Traffic Decorrelation Techniques for Countering a Global Eavesdropper in WSNs

Alejandro Proaño, Loukas Lazos, and Marwan Krunz
Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA
E-mail:{aaproano, llazos, krunz}@ece.arizona.edu

Abstract—We address the problem of preventing the inference of *contextual information* in event-driven wireless sensor networks (WSNs). The problem is considered under a global eavesdropper who analyzes low-level RF transmission attributes, such as the number of transmitted packets, inter-packet times, and traffic directionality, to infer event location, its occurrence time, and the sink location. We devise a general traffic analysis method for inferring contextual information by correlating transmission times with eavesdropping locations. Our analysis shows that most existing countermeasures either fail to provide adequate protection, or incur high communication and delay overheads. To mitigate the impact of eavesdropping, we propose resource-efficient traffic normalization schemes. In comparison to the state-of-the-art, our methods reduce the communication overhead by more than 50%, and the end-to-end delay by more than 30%. To do so, we partition the WSN to minimum connected dominating sets that operate in a round-robin fashion. This allows us to reduce the number of traffic sources active at a given time, while providing routing paths to any node in the WSN. We further reduce packet delay by loosely coordinating packet relaying, without revealing the traffic directionality.

Index Terms—Wireless Sensor Networks (WSN), eavesdropping, contextual information, privacy, anonymity, graph theory.

1 Introduction

Wireless sensor networks (WSNs) have shown great potential in revolutionizing many applications including military surveillance, patient monitoring, agriculture and industrial monitoring, smart buildings, cities, and smart infrastructures. Several of these applications involve the communication of sensitive information that must be protected from unauthorized parties. As an example, consider a military surveillance WSN, deployed to detect physical intrusions in a restricted area [21], [25]. Such a WSN operates as an event-driven network, whereby detection of a physical event (e.g., enemy intrusion) triggers the transmission of a report to a sink.

Although the WSN communications could be secured via standard cryptographic methods, the communication patterns alone leak *contextual information*, which refers to event-related parameters that are inferred without accessing the report contents. Event parameters of interest include: (a) the event location, (b) the occurrence time of the event, (c) the sink location, and (d) the path from the source to the sink [10], [20], [23], [29]. Leakage of contextual information poses a serious threat to the WSN mission and operation. In the military surveillance scenario, the adversary can link the events detected by the WSN to compromised assets. Moreover, he could correlate the sink location with the location of a command center, a team leader, or the gateway. Destroying the area around the sink could have far more detrimental impact than targeting any other area. Similar operational concerns arise in personal applications such as smart homes and body area networks. The WSN communication patterns could be linked to one's activities, whereabouts, medical conditions, and other private information.

Contextual information can be exposed by eavesdropping on over-the-air transmissions and obtaining *transmission attributes,* such as inter-packet times, packet source and destination IDs, and number and sizes of transmitted packets. As an example, consider the detection of event Ψ by sensor v_1 in Fig. 1. Sensor v_1 forwards an event report to the sink via v_2, v_5, and v_6. Transmissions related to this report are intercepted by eavesdroppers $e_1 - e_5$. The event location can be approximated to the sensing area of v_1. The latter can be estimated as the interception of the reception areas of e_1 and e_4, which overhear v_1's transmissions. Moreover, the event occurrence time can be approximated to the overhearing time of v_1's first transmission.

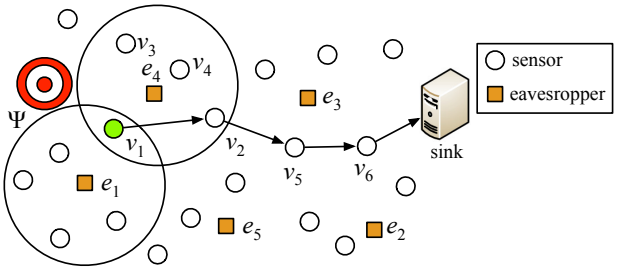

Fig. 1: Detection of event Ψ by eavesdroppers $e_1 - e_5$.

