LPSDN: Sink-Node Location Privacy in WSNs via SDN Approach

Yawar Abbas Bangash, Lingfang Zeng, Shijun Deng, and Dan Feng

Wuhan National Laboratory for Optoelectronics
School of Computer, Huazhong University of Science and Technology, China, Wuhan.
E-mail:yawar.parachinar2003@gmail.com, dengshijun1992@gmail.com, {lfzeng,dfeng}@hust.edu.cn

***Abstract*—In a WSN, sink-node gathers data from surrounding nodes and sends it to outside world via a gateway. Therefore, its location information is important to both attacker and network operator. The former can launch attacks on a sink-node to steal information or damage it, while the latter must hide its location to ensure data's safety, and physical protection. Providing sink-node location information anonymity in WSN against a local and global adversary with minimal energy penalties, we propose a novel technique *Location Privacy via Software Defined Networking (LPSDN)*, inspired by the concept of Software Defined Networking. LPSDN uses three kinds of nodes: a centralized controller, a special buffering and forwarding node, and a slave node. These nodes work together to hide sink-node location information against traffic analysis attack. LPSDN conserves more energy, because it does not generate fake packets to hide sink-node location information. A GUI/command-line can be used to monitor (energy status, traffic density) and control the whole network. Comparing with traditional WSNs, LPSDN efficiently reduces processing burden on slave nodes, increases overall network life, and creates on-demand traffic density zones to deceive an adversary. It also provides better maintainability, manageability, and control over all nodes and their behavior.**

***Index Terms*—Wireless Sensor Network, Location Privacy, Software Defined Networking, Energy, Latency.**

I. Introduction

Wireless Sensor Networks (WSNs) consist of a large number of unattended, low-cost, small memories, less processing capable, and distributed embedded tiny sensor nodes [1], [2]. These interconnected nodes communicate with each other through wireless media to process and report data to a sink-node for further actions. WSNs are used in every field of life: field monitoring, industrial applications, military applications, wild-life, animal rearing, health-care systems, traffic monitoring, and environmental monitoring. The extensive use of WSNs in these applications generates an enormous amount of data. This traffic volume can have hidden value—the traffic volume itself without knowing its contents. In a mission-critical system such as military, frequent communication, time-specific communication, and lack of communication can lead to a certain pattern, which an adversary can exploit to know about the military secrets: planning, short-range communication, and command change. Whitfield Diffie and Susan Landu said that "the heart of a communications intelligence organization, however, is not cryptanalysis but traffic analysis" [3]. Thus, to ensure location information privacy, a proper mechanism is needed to guard against traffic analysis attack.

Traffic analysis is the process of deducing information from the monitored traffic volume—the communication patterns—without being aware of the contents of data [3]. In most cases, traffic analysis attack is launched on a sink-node to exploit its location privacy. An attacker can use expensive radio transceivers to detect message flow [4]; whenever he sees a huge amount of traffic density, he deduces the dense communication area as a sink-node location. Thus, traffic analysis attack can exploit sink-node location information.

In WSNs, sink-node gathers data from surrounding nodes and reports to outside world via a gateway. As a result, traffic volume surrounding sink-node becomes dense compared to normal sensor nodes. Sink-node has important data about network topology and sensor nodes; it also has mission-critical and sensitive information [5]. Sink-node location information must be protected; location privacy needs more than confidentiality [2]. Confidentiality helps to encrypt a message; if an adversary captures a message, he will not be able to read it. However, confidentiality alone cannot help to guard against traffic analysis attack [6], [7]. In a battle field, an adversary even can destroy a sink-node remotely with traffic analysis attack. Research community [8], [9], [10], [11], [12], and [13] have proposed some solutions for guarding against traffic analysis attack and provided sink-node information location privacy via fake packets generation, fake hot spot generation, and randomness in the traffic density. These approaches are costly, because of substantial communication overhead, packet collision, battery power drain, and extra processing. The traditional approaches of protecting sink-node location are lacking a centralized mechanism and, thus, lack in control and manageability. WSNs are rigid to policy changes and hard to manage [14].

In WSN, the tightly coupled hardware-software standalone sensor node raises the problem of scalability, manageability, and troubleshooting. For example, a large number of distributed standalone sensor nodes need installation, upgradation, and fault-removal individually. Furthermore, open wireless media can be eavesdropped easily [7] and, thus, needs a mechanism to overcome this inherited problem. Furthermore, small communication range leads to deploy more nodes for proper communication, which finally incurs a high capital cost. Tackling these issues, future research directions are looking for a novel platform; the novel platform should address the inherited WSN resource constraint nature (processing, memory,

978-1-5090-3315-7/16/$31.00 ©2016 IEEE

open media, communication range, and energy). In addition, the novel platform should be easy to install, configure, manage, maintain, scale, and administer. Thus, we propose, a very first technique to provide sink-node location information privacy in WSNs against a local and global adversary with minimum energy penalties via Software Defined Networking (SDN) approach. The technique is called Location Privacy via SDN *(LPSDN)*.

Local traffic analysis refers to the traffic analysis of only one portion of the total deployed area, and the related adversary is called local adversary; global traffic analysis refers to the traffic analysis of the entire deployed area, and the related adversary is called global adversary [15]. Global adversary has the global view of the entire network; he can monitor any place at any time. LPSDN delivers packets with minimum delay, conserves more energy, shifts traffic volume from Base Station (BS) to other nodes, increases overall network life, creates on-demand traffic density zones in certain locations to deceive an adversary (adversary will believe multiple sink-nodes), and hides sink-node location information against traffic analysis attack without using fake packets.

SDN is a new paradigm where data planes and control planes are decoupled [16]. Data plane forwards packets, while the intelligent piece—SDN controller—manages and controls the overall routing decisions. The separation of control plane from forwarding/data plane reduces network complexity [17], and enhances monitoring, manageability and maintainability [18]. This new approach has given birth to innovations in the networking industry. From carrier grade network to data centers, from cloud computing to enterprise networks, and from storage industries to power grids, SDN is reshaping traditional networking architecture. The same way, LPSDN is an inspiration of the SDN architecture. A comprehensive general study about SDN can be found at [19], [18] while a particular study about SDN and WSN can be found at [20].

