-mentioned two critical issues. Finally, we design a trust isolating scheme and a trust routing scheme.

5.1. Notations

Notations used in this section are listed in Table 2.

5.2. Centralized trust evaluation

At the beginning of a network, there is no attacker or the attackers are few. Thus, the controller can collect the network topology information with local trust values through the simple broadcasting or reporting mechanisms at the end of every Δt. The controller builds the network graph $G = (V, E)$ where V is the set of the nodes, E is the set of edges, and $|V|$ is the number of sensor nodes. Then the local trust matrix $M(t)$ could be constructed as follows.

$$M(t) = \begin{bmatrix} LT_{11} & LT_{12} & LT_{13} & \cdots & LT_{1n} \\ LT_{21} & LT_{22} & LT_{23} & \cdots & LT_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ LT_{n1} & LT_{n2} & LT_{n3} & \cdots & LT_{nn} \end{bmatrix} \tag{5.1}$$

LT_{ij} is the local trust value from node N_i to its neighbor node N_j, where $i, j \in [1, n]$ and $n = |V|$. The trust value to itself LT_{ii} is discarded to maintain the justice. Searching the column vector in $M(t)$ whose $LT_{ij} > 0$, we obtain a local trust value set containing the opinions from node N_j's neighbor nodes to N_j, as $S_j = \{LT_{ij}, LT_{kj}, \cdots, LT_{mj}\}$. So in the global view, a straightforward way to evaluate node N_j is averaging the values of S_j, as shown in Eq. (5.2).

$$\overline{T_j} = \frac{\sum_{LT_{ij} \in S_j} LT_{ij}}{|S_j|} \tag{5.2}$$

However, as discussed in [41,42], this average reputation value $\overline{T_j}$ is easily affected by the promotion or demotion from the bad-mouthing attack. In order to identify and isolate these false trust opinions, many outlier detection mechanisms for WSNs have been proposed [61]. Here we adopt a general method like the one presented in [41]. Based on the original average value $\overline{T_j}$, the standard deviation σ_j of the trust set S_j is calculated by

$$\sigma_j = \sqrt{\frac{1}{|S_j|} \sum_{LT_{ij} \in S_j} (LT_{ij} - \overline{T_j})^2} \tag{5.3}$$

Compared to a standard deviation threshold σ_τ [41], if $\sigma_j \le \sigma_\tau$, we consider the trust recommendations to N_j from its neighbor nodes are consistent and we have the global trust value $T_j = \overline{T_j}$. Otherwise, the trust recommendations to node N_j are suspicious and may be combined with the bad-mouthing attack. The LT_{ij} with the largest deviation to $\overline{T_j}$ will be removed from S_j and the new average trust value $\overline{T_j'}$ is updated until $\sigma_j \le \sigma_\tau$. Then, we have $T_j = \overline{T_j'}$. Considering the powerful storage and computing capability, the global trust value T_j in the controller is a real number between [0,1].

If T_j is smaller than a certain trust threshold T_τ, N_j will be judged as a malicious node. If the controller receives many new flow requests sourced from N_j and certain accusations from its neighbor nodes, it will be a new-flow attacker. Otherwise, it will be a selective packet forwarding attacker. No matter what kinds of attacks, it will be treated as an untrusted node and not be considered in the normal network management.

Table 3
Simulation parameter used in this section.

Type	Parameter	Value	Parameter	Value
Network	Network size	400 m × 400 m	Sink node	{200 m, 200 m}
	N	100	Initial energy	1.5 J
Application	λ	0.5	Δt	30 s
	data payload length	240 bits	rule timeout	50 s
Energy consumption model	E_{tx}	0.0010875 mJ/bit	E_{rx}	0.0009 mJ/bit
	Standby power	0.708 mJ/s	E_{da}	5 nJ/bit

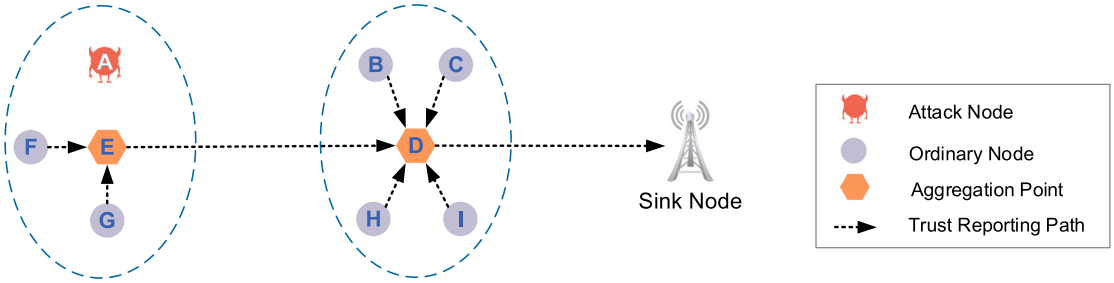


Fig. 7. An example of trust aggregation.

5.3. Trust aggregation

As described in Section 4.5, the simple broadcasting and reporting approaches may result in critical energy consumption and high packet loss ratio problems. Also, the network traffic will become uncontrollable with the increment of attackers. To protect the transmission of the control traffic, balance the energy consumption, and also prolong the network lifetime, we present an Energy-efficient Report Message Aggregating Scheme (ERMAS) for SDWSNs.

5.3.1. Aggregation problem formulation

AP Selection Problem: *Given a network with the existence of the malicious forwarding nodes, find a subset of nodes that act as aggregation points with the objective of minimizing the reporting energy consumption and guarantee the reporting delivery simultaneously.*

To make it clear, we present an example of ERMAS based on the network topology displayed in Fig. 4. As shown in Fig. 7, the homogeneous sensor nodes are divided into the **ordinary nodes** and the **APs**, respectively. The ordinary nodes have the same communication radius. During the trust reporting phase, the ordinary nodes need to send their neighbor report messages together with trust information to their APs. The APs are the sensor nodes that have high residual energy and trust level. Similar to the APs designed for SDN based smart grid network [51], the APs are also used to assist the information exchanging between ordinary nodes and the controller. But different from the usual clusters in WSNs [30], the APs do not need to allocate the tasks or provide the trust feedbacks to their members in the domains.