Defending against eavesdropping poses significant challenges. First, eavesdroppers are passive devices that are hard to detect. Second, the availability of low-cost commodity radio hardware makes it inexpensive to deploy a large number of eavesdroppers. Third, even if encryption is applied to conceal the packet payload, some fields in the packet headers still need to be transmitted in the clear for correct protocol operation (e.g., PHY-layer headers used for frame detection, synchronization, etc.). These unencrypted fields facilitate accurate estimation of transmission attributes.

The problem of preserving contextual information privacy has been studied under various adversarial scenarios. Threat models can be classified based on the adversary's network view (local vs. global) or the capabilities of the eavesdropping devices (packet decoding, localization of the transmission source, etc.). Under a local

A preliminary version appeared at the ACM WiSec 2013 Conference.

model, eavesdroppers are assumed to intercept only a fraction of the WSN traffic [12], [16]–[20]. Hiding methods include random walks, adding of pseudo-sources and pseudo-destinations [14], [17]–[19], [27], creation of routing loops [12], and flooding [12]. These methods can only provide probabilistic obfuscation guarantees, because eavesdroppers locations are unknown. Under a global model, all communications within the WSN are assumed to be intercepted and collectively analyzed [7], [20], [29]. State-of-the-art countermeasures conceal traffic associated to real events by injecting dummy packets according to a predefined distribution [4], [20], [23], [28]. In these methods, real transmissions take place by substituting scheduled dummy transmissions, which decorrelates the occurrence of an event from the eavesdropped traffic patterns. However, concealment of contextual information comes at the expense of high communication overhead and increased end-to-end delay for reporting events.

Our Contributions: We study the problem of resource-efficient traffic randomization for hiding contextual information in event-driven WSNs, under a global adversary. Our main contributions are summarized as follows:

- We present a general traffic analysis method for inferring contextual information that is used as a baseline for comparing methods with varying assumptions. Our method relies on minimal information, namely packet transmission time and eavesdropping location.
- We propose traffic normalization methods that hide the event location, its occurrence time, and the sink location from global eavesdroppers. Compared to existing approaches, our methods reduce the communication and delay overheads by limiting the injected bogus traffic. This is achieved by constructing minimum connected dominating sets (MCDSs) and MCDSs with shortest paths to the sink (SS-MCDSs). We characterize the algorithmic complexity for building SS-MCDSs and develop efficient heuristics.
- To reduce the forwarding delay, we design a rate control scheme that loosely coordinates sensor transmissions over multi-hop paths without revealing real traffic patterns or the traffic directionality.
- We compare privacy and overhead of our techniques to prior art and show the savings achieved.

Organization: Section 2 presents related work. In Section 3, we state the system and adversary models. Traffic analysis techniques for extracting contextual information are presented in Section 4. In Section 5, we introduce our mitigation techniques. We evaluate their privacy and performance in Section 6 and conclude in Section 7.

2 Related Work

Prior art on contextual information privacy can be classified based on the privacy type and the eavesdropper capabilities. Extensive literature reviews can be found in recent surveys [5], [6]. Here, we present related work for countering local and global eavesdroppers.

Local Eavesdropper: A local adversary can intercept a limited number of transmissions within the WSN. Typically, this adversary deploys a single or a few mobile devices that attempt to localize source by backtracing the intercepted transmissions. In [16], the authors proposed the use of multiple routing paths to prevent local adversaries from tracing packets to their source. A sensor with a real packet for transmission forwards it to one neighbor on the shortest path to the sink. Any overhearing sensor that does not belong to the shortest path, broadcasts a dummy packet with some probability. This probability is adapted to maintain the same average communication overhead per sensor.

Mahmoud et al. [17]–[19] considered a highly-capable adversary that can precisely localize the source of a transmission using radiometric hardware. They proposed the *hotspot-locating attack* for identifying areas with high transmission activity and analytically showed that the source can be located via backtracing. To hide the source location, the authors proposed the introduction of dummy traffic from sensor clouds that become active only during real transmissions.