Until now, to the best of our knowledge, there is no such SDN inspired method to hide sink-node location information from a local and global adversary. Our main contributions are *advantages* provided by SDN approach, *key challenges* while implementing SDN approach to hide sink-node location, *centralized control logic* to control the whole network behavior, and *on-demand traffic reshaping with minimal energy penalties* to deceive an adversary about sink-node location. In LPSDN, Software Defined Base Station (SDBS), sink-node, BS, and controller are used interchangeably. The remainder of this paper is organized as follows. Section II discusses related work, key challenges, and requirements; Section III discusses LPSDN architecture and model, Section IV presents the simulation and experimental results, while Section V concludes the paper.

II. Related Work

A. BS Location Privacy in Traditional WSNs

In Base station Location Anonymity and Security Technique BLAST [9], authors introduced BLAST nodes and common nodes to hide BS location information from an adversary. Main disadvantages of this scheme are: maximum traffic overhead due to excessive dummy packets generation, shortest path usage in delivering a packet to BS inside BLAST ring, and the placement of BS node inside BLAST ring. In Location Privacy Routing (LPR) scheme [21], authors used real and fake packets to minimize the traffic direction from eavesdropping. Traditional single path routing is extremely vulnerable to attack; it provides merely one path for packet movement. LPR introduced path diversity by combining real and fake packets. In LPR, every incoming and outgoing packet traffic are equally distributed in all directions, which incurs huge packet overhead and collision. Furthermore, LPR packets are not always directed towards a receiver; this leads to packet loss and extra delay. Another very important technique called Differential Enforced Fractal Propagation (DEFP) [22], used multi-path routes and fake packets to create multiple arbitrary high communication areas to deceive an adversary. Adversary believes the created random communication areas as true sink-nodes. In DEFP, multiple random high communication areas introduce packet overhead, quick battery drain, packet collision, and considerable packet loss.

In [11], authors introduced Random Routing Scheme (RRS), Dummy Packet Injection (DPI), and Anonymous Communication Scheme (ACS) to deceive an adversary via random packet movement and a lot of fake packets generation. Because of excessive randomness and fake packets, this scheme suffered from communication overhead, packet collision, and extra energy consumption. In [12], authors proposed a protocol for sink location privacy via topology discovery protocol and data transmission techniques. Concealing of the Sink Location (CSL) [13] technique is used to hide BS location via injecting fake packets to deceive an adversary. Another similar approach to hide BS is proposed by [8]; they have used the concept of deceptive packets to increase the anonymity of BS—it aims to prevent the disclosure of participants (sender, receiver) in communication [5]. In all these techniques [11], [12], [13], [8] excessive packets overhead leads to shorten sensor power capacity, increase packet collision and drop rate, and finally, introduce dead nodes; these nodes sacrifice availability—an important pillar of the cryptography triangle which consists of Confidentiality, Integrity, and Availability (CIA).

B. BS Location Privacy in SDN Based WSNs

Authors in [14] have provided the very first model (they claimed) for SDN based WSN. Their proposed model comprises of three layers: sensor layer, control layer, and application layer. Another attempt is made by [23]; they have highlighted some key requirements, protocol architecture, and packet format for software defined WSNs.

In traditional WSNs, because of the limited power capacity, wireless sensor node run-out of battery power quickly. Work done by [24] is an attempt to reduce the energy consumption in SDN enabled WSN environment. Further step is forwarded by [25]; in their scheme, they have shown multiple controllers to manage the SDN based WSN. Some advantages, for example, versatility, manageability, intelligence and speed, multi-

tenancy, and flexibility are already discussed by [14], [18], but these advantages are not only limited to, as mentioned. Authors in [26] have analyzed security for SDN applications in general. Their one aspect is to secure the main controller from DDoS attack. Further study has conducted by [17]; they have provided LineSwitch solution to tackle DoS attack effectively over an SDN control plane.

C. SDN Advantages & Challenges

Using SDN paradigm in a WSN BS location privacy, we can efficiently scale the deployed network. New nodes can be added, and faulty ones can be removed. With SDN, new policy can be implemented and new applications can be built with reprogramming [27] the desired node.

The advantages and opportunities of reprogramming, on-demand traffic reshaping, and scaling incur cost and demand better IT skills. In traditional sensor nodes, the intelligent behavior and the forwarding behavior are embedded in a single device. The key design challenge for sensor node is, to behave only like forwarding device; it should not have routing-intelligence. Consequently, this will create pressure on already built hundreds of thousands of sensor nodes to be redesigned from scratch. Either industry must provide newly built sensor nodes with sensor OpenFlow protocol [14], or they must upgrade the already built nodes to support new protocol stack.

Another key challenge is the BS design itself. Is it good to provide control plane logic in a sink-node, or to embed control functionality outside BS, for example, in a PC? Another important challenge is the wireless media communication range. Either industry should go for high communication range, or it should keep to the IEEE 802.15.4 practiced standard. In addition, all sensor nodes exchange packets with BS; ultimately, BS has a tremendous amount of request packets and data packets. The challenge is, how to overcome the communication overhead incurred by sensor nodes and controller? Moreover, if a control plane receives a lot of packets and flow requests, it can reach to a state called control plane saturation [28]. In this state, SDN controller cannot handle any further request. Authors [17] [28], have provided solutions to tackle control plane saturation attack, but BS' defense against saturation attack is still unknown. Furthermore, while communicating with controller, the communication channel and sensor OpenFlow protocol [14] need to be protected by security protocols and encryption mechanism.

Another very important challenge is to hide *BS* location information from an adversary. In LPSDN, we came across the optimal solution to this challenge —providing SDBS anonymity—while providing balanced energy distribution. Balanced energy distribution means, all nodes have a uniform energy level at any specific time slot. Our scheme efficiently removes the energy hole problem [29]; it will not be the case that one sensor node will have 90% energy, while the other will have 20% energy. Traditional WSNs requirements may be valid for SDN approach; however, SDN model needs its own set of requirements. Forwarding data to SDBS, there must be a low duty cycle mechanism to provide long term battery power. It had better supply wireless charging facility, but it would incur design and cost issue. Sensor nodes must support an efficient installation of new rules and policies. Communication between sensor node and SDBS must be well-designed and secured. Control plane must provide better set of abstractions and APIs to an application layer (for supporting multiple applications); such a platform will be an open standard platform with no proprietary based hardware and software. Proprietary based solutions are rigid to go beyond their products and, thus, lack in innovations and flexibility. With SDN support, WSNs looks like plug and play systems. Moreover, in case of any catastrophic situation such as, controller failure, multiple SDN controllers are opted to provide backup and redundancy. Multiple controllers' issues, for instance, control packet overhead, multiple controllers synchronization, and the master slave role must be well tackled to reduce network complexity.