The APs will do the aggregation of their neighbor nodes' report messages firstly and then send them to the controller. To estimate the energy consumption of the communication between two nodes, free space wireless radio model (d^2 power loss), multi-path fading channel model (d^4 power loss) [54], real data-sheets of sensor nodes or other energy consumption models all can be adopted in ERMAS. So during the reporting stage, the energy consumption of an ordinary node $E_{ON}(i)$ can be calculated by Eq. (5.4). $L(i)$ is obtained by a fixed length of report packet header L_f added with the length of its neighbor list. The length of the neighbor list is calculated by one neighbor information length L_n multiplying its current neighbor number $|Nb(i)|$.

$$E_{ON}(i) = L(i) \cdot E_{tx} \tag{5.4}$$

$$L(i) = L_f + L_n \cdot |Nb(i)|$$

For an AP, in an aggregation procedure, its energy consumption $E_{AP}(i)$ is combined with the reception and transmission energy, as shown in Eq. (5.5).

$$E_{AP}(i) = E_{RX}(i) + E_{TX}(i) \tag{5.5}$$

Its reception energy consumption $E_{RX}(i)$ is approximately related to the length of all its neighbor nodes' report messages $L_{Nb}(i)$, as calculated by Eq. (5.6),

$$E_{RX}(i) = L_{Nb}(i) \cdot E_{rx} \tag{5.6}$$

$$L_{Nb}(i) = \sum_{j \in Nb(i)} L(j)$$

Aggregating and reporting its neighbor nodes' report messages with its report message can reduce the size of the packets and the times of forwarding. The aggregated data packet length and data aggregation energy consumption are computed by Eqs. (5.7) and (5.8), respectively. Then the aggregated report packets will be sent to the sink via $d_{sink}(i)$ times transmission and $d_{sink}(i) - 1$ times reception through the relay APs.

$$L_{Agg}(i) = \lambda \cdot (L_{Nb}(i) + L(i)) \tag{5.7}$$

$$E_{DA}(i) = E_{da} \cdot (L_{Nb}(i) + L(i)) \tag{5.8}$$

$$E_{TX}(i) = E_{DA}(i) + L_{Agg}(i) \cdot ((d_{sink}(i) - 1) \cdot E_{rx} + d_{sink}(i) \cdot E_{tx}) \tag{5.9}$$

As shown in Eq. (5.9), the total transmission energy consumption $E_{TX}(i)$ can be calculated by the addition of the data aggregation energy cost of AP_i and the transmission energy consumption of APs on the path to the sink.

Finally, the optimization goal is as shown in Eq. (5.10). We aim to find a suitable number of high-quality APs to minimize the reporting energy consumption of the whole network. The energy consumption of APs varies at different positions. We set the threshold $E_{\tau_1}(i) = \tau_1 \cdot E_{Avg}(i)$ which is related to the average residual energy level of its neighbor nodes. It is adaptively adjusted with the growth of time due to the consumption of node energy. And taking security into consideration, the AP should have not only higher residual energy but also larger global trust value. Subsequently, the last constraint means that all the normal nodes are

either in the AP set, or adjacent to some neighbor nodes in the AP set. This relationship is used to ensure that all the ordinary nodes (except the compromised node) could be covered by the APs.

$$\min \sum_{i=1}^{|N|} v_i \cdot E_{AP}(i) + \sum_{i=1}^{|N|} |v_i - 1| \cdot E_{ON}(i)$$

$$\text{s.t.} \begin{cases} E_{AP_i}(t) \geq E_{\tau_1}(i) \\ T_{AP_i}(t) \geq T_\tau \\ \prod_{i=1}^{N} (v_i + \sum_{j=1}^{N} v_j \cdot Adj_{ji}) \geq 1 \end{cases} \tag{5.10}$$

where

$$v_i = \begin{cases} 1, & \text{Selected as the AP;} \\ 0, & \text{Ordinary Node.} \end{cases}$$

It is difficult to directly solve the above integer nonlinear programming (INLP) problem when the network is large-scale. A candidate AP set can reduce the searching range of feasible nodes. At the current time t, the node N_i is a candidate AP if it has higher residual energy and larger global trust value than the average level. Furthermore, considering the unlimited computing capability of the controller, many heuristic algorithms can be used to solve the above problems. For example, to investigate the task scheduling problem in the Software-Defined Embedded System, Zeng et al. [59] formulate the task completion time minimization problem with consideration of task scheduling and image placement as a mixed-integer nonlinear programming (MINLP) problem. A three-stage heuristic algorithm is proposed to tackle this complex issue. More solutions can be found in [2].

5.3.2. Aggregation Point responsibilities

At the set-up stage, the controller will notice the selected APs using the source routing scheme. Then the APs broadcast their elected announcements which are signed by the controller. When the ordinary node N_i receives multiple broadcast announcements and is covered by several APs, it will select the AP with the highest residual energy and becomes its AP member. Then N_i sends a join-in message to its selected AP. This AP will set up the aggregation related flow rules. Other control messages, like rule requests, the ordinary nodes can also forward to their APs to aggregate and forward them.

Because the attacker has not invaded the network and there is no compromised node at the beginning, as long as the ERMAS selects the trusted APs, the delivery of report messages and other control traffic would be guaranteed. Indeed, there is a possibility that the attacker will compromise the AP. This will lead to the loss of the control traffic. In this situation, the compromised AP will be quickly detected by its neighbor nodes by utilizing the local trust evaluation mechanism. Each neighbor node will do the following steps:

- if it directly connects to the sink or is covered by multiple APs, it will deliver the trust report to the sink or those APs.
- if it has the next hop to the controller, it will deliver the trust report to the next hop node.
- if it has no way to the controller, it just broadcasts the trust report.
- if it has a way to the controller and receives trust reports from other nodes, it will help them forward the reports.

By utilizing the received report information, the controller will replace this malicious AP. Additional Shadow Aggregation Points (SAPs) can also be assigned to each AP domain to monitor the input and output traffic of the elected AP.

Furthermore, compared with the ordinary nodes, the APs take more responsibilities of aggregating the report messages and forwarding the rule request and response packets, and thus the energy consumption of APs will be more than that of the ordinary nodes. Here we discuss two ways to avoid the excessive consumption of APs' energy. First, the controller continuously monitors the residual energy of the APs based on their report messages. When an AP's residual energy is less than a certain degree of the average residual energy of its neighbor nodes, the controller will do a local AP rotation utilizing the current graph G and ERMAS. The controller replaces the corresponding APs partially. The other way is that the controller does a global AP rotation when it observes that certain APs have low residual energy.

5.4. Trust isolating

Different from the distributed trust mechanisms, the controller takes centralized responses to the untrusted sensor nodes which are detected by the global trust evaluation. If the malicious node is an ordinary node, the controller needs to send out drop rules like $< src\_addr =$'j', $action =$'$Drop$'$>$ to its neighbor nodes. However, if the controller just sends a broadcast to the whole network or multiple unicast to every affected node, it will waste much energy and fail to guarantee the transmission.

We propose a minimum AP broadcast mechanism to reduce the energy consumption. Taking an untrusted node N_j for example, we create its victim neighbor set $VN(N_j)$ which are all normal nodes and each AP's neighbor set $Nb(AP_i)$ utilizing the graph G and the global trust evaluation. The AP broadcast set $B(AP_i)$ will be selected with minimum $|B(AP_i)|$ members to cover all the victim neighbor nodes as follows.

$$VN(N_j) \subset \bigcup_{i \in B(AP_i)} Nb(AP_i) \tag{5.11}$$

These control messages will be signed by the controller and sent to the APs in the set $B(AP_i)$. The APs in the set will broadcast the signed messages and then these victim neighbor nodes will insert the drop rules into their access control flow tables, as shown in Fig. 5. They will also clear the rules in the data and control flow tables whose $next\_hop =$'j' to avoid forwarding packets to the attacker. Once if the attacker is an AP node, the controller just does an extra local AP rotation which is similar to ERMAS. If the above solutions fail, the controller will do a whole network broadcast as the worst case.

5.5. Trust routing

When a source node N_S wants to communicate with a destination node N_D, but it does not know the path to node N_D, or it does not know how to handle a new legal flow to N_D from its neighbor node, it will send a rule request message to the controller via its AP. When the controller receives the rule request, it first checks whether the requester is a potential new-flow attacker and then decides to execute the secure routing algorithm or not.

The high trusted node will guarantee the delivery of the packets. And the node with higher residual energy will have better ability to forward the packets. So in this work, we consider the trust values and residual energy of the nodes on the path to design a secure and reliable routing mechanism. For a candidate routing path P from N_S to N_D, its path cost $C_{SD}(P)$ is calculated by Eqs. (5.12) and (5.13), where H_{ij} is a one-hop combined metric function from path node N_i to N_j, and $<\theta_1, \theta_2, \theta_3>$ are the coefficients to scale the impact of each factor. We can see that the greater trust value and higher residual energy of N_j are, the smaller H_{ij} will be, and the higher the possibility where node N_j will be chosen as the forwarding node is.

$$C_{SD}(P) = \sum_{i,j \in P} H_{ij} \tag{5.12}$$

$$H_{ij} = \theta_1 + \theta_2 \cdot \left(1 - \frac{E_j(t)}{E_{init}}\right) + \theta_3 \cdot (1 - T_j(t)) \tag{5.13}$$

As shown in Eq. (5.14), the problem of routing determination is formulated as selecting a path with the minimum routing cost. Because the node on the key position may run out its energy quickly, the node should satisfy the energy constraint to guarantee the reliability of the routing path. We set the threshold $E_{\tau_2}(j) = \tau_2 \cdot E_{Avg}(j)$ where τ_2 can be adaptively adjusted to the location. And the nodes whose global trust values below the threshold T_τ will be ignored.

$$\min C_{SD}(P), P \in \Omega_{N_S \to N_D} \tag{5.14}$$
$$\text{s.t.} \begin{cases} E_j(t) \geq E_{\tau_2}(j) \\ T_S \geq T_\tau \\ T_D \geq T_\tau \\ T_j \geq T_\tau \end{cases}$$

To solve this problem, as shown in Algorithm 1, we first get the reduced network graph G' by removing the nodes that do not meet the above constraints. The weight of each link is calculated subsequently. Finally, the Dijkstra algorithm is used to find the minimum routing cost path in the G'. The construction of the routing path can be built by delivering the control messages to the source node N_S based on the previously selected APs. The control messages can simply contain the next hop node like OpenFlow does in SDN. Besides, like the source routing mechanism, it can also contain complete forwarding nodes to reduce the number of requests. Here, we adopt the former one to follow the principles of OpenFlow. The latter one will be also used when the network security situation becomes worse.

Algorithm 1: Trust routing.

Input: $G = (V, E), N_S, N_D, \tau_2, T_\tau, E_{init}, \theta$
Output: Minimum Cost Routing Path: $P_{N_S \to N_D}$

$P = \emptyset$;
if $T_S < T_\tau$ *or* $T_D < T_\tau$ **then**
  return P;
end
for *nodes* $N_i \in V$ **do**
  if $T_i < T_\tau$ *or* $E_i(t) < E_{\tau_2}(i)$ **then**
    remove N_i from V;
    for *node* $N_j \in Nb(i)$ **do**
      remove $Link_{ij}$ from E;
      remove $Link_{ji}$ from E;
    end
  end
end
get reduced graph $G' = (V', E')$;
for *node* $N_i \in V'$ **do**
  for *node* $N_j \in Nb'(i)$ **do**
    $H_{ij} = \theta_1 + \theta_2 \cdot (1 - \frac{E_j(t)}{E_{init}}) + \theta_3 \cdot (1 - T_j)$;
  end
end
get reduced graph $G' = (V', E', H)$;
P=Dijkstra(G', N_S, N_D);
return P;

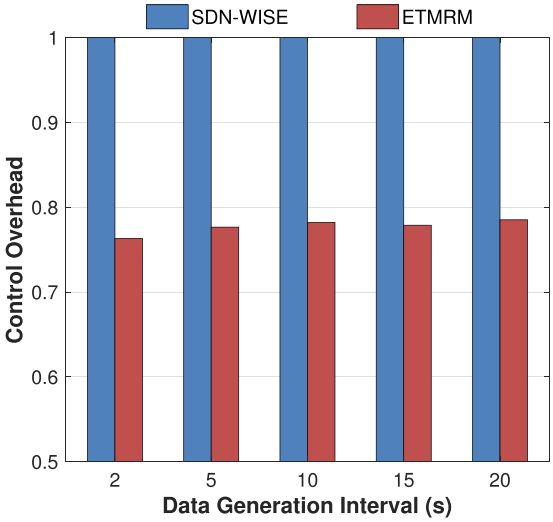


Fig. 8. Normalized control overhead ratio.

6. Experiment

6.1. Experimental setup

We implement a prototype system of the ETMRM based on the SDN-WISE project [7]. Each SDWSN node utilizes a light-weight Bayesian based trusted model. The data plane is implemented with a modified Cooja platform proposed in Contiki 2.7 [11], which is an open source operating system both for IoT and WSNs, and provides communication interfaces with the controller. The controller runs in a desktop computer which is equipped with an Intel(R) Core(TM) i7-6770HQ CPU and 32GB DDR4 memory. The centralized trust evaluation, isolating, routing, and aggregation mechanisms of ETMRM are implemented as new models in the controller.

We conduct several simulation experiments to test the performance of our solution under a large-scale network. In order to simulate the actual energy consumption of WSN devices, we measured the energy consumption of Texas Instruments CC2530 with a 1 dBm output power using the YOKOGAWA WT310 digital power meter. The measured results together with other experimental parameters are presented in Table 3.

We randomly deploy 100 sensor nodes with the same initial energy in the network. The nodes send data packets to other nodes in the same network randomly. We set the data packet generation interval between 2 and 25 s to evaluate the performance of the ETMRM and SDN-WISE under different workloads. We use a pseudo-random mechanism to generate the same destination addresses for both schemes. Firstly, we compare the performance of the ETMRM with the SDN-WISE in terms of control overhead, network lifetime, and energy distribution in the normal network environment. Furthermore, with different attacker numbers of selective forwarding attack and new-flow attack, we compare the ETMRM with the SDN-WISE according to the packet delivery ratio, detection ratio, energy consumption, etc.