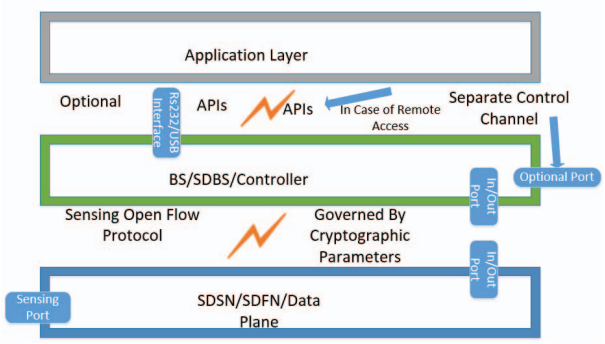

Fig. 1: Basic Architecture of a Software Defined WSN.

III. Architecture and Proposed Model Inspired by SDN

Sensor node as shown in Figure 1, called Software Defined Sensor Node *(SDSN)* behaves like an SDN switch. SDSN main functions are: sensing the surrounding data, buffering data if instructed by controller, internal processing of data if necessary, receiving instructions from controller and feedback, and sending data packets to controller. SDSNs have *flow tables* and *flow table entries* where different rules are matched like drop packet, and forward packet. SDSN also has neighbor nodes' list. This list is shared with other neighbor nodes and SDBS; neighbor nodes' list helps where to forward a specific packet. To reduce the communication complexity, all SDSNs have two ports: sensing port senses the outside environment, and input/output port forwards data and control packets to the controller.

In addition, the LPSDN uses a special node, called Software Defined Forwarding Node *(SDFN)* or Software Defined Buffering Node *SDBN*. Controller can dynamically direct any SDSN to behave like SDFN. Because BS location anonymity depends on SDFNs together with SDSNs, therefore, SDFNs are selected very intelligently by the controller. The intelligent selection is explained in the beginning of Section IV. When a message arrives to any SDFN, it is buffered for some time

or some predefined number of packets threshold; SDFN then checks for similar messages. In such a case, SDFN only forwards that single message to BS. This method can reduce the huge traffic density and battery consumption. Without SDFNs, for 20 packets, for example, all 20 packets traverse via some nodes to reach BS, but in LPSDN, 20 packets reach to some SDFN, and for similar messages, only one packet is forwarded to the controller. This technique reduces battery power consumption, increases network performance, manages network operations, and reduces traffic density surroundings BS. Some amount of traffic density is shifted towards SDFNs; consequently, SDFNs receive more packets. However, for similar messages, SDFNs send only one packet to BS. Shifting traffic density from BS to other SDFNs can guard against traffic analysis attack.

SDBS is responsible to communicate with all SDSNs and SDFNs, collect information, generate network map, and create topology. SDBS is a node that has more memory capacity, more computational power, and enough power supply [2]. SDBS has a secure communication channel to interact with SDFN, SDSN, and application layer. Communication channel must be well protected to ensure cryptographic parameters such as confidentiality, integrity, and availability. The optional separate control channel provides advantages of security and dedicated communication for control packets; however, it incurs extra manufacturing cost. The upper layer is the application layer where an application developer can build applications. Proper set of APIs must be provided to interconnect with SDBS. This is the application layer where the network operator can change the behavior of overall network.

LPSDN provides flexibility to host both SDBS and application layer in one physical machine, for instance, in a PC. In such case, the SDBS must have proper communication port to interact with a computer; serial or USB interface can be used for such communication. On the other hand, SDBS may be deployed in an area, for example, battle field. In such case, wireless communication will take place. This behavior is illustrated clearly in Figure 1. Control Packet Discovery *(CPD)* is broad-casted to all neighbor nodes, which is further broad-casted to other neighbor nodes. Upon receiving CPD packet, all neighbor nodes send Control Packet Discovery Response *(CPDR)* to SDBS. These information are processed by the map module of SDBS to generate the network map and topology. Network operator, then, can monitor the whole network. SDSN behavior can be changed to SDFN dynamically via SDBS's intervention; traditional WSNs lack this facility. SDSN can forward data; it can operate in a sleep mode, and it can be directed to stop communication with other nodes except SDBS. All these on-demand modes are not available in traditional WSNs.

Unlimited number of control messages and events are generated in any network at high-speed are enough to overload any centralized controller [17]. This situation leads to BS's unavailability. To guard against single controller failure, it is optional to have backup controllers. In critical application like military, synchronization among controllers should be applied after some time, for example, five minutes or so on. Of course, it costs extra controllers and communication overhead; however, multiple controllers provide better reliability and high availability. In traditional WSNs, predefined aggregator nodes or cluster nodes [30] are used to collect data from limited portion (neighbor nodes) of the deployed network. SDFN is very similar to aggregator node, but with more power and control; it is not predefined —any SDSN can behave like SDFN dynamically.

SDSN has distinct modules for different purposes. SDSN must have at least one hardware flow table for fast packet processing. More hardware flow tables provide better performance, but they incur extra design and capital cost. SDSN also has software flow tables; these are relatively slow. These flow tables have flow entries, where action-match operations take place. Upon a table-miss (if no rule matches an action like drop, modify, and forward), packets are directed towards a controller; the controller generates rules for all table-miss packets. For aggregator or cluster node, in-network processing module is introduced to process in-situ data aggregation. This mechanism can reduce redundant communication and, thus, provides prolonged network life [31], [32]. Neighbor list module has information about the neighbor nodes' list; this list can be shared with other neighbor nodes periodically to calculate the topology.

Control plane is responsible for handling all control requests, data packets, flow definition and modification rules, topology generations, and other related tasks. Authors [33] have proposed a robust NOS for SDN, but it lacks WSN support.

Application layer gives an interface (graphical or command-line) to end-users for interaction and network operations. End-user can monitor different parameters of nodes: node's energy capacity, topology status, link quality, and number of failed nodes. In managing and maintaining part, end-user (administrator, network operator) can change the behavior of deployed nodes dynamically. End-user can modify node's code, generate high traffic density zones via multiple controllers or SDFNs to deceive an adversary, and can direct deployed nodes to change the traffic pattern.

As discussed in Section I that sink-node gathers a huge volume of data traffic and control traffic from SDSNs and SDFNs. Traffic density surrounding SDBS is very dense as compared to other SDSNs. This can lead to a dense traffic pattern—high traffic density zone—in BS surroundings. The generated pattern, finally, gives a clue to an adversary about BS location information. We provide BS location anonymity against traffic analysis attack via shifting dense traffic pattern from SDBS to other SDFNs and creating on-demand traffic density zones.

Theorem III.1. *In a sensor network, according to Pareto's Law [34],* 20% *of the network nodes carrying about* 80% *of the network traffic. This means nodes near BS, which are* 20% *of all nodes, can carry* 80% *of the entire network traffic. This situation can create high density traffic pattern, which is*

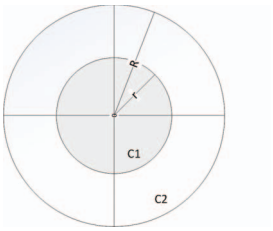

Fig. 2: Integral Model of WSN Traffic.

extremely vulnerable to BS attack.

Proof. Large area of radius R, and sensor nodes which are uniformly distributed with some number N are supposed, where R can accommodate maximum nodes as possible. Nodes generate packets towards BS; this phenomenon can be modeled as an area of a cumulative integral model, which is shown in Figure 2.

An area of the inside circle $C1$ is calculated as Eq. (1).

$$C1 = f(r) = \pi r^2 \tag{1}$$

To calculate the network traffic at inside circle $C1$, the integral of this area, i.e. traffic at this area is:

$$Traffic_{C1} = \int_0^r f(r)\,dr \tag{2}$$

Where $f(r) = \pi r^2$. Solving Eq. (2), we get traffic density at circle $C1$:

$$Traffic_{C1} = \frac{\pi r^3}{3} + Constant \tag{3}$$

The annular region area $R - r$ can be calculated as:

$$\delta r = \pi R^2 - \pi r^2 \tag{4}$$

Total network traffic in outside circle $C2$ can be calculated as:

$$Traffic_{C2} = \int_0^R f(r)\,dr = \int_0^r f(r)\,dr + \int_r^R f(\delta r)\,dr \tag{5}$$

Solving Eq. (5), yields:

$$Traffic_{C2} = \frac{2\pi r^3 + 2\pi R^3 - 3\pi R^2 r}{3} + Constant \tag{6}$$

To prove Theorem III.1, we calculate the ratio of whole network traffic $C2$ to small circle area $C1$, i.e.

$$\frac{Traffic_{C2}}{Traffic_{C1}}$$

Which yields:

$$\frac{2r^3 + 2R^3 - 3R^2 r}{r^3} + Constant \tag{7}$$

Putting different values for r and R, Eq. (7) supports our claim. When $r = 2$ and $R = 5$, Eq. (7) shows $4:2$, when $r = 5$ and $R = 9$, Eq. (7) shows $3.944:1$, and when $r = 9$ and $R = 16$, Eq. (7) shows $3.75:1$.

Random values for r and R yield unexpected results. Care must be taken to choose as pragmatic values as possible based on deployed area and understanding the covered area by all nodes. Constant value in the Eq. (7) can help to set the results more smoothly. □

- According to above proof, 20% nodes are generating 80% of the whole network traffic, and 80% of nodes are merely generating 20% of the entire network traffic. The intelligent adversary will only search in 20% area of traffic, which can exploit BS location information.
- As there is only one high traffic zone in 20% of the network traffic area, this will give a very short amount of attack time to an adversary. The remaining 80% area will be of very limited interest for him. LPSDN generates multiple traffic density zones to deceive an adversary. In addition, for tracing the BS location, LPSDN gave a zigzag path to an adversary. This path is created by the intelligent random selection of SDFNs and their forwarding behavior, which is explained in Section IV.

In a traditional WSN BS location privacy, a predefined proprietary embedded algorithm runs to generate fake packets and traffic density zones. This hardware-software confined architecture faces problems of flexibility, programmability, and maintainability. If any node fails or needs firmware upgradation, a manual operation is required to troubleshoot the issue. LPSDN technique can change the traffic behavior on-demand; it can also generate traffic density zones according to network operator's desire. For mission-critical application such as battle field surveillance, network operator creates more hotspots to deceive an adversary; he selects some SDSNs from the deployed network and instructs them to behave like SDFN. Furthermore, inactive nodes that are not monitoring an area for a long time, can be suspended to save battery power.

IV. Performance Analysis and Experimental Results

There are two cases to protect BS location against traffic analysis attack: global adversary, and local adversary.

- Global adversary: When BS location is static, a global adversary, who sits for a long time, can deduce the SDBS location based on network traffic pattern. As this is the one hotspot location that is not changing all the times. Static traffic pattern gives a clear clue to the global adversary about the constant BS position. To counter this situation, two strategies are proposed: displacement of SDBS, and usage of multiple controllers combined with SDFNs. SDFNs can be used to generate fake traffic density zones, while multiple controllers can be used to shift the constant traffic pattern from one controller to another. This introduces extra cost; however, it provides better network availability and security.
- Local adversary: Local adversary searches for high communication zone to know about BS location. He monitors a particular area's traffic; when he observes high traffic zone, based on traffic analysis attack, he deduces BS location information. To guard against the local adversary, LPSDN uses a single controller with multiple SDFNs. BS instructs some SDSNs to behave like SDFNs. Because,

SDFNs have huge traffic volume in their surroundings, an adversary will believe this high communication zone as a real BS traffic zone. LPSDN shifts traffic from one SDFN to another; the local adversary will jump to the newly created dense traffic zone. Consequently, he will follow this zigzag path and will not be able to deduce the real BS location.

We performed various experiments and presented the results of average results; the evaluation parameters are taken from Texas Instruments mote CC2530 [35] like mote's current capacity, Tx power, Rx power, and energy dissipation during packet transmission. These evaluation parameters have provided a realistic output for our analysis. We have done all these experiments for $30*30$ nodes, 500 messages, while increasing number of SDFNs. In the simulation, BS has $6000*2$ Joule of energy, each SDSN has 6000 Joule of energy, and each SDFN has $6000*2$ Joule of energy. Each receiving and transmitting process can consume 2 Joule of energy.