6.2. Control overhead

The control overhead is defined by the ratio of the control packets to all the packets in the network. The normalized control overhead ratios of the two schemes are shown in Fig. 8. With different data generation intervals, we see that the ETMRM reduces the control overhead by 22.3% on average compared with SDN-WISE. In other words, with the same number of data packets, the ETMRM generates fewer control packets than SDN-WISE. This is due to the

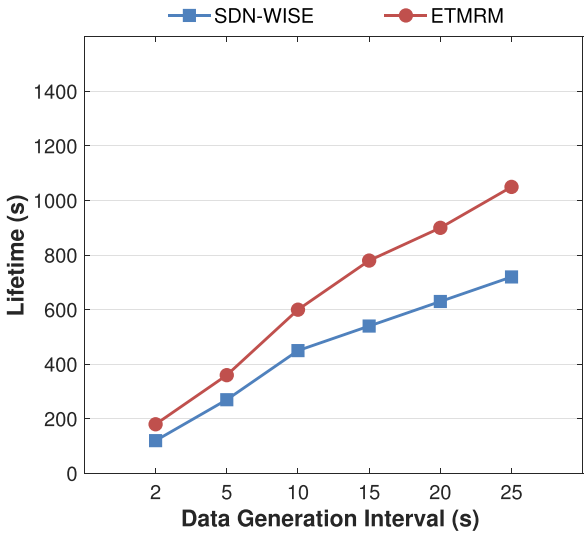


Fig. 9. Networklifetime.

fact that the SDN-WISE needs to separately deliver every report, request or response packet of each node through the whole network. On the other hand, the ETMRM aggregates the periodical report messages and other control packets in the network and delivers them through the efficient AP paths. Thus, the number of control packet forwarding in the network is reduced. The less control overhead also leads to less control energy consumption in the transmission process. This also benefits the network lifetime.

6.3. Network lifetime

The sensor node which exhausts its energy will be dead. The lifetime of the network is defined as the first node death time. Fig. 9 shows the network lifetimes of the two schemes under different data traffics. The data transmission, reception, aggregation, and standby power consumption are based on the parameters listed in Table 3. It can be observed that when the data interval is small, both schemes have a small network lifetime. This is because the frequently sending data packets will also lead to a lot of control traffic from the intermediate nodes. The nodes at the key points will drain their energy quickly. However, with the increment of data interval, we can see that the survival time of ETMRM is becoming better than SDN-WISE. This is because the ERMAS of ETMRM greatly reduces the control overhead to save the transmission consumption. The AP rotation mechanism balances the energy consumption of APs. Besides, taking advantages of the global view, the controller makes the energy-aware routing bypassing the low residual energy nodes at the critical positions to decrease their overhead and save their energy. So compared with SDN-WISE, ETMRM prolongs the network lifetime by 33.3% up to 50%.

6.4. Energy distribution

Fig. 10 shows the 3D maps of the residual energy distributions of two schemes when the first node dies. As shown in Fig. 10(a), there is a large energy hole near the sink node where the nodes have the lowest residual energy. This is because the sensor nodes of SDN-WISE need to send their periodical report messages and other control packets through the control path to the controller. The nodes near the controller are forced to forward these packets, which consumes up their energy quickly. Whereas, in the ETMRM, the energy consumption is more balanced than the SDN-WISE, as shown in Fig. 10(b). The reason is that the higher residual energy APs will bear these forwarding pressures and reduce the energy consumption by aggregating the control traffic. Besides, the monitoring mechanism of the controller timely adjusts these APs whose residual energy is low. And the energy-aware routing scheme also reduces the forwarding tasks on the low residual energy nodes near the sink to save their energy. Consequently, the energy hole is revealed to a degree.

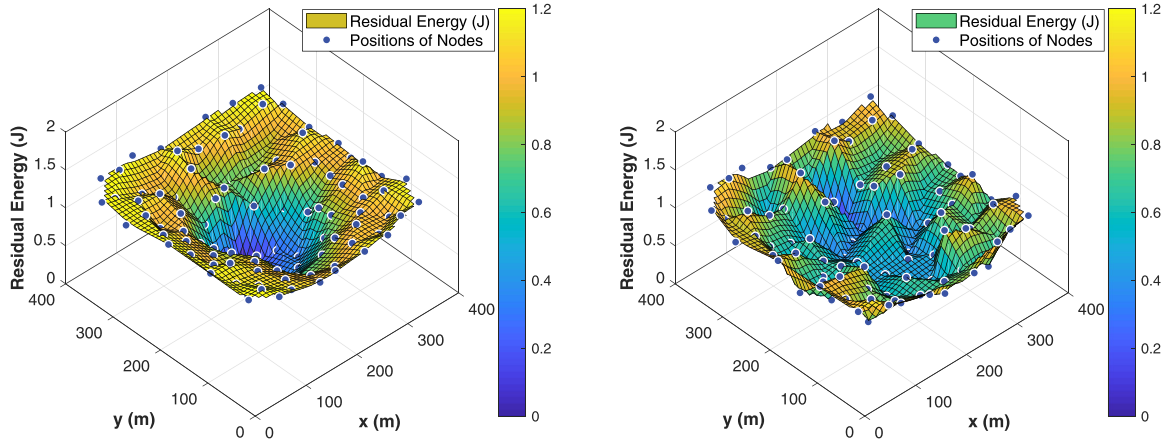
6.5. Selective forwarding attack

To compare the resistance abilities to malicious attacks of the ETMRM and the SDN-WISE, we do a set of experiments under different kinds of attacks and workload situations. The experimental parameters related to the attacks and trust mechanism are presented in Table 4.

The simulation time is set to 300 s and the reporting period is set as 30 s. We analyze the performance of these two schemes between 100 s and 200 s. The initial energy of each sensor node is reset to 5 J and the attack nodes are randomly selected.

6.5.1. Selective forwarding attack on data traffic

Firstly, we test the selective forwarding attack on the data traffic where the attacker will drop the data packets it receives. This

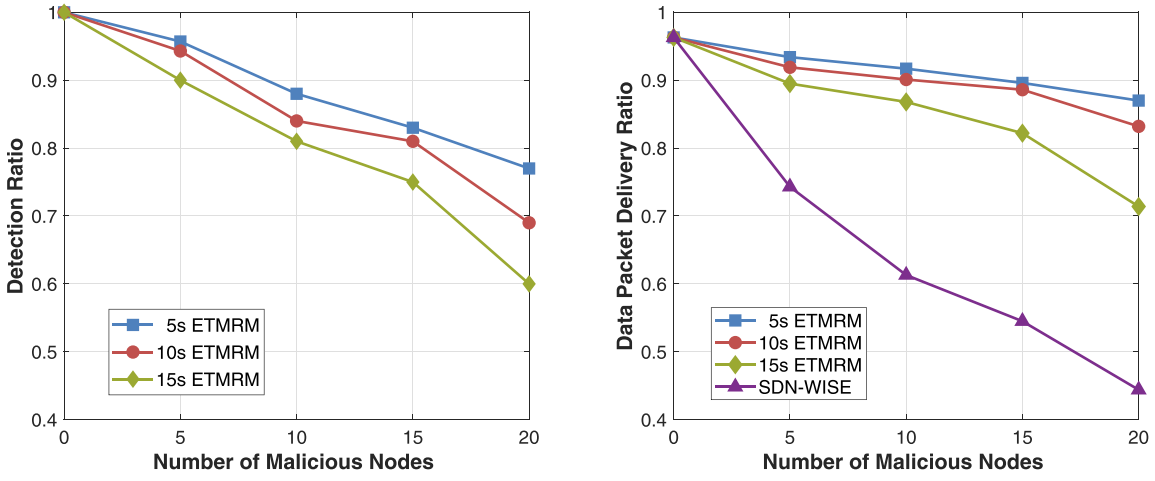


(a) SDN-WISE (b) ETMRM

Fig. 10. Energy distributions of two schemes.

Table 4
Simulation parameter used in this section.

Type	Parameter	Value	Parameter	Value
Network	Simulate time	300 s	Δt	30 s
Attack model	Number of attacker nodes	4–20	Attack time	100 s–200 s
	Greyhole attack on data	Drop data packets	Greyhole attack on control	Drop control packets
	Blackhole attack	Drop all packets	New-flow attack	Flood new packets
Trust mechanism	α	0.6	β	0.4
	τ_1	0.75	T_τ	0.7
	$\theta_1, \theta_2, \theta_3$	1,1,1	E_{init}	5J



(a) Malicious Node Detection Ratio (b) Data Packet Delivery Ratio

Fig. 11. Performance of two schemes under attacks on data traffic.

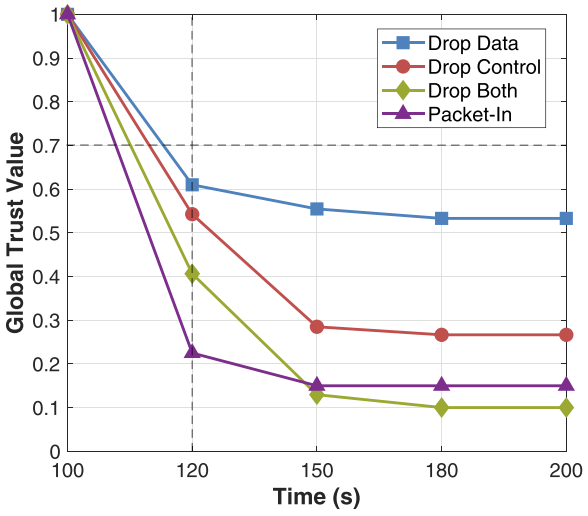


Fig. 12. Trustvalue curve under attacks.

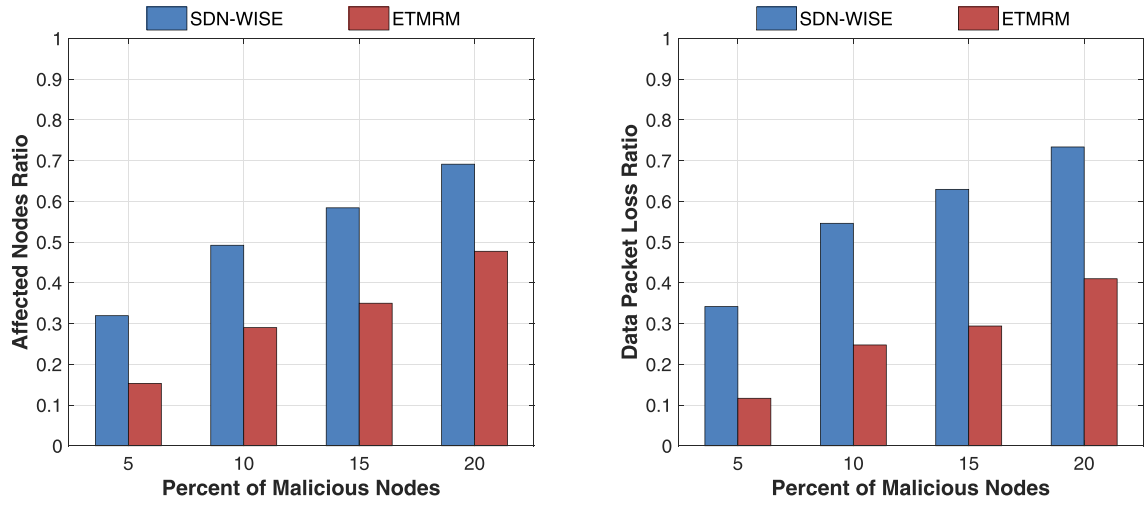
attack can be called as the Greyhole attack on the data traffic [23]. As shown in Fig. 11(a), with the increment of the malicious nodes, the average detection ratios of ETMRM decrease gradually. This is because that it will be more difficult for the controller to detect a large number of attackers with polluted reports. However, when the data generation interval becomes smaller, the detection ratios increase gradually. This is because the smaller data generation interval is, the more the interactions between two nodes are. So the node will have more chance to monitor and evaluate whether its neighbor node is an attacker. When the data generation interval is 10 s, Fig. 12 shows the trust value curves of a selective forwarding attacker at each reporting time without the proposed trust isolating mechanism. We see that the global trust value of the Greyhole attacker on data traffic will fall below the T_τ quickly. Overall, as long as the nodes have frequent communications, ETMRM can have the detection ratio higher than 77% when the malicious attackers are less than 20%. Due to the lack of security mechanism, SDN-WISE does not detect the malicious nodes.

As shown in Fig. 11(b), the average packet delivery ratio of SDN-WISE decreases to 44% sharply with the increment of the malicious nodes. However, ETMRM guarantees at least 71.4% packet delivery ratio with 20 attacker nodes when the data generation interval is 15s. This is because that the detected malicious nodes will be quickly isolated from the network in the previous rounds of reporting periods. Utilizing the minimum AP broadcasting mechanism, their neighbor nodes will receive the drop rules and will not forward packets to them anymore.