- With loop method: In this method, any message can visit the same node multiple times. It quickly exhausts nodes' energy and, thus, encounters huge messages drop. Very few messages are dropped when more SDFNs are used; however, it incurs maximum energy drain and very long packet delay. In this method, for all experiments, node selection is random, which is governed by the Mersenne Twister [36] method; this method is widely used in Pseudo Random Number Generator (PRNG). While performing numerous experiments, Mersenne Twister method provided better random distribution over Java embedded random method.
- Without loop method: A node can be visited only once by a message. If already visited node is selected again by the same message ID, that message is simply dropped, and we report this. Due to this property, Figure 4 shows a substantial message drop.
- Energy based with loop method: This method is the core of our algorithm. Neighbor nodes are not randomly selected, but they are intelligently selected on their remaining energy capacity. Higher the remaining energy of a neighbor node, higher is the selection of that node. A message can visit the same node again to deliver a message properly and avoid a message drop. We also conducted experiments for energy based without loop method, which encountered a huge delay and enormous energy exhaustions. This is the reason we have not included its results in all figures.
- Energy based acknowledgment: To see the difference of an acknowledged message with a remembered path, we performed further simulations to compare with *energy based with loop method*. This method and energy based with loop method work the same, but energy based acknowledgment method selects the same path for a message, from which it receives a message. We have done these experiments to tackle any dead node phenomenon, when sending back the acknowledge message to sender node ID. Dead node cannot forward acknowledge message to its neighbors. The lost acknowledgment situation consumes double energy (retransmission) as compared to energy based with loop method.

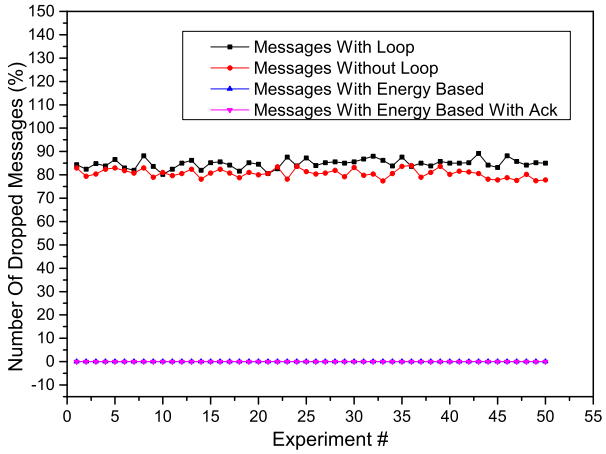

Fig. 3: Number of Dropped Messages with Single Controller.

In Figure 3, number of dropped messages are elaborated. For dropped messages simulation, each individual experiment shows message failure's percentage. The average drop messages' ratio is very huge in with loop (85%) and without loop method (80%), while it is zero for energy based and energy based with ack. In practical situation, controller can drop some messages. When we increased the number of SDFNs to 4, the message drop ratio in with loop and without loop method also decreased to 38% and 42% respectively.

For 9 SDFNs as shown in Figure 4, message drop ratio almost goes down to zero. On the average, 20% message drop ratio is observed. In without loop method, a message cannot visit the same node twice, which finally introduces a massive packet drop. Huge packet drop incurs unavailability and longer delay. From here, we can strongly conclude, that *due to this enormous message drop ratio, without loop method should not be used to provide BS location anonymity.*

With loop method quickly exhausts node's energy. Initially, all nodes have full energy, but after running for some time, with loop method drains all nodes' energy quickly, as depicted in Figure 5. In hops count simulation, number of steps (hops count) for individual message ID to reach BS is elaborated. Hops count is very high (on average 30000 steps) in with loop method as shown in Figure 5. Big hops count suffers from longer delay.

For 4 SDFNs, hops count decreased to 3000 steps (on average); it improved node's energy as well. Because of longer delay and maximum energy consumption, with loop method is very difficult to use in practical situation.

For 9 SDFNs, with loop method still suffered the same problem—longer delay (on average 2000 steps). Other methods are good enough to deliver packets quickly. Our second conclusion is, as deduced from Figure 6, *it is not advised to use with loop method due to its longer delay*.

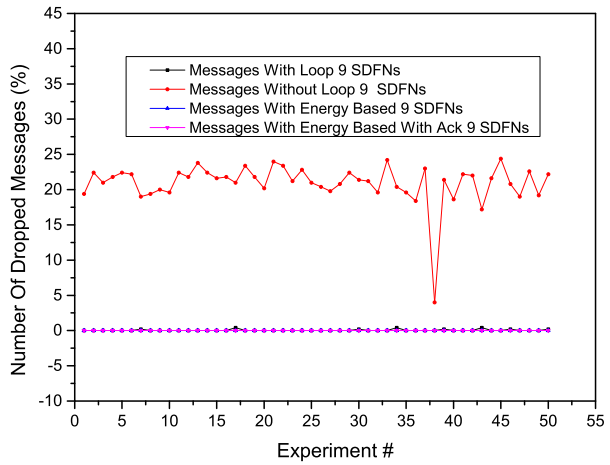

Fig. 4: Number of Dropped Messages with 9 SDFNs and Single Controller.

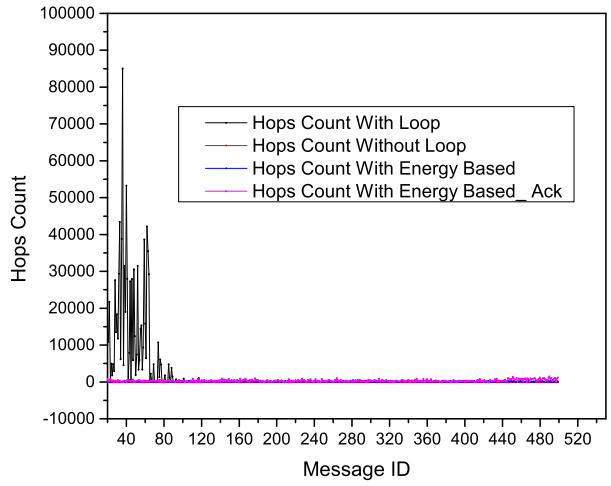

Fig. 5: Hops Count for Single Controller.

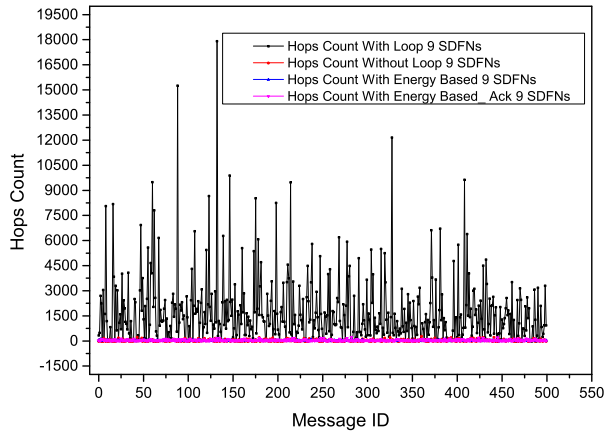

Fig. 6: Hops Count for 9 SDFNs and Single Controller.