6.5.2. Selective forwarding attack on control traffic

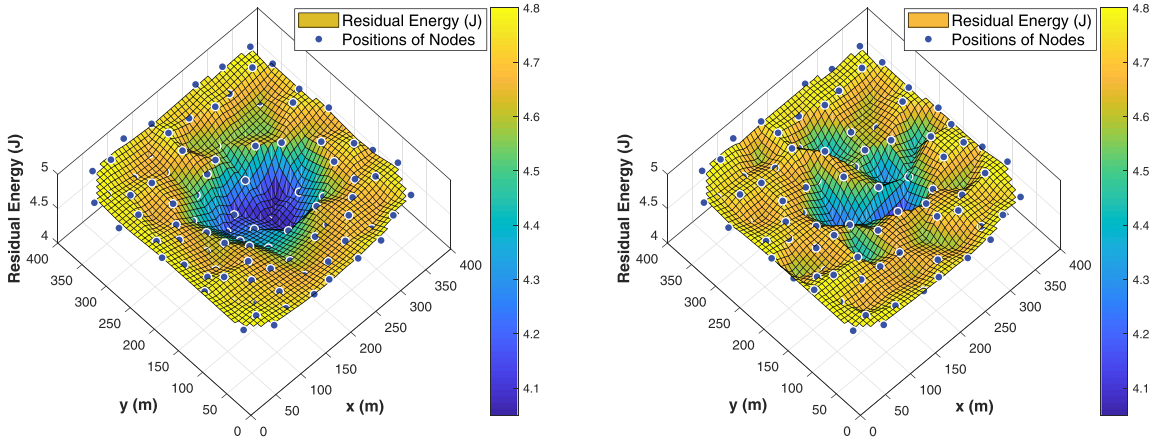
Different from SDN-WISE, the control traffic in the ETMRM is mainly forwarded by the APs. The number of APs in a sensor network with 100 nodes is between 15 and 20. To make a better comparison, we use the percentage between 5% and 20% to indicate the number of control traffic attackers.

Dropping the control packets will affect the normal topology collection and rule set-up operation of the network. As shown in Fig. 13(a), with the increment of the percentage of attackers, the ratios of the nodes whose report packets are deliberately dropped also grow rapidly. With 20% malicious nodes in the SDN-WISE, nearly 70% of the report packets will be lost. This is because the attackers can easily attract the control traffic by falsely claiming that they have the higher residual energy level and have closer distances to the sink at the topology discovery stage. The controller could not get the underlying node statuses and the routing decision will be affected. As shown in Fig. 13(b), this attack also leads to the growth of the data packet loss ratios. This is because the



(a) Node Report Loss Ratio (b) Data Packet Loss Ratio

Fig. 13. Performance of two schemes under attacks on control traffic.



(a) SDN-WISE (b) ETMRM

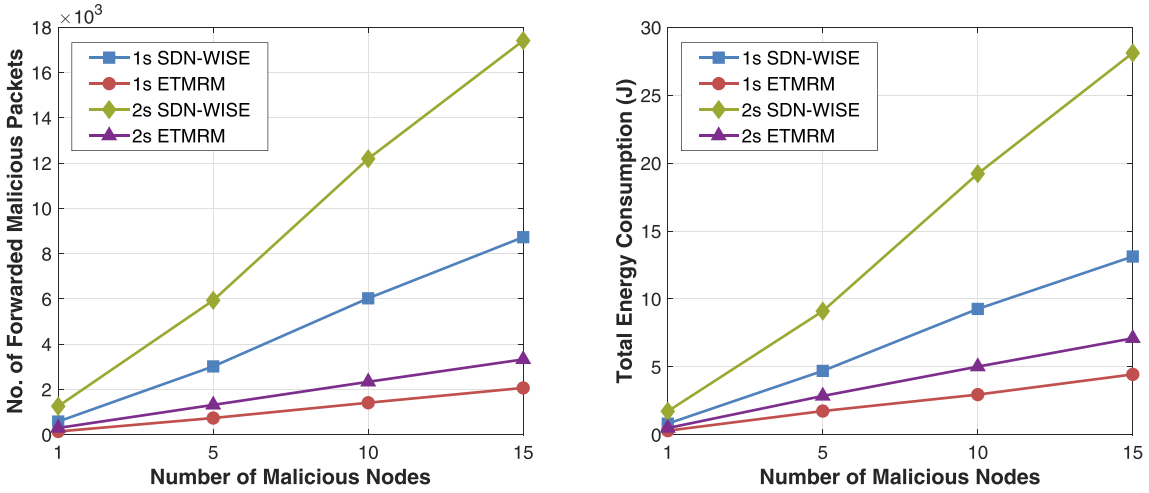
Fig. 14. Energy distributions of two schemes under new-flow attacks.

attackers not only drop the report packets but also drop the rule request and response packets. In the SDN-WISE, 73.4% of the data packet transmission will be interrupted when the percentage of attackers is equal to 20%. Compared with Fig. 11(b), we see that in the SDWSNs, the attack on the control traffic will make more serious damage to the network than that of the data traffic. In the ETMRM, as shown in Fig. 13(a) and (b), 47.7% of the report packets and 41.1% of the data packets will be affected when 20% of the APs are compromised. However, this just happens at the first report period (100 s – 120 s) as the reporting period Δt is set 30 s. Although the AP takes charge of the forwarding of control traffic, once it is judged as the malicious node by the local trust model, its members and neighbor APs will deliver the report packets through other alternative ways to the controller. As shown in Fig. 12, its global trust value in the controller will fall below T_τ quickly at the reporting time point 120 s. Furthermore, if the AP becomes a Blackhole attacker and drops both the data and control traffic simultaneously, its global trust value will rapidly decrease to 0.41 at the time 120 s. So the controller can utilize the global view and received trust evaluation reports to detect the misbehavior AP and replace it with new high trusted APs.

6.6. New-flow attack

We first test one new-flow attacker situation which locates at the position {270 m, 230 m} and attacks from 100 s to 200 s at the attack rate 2 packet/s. The normal data generation interval is set to 10s and $F_\tau = 10$. As shown in Fig. 14(a), the new-flow attacker also makes an energy hole around it where its neighbor nodes' residual energy is much lower than the average level. This is because the new-flow attacker will produce excessive new flows to its neighbor nodes and lead to frequent packet-in requests to the controller. So compared with the selective forwarding attacks, this kind of attack will incur more critical energy consumption damage to the network. However, the energy hole phenomenon can be relieved to a degree in our scheme, as shown in Fig. 14(b). In the ETMRM, once this attacker is detected by its neighbor nodes, they will report to the controller. The global trust value of this new-flow attacker drops to 0.23 sharply at the time 120 s when the controller receives certain accusations from its neighbor nodes, as displayed in Fig. 12. Then it will be isolated from the network immediately.

Furthermore, as shown in Fig. 15(a) and (b), we can see that the numbers of the forwarded malicious packets and the total energy consumption of the whole network increase rapidly with the



(a) The Number of Forwarded Malicious Packets (b) Total Energy Consumption

Fig. 15. Performance of two schemes under new-flow attacks.

increment of the new-flow attackers and attack rates in the SDN-WISE. However, as discussed above, the attackers will be quickly isolated from the network in the ETMRM. So the numbers of the forwarded malicious packets are much smaller than those in the SDN-WISE. The total wasted energy in the ETMRM is also less than that in the SDN-WISE, as shown in Fig. 15(b).