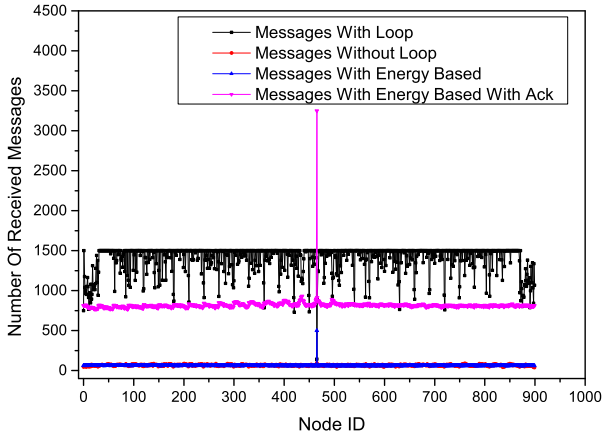

Fig. 7: Number of Received Messages (per node) for Single Controller.

In Figure 7, number of received messages per node are compared. BS location information is very prone to traffic analysis attack, because of only one peak (high communication traffic zone) as shown in Figure 7. To provide BS location privacy, we can use multiple controllers or multiple SDFNs to have many traffic peaks. Multiple traffic peaks/traffic density zones can deceive an adversary; we clearly illustrated this case in Section IV.

For the same experiment, but with 4 SDFNs, energy based method outperformed other methods. In this case, simulation results showed 5 peaks, minimum number of packet received, and shorter delay; such parameters encourage to use energy based method. Energy based with acknowledge method suffered from large volume of received messages, as a result, it can drain node's energy quickly.

With 9 SDFNs as shown in Figure 8, we have 10 peaks for energy based method. Multiple traffic density zones give a daunting challenge to an attacker (global and local). Multiple traffic peaks provide better BS location privacy; an adversary will have more traffic density zones for his attack space. Increasing the number of SDFNs/Controllers, finally, provides better BS location privacy. For an attacker, a single high traffic density zone gives higher success probability equal to $1/1$, while for multiple traffic peaks, his success probability decreases to $1/\{No-Of-Peaks\}$.

The graph behavior for the number of sent messages is almost same to the behavior of received messages. We do not includ those graphs here.

First-order radio formula [37] where authors have discussed how to estimate the energy needed to send a packet of a bits of data from the transmitter to the receiver, can be used to model the energy consumption in LPSDN. Normally, packet transmission consumes more energy compared to packet reception, because an extra process of signal amplification takes place in transmitting electronics.

The general formula for transmitting a bits is shown in Eq.(8) where $E_{Tx}(a)$ is the energy that the radio circuit consumes in order to process a bits, aE_{Tx} is the amount

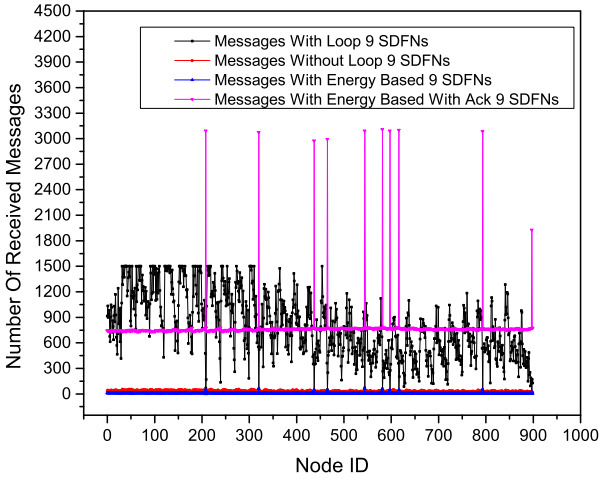

Fig. 8: Number of Received Messages (per node) with 9 SDFNs and Single Controller.

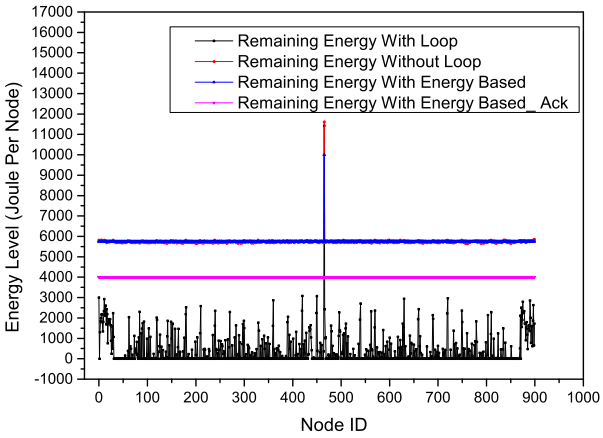

Fig. 9: Remaining Energy (per node) with Single Controller.

of energy consumed by processing a single bit by the radio circuit, and $E_{amp}(a,b)$ is the energy needed by the radio amplifier circuit to send a bits of message over distance b. For receiving a bits, the receiving formula is shown in Eq. (9), where aE_{Rx} is the energy consumed by receiving circuit for processing a single bit.

$$E_{totalTx}(a,b) = E_{Tx}(a) + E_{amp}(a,b) = aE_{Tx} + E_{amp}(a,b) \quad (8)$$

$$E_{totalRx}(a) = E_{Rx}(a) = aE_{Rx} \quad (9)$$

Total delay can be calculated by adding processing delay, queuing delay and propagation delay. Eq. (10) is the general formula to model point to point delay. T_p is the processing delay, $Delay_{que}$ is the queuing delay, and $Delay_{prop}$ is the propagation delay. We simulated this case related to the number of packets received or transmitted by any node.

$$Delay_{total} = T_p + Delay_{que} + Delay_{prop} \quad (10)$$

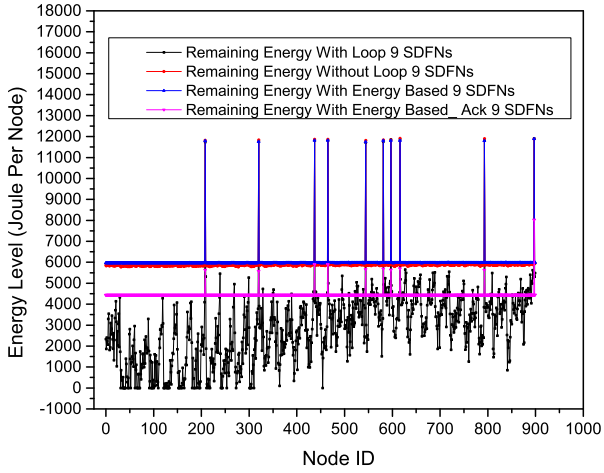

Fig. 10: Remaining Energy (per node) With 9 SDFNs and Single Controller.