7. Conclusion

SDWSNs face both security and energy efficiency problems. In this paper, to concur the malicious forwarding behaviors - selective forwarding attack and new-flow attack, we build an energy-efficient trust management and routing mechanism for SDWSNs. We first extend the flow tables of the sensor nodes to achieve a light-weight trust monitoring and evaluation scheme at the node level. Utilizing the collected information, a centralized trust management mechanism at the controller level is proposed to detect and isolate the malicious nodes. Then an energy-efficient report message aggregating scheme is designed to reduce the energy consumption and guarantee the transmission of the control traffic. Based on the route requirement and the global security status, we propose a secure routing mechanism jointly considering the node's residual energy and trust value. Finally, the experimental results show the proposed architecture effectively detects and responds to the malicious forwarding attacks. Compared with the related work SDN-WISE, the proposed mechanism improves the packet delivery ratio, reduces and balances the energy consumption, prolongs the network lifetime, and suffers lower control overhead.

Acknowledgments

This work was supported by the State Key Program of National Natural Science Foundation of China No. 61533011; National Key R&D Program of China No. 2017YFB0902600.

References

[1] A. Ahmed, K.A. Bakar, M.I. Channa, A.W. Khan, K. Haseeb, Energy-aware and secure routing with trust for disaster response wireless sensor network, Peer–to-Peer Netw.Appl. 10 (1) (2017) 216–237.
[2] J.N. Al-Karaki, R. Ul-Mustafa, A.E. Kamal, Data aggregation in wireless sensor networks-exact and approximate algorithms, in: 2004 Workshop on High Performance Switching and Routing, IEEE, 2004, pp. 241–245.
[3] F. Bao, R. Chen, M. Chang, J.-H. Cho, Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, IEEE Trans. Netw. Serv. Manage. 9 (2) (2012) 169–183.
[4] S. Bera, S. Misra, S.K. Roy, M.S. Obaidat, Soft-WSN: software-defined WSN management system for IoT applications, IEEE Syst. J. PP (99) (2016) 1–8.
[5] P. Bull, R. Austin, E. Popov, M. Sharma, R. Watson, Flow based security for IoT devices using an SDN gateway, in: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, 2016, pp. 157–163.
[6] C. Cao, L. Luo, Y. Gao, W. Dong, C. Chen, TinySDM: software defined measurement in wireless sensor networks, in: Proceedings of the 15th International Conference on Information Processing in Sensor Networks, IEEE Press, 2016, p. 18.
[7] University of Catania, SDN-WISE, (http://sdn-wise.dieei.unict.it/).
[8] S. Chakrabarty, D.W. Engels, A secure IoT architecture for smart cities, in: 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2016, pp. 812–813.
[9] S. Choi, J. Kwak, Enhanced SDIoT security framework models, Int. J. Distrib. Sens. Netw. 12 (5) (2016) 4807804.
[10] Y. Choi, Y. Choi, Y.-G. Hong, Study on coupling of software-defined networking and wireless sensor networks, in: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2016, pp. 900–902.
[11] C. Community, Contiki, (http://www.contiki-os.org/).
[12] A. De Gante, M. Aslan, A. Matrawy, Smart wireless sensor network management based on software-defined networking, in: 2014 27th Biennial Symposium on Communications (QBSC), IEEE, 2014, pp. 71–75.
[13] P. Di Dio, S. Faraci, L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, P. Livreri, Exploiting state information to support QoS in software-defined WSNs, in: 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), IEEE, 2016, pp. 1–7.
[14] W. Dong, G.-L. Chen, C.-H. Cao, L.-Y. Luo, Y. Gao, Towards a software-defined architecture for wireless sensor networks, Chin. J.Comput. 40 (8) (2017) 1779–1797.
[15] Y. Duan, W. Li, X. Fu, Y. Luo, L. Yang, A methodology for reliability of WSN based on software defined network in adaptive industrial environment, IEEE/CAA J. Autom. Sin. 5 (1) (2018) 74–82.
[16] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks, in: 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 513–521.
[17] C. Gonzalez, O. Flauzac, F. Nolot, A. Jara, A novel distributed SDN-secured architecture for the IoT, in: 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2016, pp. 244–249.
[18] I.T. Haque, N. Abu-Ghazaleh, Wireless software defined networking: a survey and taxonomy, IEEE Commun. Surv. Tutor. 18 (4) (2016) 2713–2737.
[19] D. He, S. Chan, M. Guizani, Securing software defined wireless networks, IEEE Commun. Mag. 54 (1) (2016) 20–25.
[20] Z. Hu, Y. Bie, H. Zhao, Trusted tree-based trust management scheme for secure routing in wireless sensor networks, Int. J. Distrib. Sens. Netw. 11 (12) (2015) 385849.
[21] R. Huang, X. Chu, J. Zhang, Y.H. Hu, Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: a prototype, Int. J. Distrib. Sens. Netw. 2015 (2015) 1. 2015-10-4
[22] U. Hunkeler, C. Lombriser, H.L. Truong, B. Weiss, A case for centrally controlled wireless sensor networks, Comput. Netw. 57 (6) (2013) 1425–1442.
[23] F. Ishmanov, Y. Bin Zikria, Trust mechanisms to secure routing in wireless sensor networks: current state of the research and open research issues, J. Sens. 2017 (2017).
[24] J. Jiang, G. Han, F. Wang, L. Shu, M. Guizani, An efficient distributed trust model for wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 26 (5) (2015) 1228–1237.