In Figure 9, the remaining energy comparison per node is illustrated. Remaining energy with loop method suffered from ample power exhaustion; it should not be used in practical applications. Remaining energy with energy based acknowledge method also encountered quick power loss, because of substantial messages' exchange. For single controller, remaining energy levels for energy based method and without loop method were almost same, but without loop method suffered from larger delay. Consequently, it should not be used in practical applications.

For 4 SDFNs, the remaining energy level in energy based method outperformed other methods.

With 9 SDFNs as shown in Figure 10, remaining energy per node is a bit improved. Remaining energy with energy based acknowledge and without loop method did not suffer from a quicker power drain, but they incurred the disadvantages of longer delay and huge message drop.

A. Discussion

The experimental results show that LPSDN efficiently hides BS location and, thus, traffic analysis attack will be very difficult for local and global adversaries.

- Dropped messages: Drop message ratio is inversely proportional to the number of SDFNs. Without loop method incurs a huge amount of message failure; the reason is that, a message cannot traverse the same node again. If any message visits the same node again, without loop method changes to with loop method. With loop method with 9 SDFNs performs very well; the message dropped ratio goes down from 80% to below 1%, but other issues such as, longer delay and massive energy consumption are the key barriers towards practical implementation.
- Hops count/Latency: Hops count for with loop method suffers huge delay, other methods behave almost same. Although, in with loop method, message drop ratio for 9 SDFNs is below 1%, but, because of huge delay, with loop method is not suited for mission-critical applications.

TABLE I: Analysis and Comparison of Different Methods.

Experiment	Single Controller				4 SDFNs + Single Controller				9 SDFNs + Single Controller			
	Dropped Messages	Hops Count	Sent or Received Messages	Remaining Energy	Dropped Messages	Hops Count	Sent or Received Messages	Remaining Energy	Dropped Messages	Hops Count	Sent or Received Messages	Remaining Energy
With Loop	Ultra high 85%	Ultra high	Ultra dense	Very huge consumption	Very high 40%	Very high	Ultra dense	Very huge consumption	Very low 0.5%	high	Very high	Very huge Consumption
Without Loop	Ultra high 80%	Very low	Optimal	Optimal consumption	Very high 45%	Very low	Optimal	Optimal consumption	High 25%	Very low	Optimal	Optimal consumption
Energy Based	Zero	Very low	Optimal	Optimal consumption	Zero	Very low	Optimal	Optimal consumption	Zero	Very low	Optimal	Optimal consumption
Energy Based With Ack	Zero	Very low	Very dense	Huge consumption	Zero	Very low	Very dense	Huge consumption	Zero	Very low	Very dense	Very huge consumption

- Adversary attack's space: For the local adversary, we can change the traffic pattern at some location on-demand, but for the global adversary, who monitors the network traffic all times, single controller is extremely vulnerable to exploit BS location information as explained in Section IV. For security purposes, multiple SDFNs with single controller and multiple SDFNs with multiple controllers must be used to provide BS location anonymity against the global adversary.
  In case of multiple SDFNs and single controller, attacking BS location is very difficult for the local adversary. In this case, BS position is static all times. To support BS location anonymity, multiple traffic peaks generated by other SDFNs are used. Traffic peaks generated by SDFNs can be changed on-demand after time to time. Figure 7 shows the network traffic for the first-time span, but when location privacy is required, on-demand traffic peaks are generated as depicted in 8. Multiple controllers are not used here. Network operator can direct any SDSN to behave like SDFN (for changing traffic pattern and generating traffic peaks). For the local adversary, it is very difficult to deduce BS location, while for the global adversary, he has to sit for a long time to analyze traffic pattern and deduce BS location information. *Only energy based method provides balanced peaks and can hide BS very efficiently.*
  BS location is more secured against the global adversary, when multiple controllers are used with multiple SDFNs. Traffic density at one constant position raised by a single controller can be transferred to other controllers and SDFNs. This technique leads to maximum BS location privacy against the global adversary. Global adversary, who monitors network all times will see many traffic peaks; these peaks change after time to time. Multiple controllers with multiple SDFNs can be used in military applications where security is the top priority. However, multiple controllers incur an extra capital cost.
- Remaining energy: Energy based method and without loop method perform well to conserver more energy. Other two methods suffer from a quick power drain. Without loop method suffers from an enormous amount of packet loss as shown in Figure 4.
- No fake packets: LPSDN does not use fake packets to provide BS location privacy. It saves an enormous amount of node's energy, which finally provides a prolonged network life—it ensures availability. In traditional WSNs, dummy packets and fake packets play an important role to hide BS location information. However, fake packets consume extra energy. A comprehensive comparison using fake packets is already presented in Section II-A.
- Network maintenance and management: Traditional WSNs lack of decoupled architecture; both forwarding behavior and routing behavior are embedded in a single sensor node. Such architecture leads to lack of flexibility, maintainability, manageability, and easy troubleshooting. In LPSDN, a centralized controller defines policy and decides all routing decisions. It decouples the intelligent behavior (routing) from all sensor nodes. Sensor nodes only forward packets and all packet rules are defined in the controller.
  Based on experimental results and rigorous simulations, it is strongly inferred that for mission-critical systems like, military and field surveillance, energy based method must be used. This method provides minimum hops count (low latency), balanced energy consumption and distribution, and minimum message failure, while providing BS location privacy against local and global adversary. A comprehensive comparison is shown in Table I.

V. Conclusion

In WSNs, sink-node gathers data from surrounding nodes and forwards to outside world via a gateway. Sink-node, which is the bridge between deployed sensor network and outside world, has a vital role in sensor network operations. Therefor, its location information must be hidden from any attacker. To protect BS location privacy against local and global adversary, we proposed a novel technique inspired by the SDN paradigm. Our scheme efficiently hides BS location with minimal energy consumption, minimum delay, and negligible message failure rate. In addition, our approach efficiently reduces the management and maintenance overhead; data plane and control plane are tightly coupled in traditional WSN, which leads to expensive management and maintenance activities. The proposed technique solves the scalability issues as well; new node addition, faulty node removal, software installation, and firmware up-gradation, are well tackled as compared to traditional WSNs.