[25] J. Jiang, G. Han, C. Zhu, S. Chan, J.J. Rodrigues, A trust cloud model for underwater wireless sensor networks, IEEE Commun. Mag. 55 (3) (2017) 110–116.
[26] K. Kalkan, S. Zeadally, Securing internet of things (IoT) with software defined networking (SDN), IEEE Commun. Mag. PP (99) (2017) 1–7.
[27] F.I. Khan, S. Hameed, Software defined security service provisioning framework for internet of things, Int. J. Adv. Comput. Sci. Appl. 7 (12) (2016) 411–425.
[28] H.I. Kobo, A.M. Abu-Mahfouz, G.P. Hancke, A survey on software-defined wireless sensor networks: challenges and design requirements, IEEE Access 5 (2017) 1872–1899.
[29] D. Kreutz, F.M.V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2014) 10–13.
[30] X. Li, F. Zhou, J. Du, LDTS: A lightweight and dependable trust system for clustered wireless sensor networks, IEEE Trans. Inf. Foren. Secur. 8 (6) (2013) 924–935.
[31] F. Liu, L. Wang, H. Johnson, H. Zhao, Analysis of network trust dynamics based on the evolutionary game, Scientia Iranica Trans. E, Ind. Eng. 22 (6) (2015) 2548.
[32] J. Liu, Y. Li, M. Chen, W. Dong, Software-Defined internet of things for smart urban sensing, Commun. Mag. IEEE 53 (9) (2015) 55–63.
[33] S. Luo, H. Wang, J. Wu, J. Li, Improving energy efficiency in industrial wireless sensor networks using SDN and NFV, in: IEEE Vehicular Technology Conference, 2016, pp. 1–5.
[34] T. Luo, H.-P. Tan, T.Q. Quek, Sensor openflow: enabling software-defined wireless sensor networks, IEEE Commun. Lett. 16 (11) (2012) 1896–1899.
[35] A. Mahmud, R. Rahmani, Exploitation of OpenFlow in wireless sensor networks, in: 2011 International Conference on Computer Science and Network Technology (ICCSNT), 1, IEEE, 2011, pp. 594–600.
[36] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.
[37] T. Miyazaki, S. Yamaguchi, K. Kobayashi, J. Kitamichi, S. Guo, T. Tsukahara, T. Hayashi, A software defined wireless sensor network, in: 2014 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2014, pp. 847–852.
[38] D. O'Shea, V. Cionca, D. Pesch, The presidium of wireless sensor networks - asoftware defined wireless sensor networkarchitecture, in: International Conference on Mobile Networks and Management, Springer, 2015, pp. 281–292.
[39] S.W. Pritchard, G.P. Hancke, A.M. Abu-Mahfouz, Security in software-defined wireless sensor networks: threats, challenges and potential solutions, in: 15th International Conference on Industrial Informatics (INDIN'2017), 2017.
[40] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, N. Venkatasubramanian, A software-defined networking architecture for the internet of things, in: 2014 IEEE Network Operations and Management Symposium (NOMS), IEEE, 2014, pp. 1–9.
[41] J. Ren, Y. Zhang, K. Zhang, X. Shen, Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks, IEEE Trans. Wireless Commun. 15 (5) (2016) 3718–3731.
[42] Y. Ren, V.I. Zadorozhny, V.A. Oleshchuk, F.Y. Li, A novel approach to trust management in unattended wireless sensor networks, IEEE Trans. Mob. Comput. 13 (7) (2014) 1409–1423.
[43] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Commun. Surv. Tutor. 18 (1) (2016) 623–654.
[44] G. Shang, P. Zhe, X. Bin, H. Aiqun, R. Kui, FloodDefender: protecting data and control plane resources under SDN-aimed DoS attacks, in: INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, IEEE, 2017, pp. 1–9.
[45] S. Sharma, S.K. Jena, Cluster based multipath routing protocol for wireless sensor networks, ACM SIGCOMM Comput. Commun. Rev. 45 (2) (2015) 14–20.
[46] A. Tajeddine, A. Kayssi, A. Chehab, I. Elhajj, W. Itani, CENTERA: a centralized trust-based efficient routing protocol with authentication for wireless sensor networks, Sensors 15 (2) (2015) 3299–3333.
[47] S.K. Tayyaba, M.A. Shah, O.A. Khan, A.W. Ahmed, Software defined network (SDN) based internet of things (IoT): a road ahead, in: the International Conference, 2017, pp. 1–8.
[48] G. Theodorakopoulos, J.S. Baras, On trust models and trust evaluation metrics for ad hoc networks, IEEE J. Sel. Areas Commun. 24 (2) (2006) 318–328.
[49] M. Tiwari, K.V. Arya, R. Choudhari, K.S. Choudhary, Designing intrusion detection to detect black hole and selective forwarding attack in WSN based on local information, in: International Conference on Computer Sciences and Convergence Information Technology, 2009, pp. 824–828.
[50] J. Wang, Y. Miao, P. Zhou, M.S. Hossain, S.M.M. Rahman, A software defined network routing in wireless multihop network, J. Netw. Comput. Appl. 85 (2017) 76–83.
[51] S. Wang, X. Huang, Aggregation points planning for software-defined network based smart grid communications, in: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.
[52] Y. Wang, H. Chen, X. Wu, L. Shu, An energy-efficient SDN based sleep scheduling algorithm for WSNs, J. Netw. Comput. Appl. 59 (C) (2016) 39–45.
[53] L. Wenxing, W. Muqing, W. Yuewei, Energy-efficient algorithm based on multi-dimensional energy space for software-defined wireless sensor networks, in: 2016 International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2016, pp. 309–314.
[54] W. Xiang, N. Wang, Y. Zhou, An energy-efficient routing algorithm for software-Defined wireless sensor networks, IEEE Sens. J. 16 (20) (2016) 7393–7400.
[55] K. Xu, X. Wang, W. Wei, H. Song, B. Mao, Toward software defined smart home, IEEE Commun. Mag. 54 (5) (2016) 116–122.
[56] T. Xu, D. Gao, P. Dong, H. Zhang, C.H. Foh, H.-C. Chao, Defending against new-flow attack in SDN-based internet of things, IEEE Access 5 (2017) 3431–3443.
[57] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Comput. Netw. 52 (12) (2008) 2292–2330.
[58] Y. Yu, K. Li, W. Zhou, P. Li, Trust mechanisms in wireless sensor networks: attack analysis and countermeasures, J. Netw. Comput. Appl. 35 (3) (2012) 867–880.
[59] D. Zeng, L. Gu, S. Guo, Z. Cheng, S. Yu, Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system, IEEE Trans. Comput. 65 (12) (2016) 3702–3712.
[60] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, Y. Xiang, Energy minimization in multi-task software-defined sensor networks, IEEE Trans. Comput. 64 (11) (2015) 3128–3139.
[61] Y.-Y. Zhang, H.-C. Chao, M. Chen, L. Shu, C.-H. Park, M.-S. Park, Outlier detection and countermeasure for hierarchical wireless sensor networks, IET Inf. Secur. 4 (4) (2010) 361–373.
[62] F. Li, Y. Yang, Z. Chi, L. Zhao, Y. Yang, J. Luo, Trinity: enabling self-sustaining WSNs indoors with energy-free sensing and networking, ACM Trans. Embed. Comput. Syst. (TECS) 17 (2) (2018) 57.

Rui Wang received the B.E. degree in the School of Software Engineering at Shandong University, in 2009. Now he is pursuing the Ph.D. degree in the School of Computer Science and Technology, Shandong University, from 2013 to now. His main research interests include embedded systems, trust computing, Software-Defined Networking and Software-Defined Wireless Sensor Networks.

Zhiyong Zhang received the Ph.D. and M.E. degree in the School of Computer Science and Technology at Shandong University, in 2017 and 2013, respectively. Now he is working as a Postdoctoral Research Fellow with Shandong University. His main research interests include real-time and embedded systems, wireless sensor network, mobile computing, and emerging non-volatile memory.

Zhiwei Zhang received the B.E. degree at Shandong University (Weihai) in 2015 and has been studying the graduate program for the M.E. degree in the School at Computer Science and Technology at Shandong University, from 2015 to now. His main research interests include Wireless Sensor Networks, Software-Defined Networking and real-time and embedded systems.

Zhiping Jia received the Master and Ph.D. degree from the School of Computer Science and the School of Control Science, Shandong University, Jinan, China, in 1989 and 2007, respectively. From July 1989, he was with the Department of Computer Science and Technology at Shandong University. Since 2002, he has been a professor in the Department of Computer Science and technology at the Shandong University. He has published more than 70 research papers in refereed conferences and journals, and served as program committee members in numerous international conferences. He received Shandong Province Award, and Teaching Award.