Acknowledgments

We thank the anonymous reviewers whose comments noticeably improved the quality of this paper. This work is

supported in part by the National 973 Program of China under grant 2011CB302301, the Fundamental Research Funds for the Central Universities (HUST:2014QN009), Natural Science Foundation of Hubei Province (2015CFB192), and Higher Education Commission Pakistan (HEC).

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, Mar. 2002.

[2] M. Conti, J. Willemsen, and B. Crispo, "Providing source location privacy in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1238–1280, Mar. 2013.

[3] W. Diffie and S. Landu, *Privacy on The Line: The Politics of Wiretapping and Encryption*, updated and expanded ed. MIT Press, 2007.

[4] J. Long, M. Dong, K. Ota, and A. Liu, "Achieving source location privacy and network lifetime maximization through tree-based diversionary routing in wireless sensor networks," *IEEE Access*, vol. 2, pp. 633–651, Jul. 2014.

[5] R. Rios and J. Lopez, "(Un)Suitability of anonymous communication systems to WSN," *IEEE Systems Journal*, vol. 7, no. 2, pp. 298–310, Jun. 2013.

[6] K. Pongaliur and L. Xiao, "Sensor node source privacy and packet recovery under eavesdropping and node compromise attacks," *ACM Transaction on Sensor networks*, vol. 9, no. 4, pp. 50:1–50:26, Jul. 2013.

[7] A. Proaño and L. Lazos, "Perfect contextual information privacy in WSNs undercolluding eavesdroppers," in *Proc. of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, Apr. 2013, pp. 89–94.

[8] Y. Ebrahimi and M. Younis, "Using deceptive packets to increase base-station anonymity in wireless sensor network," in *Proc. of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, Jul. 2011, pp. 842–847.

[9] V. P. V. Gottumukkala, V. Pandit, H. Li, and D. P. Agrawal, "Base-station location anonymity and security technique (BLAST) for wireless sensor networks," in *Proc. of the IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 6705–6709.

[10] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, May 2007, pp. 1955–1963.

[11] X. Luo, X. Ji, and M.-S. Park, "Location privacy against traffic analysis attacks in wireless sensor networks," in *Proc. of the International Conference on Information Science and Applications (ICISA)*, Apr. 2010, pp. 1–6.

[12] B. Ying, D. Makrakis, and H. T. Mouftah, "A protocol for sink location privacy protection in wireless sensor networks," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*. IEEE, Dec. 2011, pp. 1–5.

[13] B. Ying, J. Gallardo, D. Makrakis, and H. Mouftah, "Concealing of the sink location in wsns by artificially homogenizing traffic intensity," in *Proc. of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2011, pp. 988–993.

[14] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012.

[15] R. Rios and J. Lopez, "Analysis of location privacy solutions in wireless sensor networks," *IET Communications*, vol. 5, no. 17, pp. 2518–2532, Nov. 2011.

[16] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[17] M. Ambrosin, M. Conti, F. De Gaspari, and R. Poovendran, "LineSwitch: Efficiently managing switch flow in software-defined networking while effectively tackling DoS attacks," in *Proc. of the the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, Apr. 2015, pp. 639–644.

[18] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2181–2206, May 2014.

[19] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[20] N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey," *ACM Computing Survey*, vol. 47, no. 2, pp. 27:1–27:11, Nov. 2014.

[21] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, May 2007, pp. 1955–1963.

[22] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proc. of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*. IEEE, Sep. 2005, pp. 113–126.

[23] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software defined wireless networks: Unbridling SDNs," in *Proc. of the European Workshop on Software Defined Networking (EWSDN)*, Oct. 2012, pp. 1–6.

[24] Z.-J. Han and W. Ren, "A novel wireless sensor networks structure based on the SDN," *International Journal of Distributed Sensor Networks*, Mar. 2014.

[25] B. Trevizan de Oliveira, C. Borges Margi, and L. Batista Gabriel, "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks," in *Proc. of the IEEE Latin-America Conference on Communications (LATINCOM)*, Nov. 2014, pp. 1–6.

[26] M. Tasch, R. Khondoker, R. Marx, and K. Bayarou, "Security analysis of security applications for software defined networks," in *Proc. of the AINTEC on Asian Internet Engineering Conference*. ACM, Nov. 2014, pp. 23:23–23:30.

[27] R. Riggio, T. Rasheed, and M. K. Marina, "Poster: Programming software-defined wireless networks," in *Proc. of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, Sep. 2014, pp. 413–416.

[28] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, Nov. 2013, pp. 413–424.

[29] Y. Xue, X. Chang, S. Zhong, and Y. Zhuang, "An efficient energy hole alleviating algorithm for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 3, pp. 347–355, Aug. 2014.

[30] I. Ouafaa, L. Jalal, K. Salah-ddine, and E. H. Said, "The comparison study of hierarchical routing protocols for ad-hoc and wireless sensor networks: A literature survey," in *Proc. of the The International Conference on Engineering & MIS*. ACM, Sep. 2015, pp. 32:1–32:8.

[31] B.-D. Lee and K.-H. Lim, "An energy-efficient hybrid data-gathering protocol based on the dynamic switching of reporting schemes in wireless sensor networks," *IEEE Systems Journal*, vol. 6, no. 3, pp. 378–387, Sep. 2012.

[32] B. Sun, X. Shan, K. Wu, and Y. Xiao, "Anomaly detection based secure in-network aggregation for wireless sensor networks," *IEEE Systems Journal*, vol. 7, no. 1, pp. 13–25, Mar. 2013.

[33] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, "Rosemary: A robust, secure, and high-performance network operating system," in *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, Nov. 2014, pp. 78–89.

[34] F. John Reh, "Pareto's principle-the 80-20 rule," *Business Credit*, vol. 107, no. 7, Aug. 2005.

[35] Texas Instruments, "CC2530 second generation System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / RF4CE / ZigBee," http://www.ti.com/product/CC2530/technicaldocuments, 2015, [Online] Accessed 2015-9-9. [Online]. Available: http://www.ti.com/product/CC2530/technicaldocuments

[36] M. Matsumoto and T. Nishimura, "Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

[37] D. Tudose, L. Gheorghe, and N. Tapus, "Radio transceiver consumption modeling for multi-hop wireless sensor networks," *UPB Scientific Bulletin Series C*, vol. 75, no. 1, pp. 17–26, Jan. 2013.