In [22], the authors proposed a two-stage routing method called *phantom flooding*. In the first stage, the source divides its neighbors into two sets, located in opposite directions (e.g., North-South). The source forwards a packet to a randomly selected neighbor in one direction. This neighbor continues to forward the packet in the same manner, but in the opposite direction. The process is repeated until h hops are traversed. In the second stage, the packet is forwarded to the sink using probabilistic flooding. In [14], [15], [27], real packets are diverted to a fake source located several hops away, using unicast transmissions. The fake source forwards packets to the sink using flooding or over the shortest path. These works differ in the selection process of the fake source. In STaR [15], an intermediate node is selected from a *sink toroidal region*. This area forms a ring around the sink, starting from radius r and ending at R. To report an event, the source routes packets to a random destination in the toroidal region. The intermediate fake source relays the packet to the sink via the shortest path.

Global Eavesdropper: In [20] the authors proposed two traffic normalization techniques: periodic collection and source simulation. In periodic collection, each sensor generates bogus packets at a fixed rate. Real packets are transmitted by substituting bogus ones, while maintaining the same total rate (bogus and real). This method hides the source location, the path to the sink, and the sink location, at the expense of significant communication and delay overheads. In the source simulation method, the communication overhead is reduced by limiting dummy traffic to a subset of fake sources. The fake source location is selected to follow the distribution of real events. However, the spatial and temporal event distribution must be known a priori.

The authors in [23] proposed the transmission of bogus traffic by all sensors using a pre-determined probability distribution. To reduce the end-to-end delay, sensors with real packets "rush" their transmissions relative to scheduled dummy transmissions. Future dummy transmissions are delayed to compensate for the rushed real packets. This method is not effective when multiple real packets must be transmitted by the same source. In addition, the authors in [2] proved that the short-long inter-packet time patterns observed due to the rushed transmissions can be used to identify time intervals that contain real packets. To address this vulnerability, they introduced fake short-long patterns.

In [29] the authors proposed several methods for reducing dummy traffic. The network was divided into square cells of size equal to the minimum area unit where events can occur. Each cell generates encrypted bogus traffic, which is replaced with real traffic when available. In the *Proxy-based Filtering Scheme (PFS)*, a subset of cells are designated as proxies. Each cell transmits packets (real or dummy) to the closest proxy, which filters dummy traffic and forwards real packets to the sink. In the *Tree-based Filtering Scheme (TFS)*, proxies are organized as a tree rooted at the sink to expedite packet delivery and reduce the filtered dummy packets. However, TFS reveals the sink location. In [4], the authors proposed the *Optimal Filtering Scheme (OFS)*, in which proxies are organized into a directed graph instead of a tree. This allows each proxy to filter packets from every proxy as well as from individual sensors.

An aggregation-based scheme was introduced in [28]. The WSN is divided into clusters, each with one clusterhead (CH). The CHs are organized in a tree rooted at the sink. Each sensor transmits dummy traffic to its respective CH. The CH is responsible for filtering dummy packets, aggregating real packets, and relaying them to the sink. This method does not hide the sink location, which corresponds to the root of the CH tree.

3 System and Adversarial Model

System Model: We consider a set of sensors $\mathcal{V}$, deployed to sense physical events within a given area. When a sensor detects an event of interest, it sends a report to the sink via a single-hop or a multi-hop route (depending on the relative sensor-sink position). The confidentiality of the report is protected using standard cryptographic methods. Packet transmissions are re-encrypted on a per-hop basis to prevent tracing of relayed packets [3], [17], [19]. Sensors are aware of their one- and two-hop neighbors by using a neighbor discovery service [24]. The sensor communication areas could be heterogeneous and follow any model. The WSN is loosely synchronized to a common time reference [1], [26]. The maximum network-wide synchronization error is Δt. Finally, the wireless medium is assumed to be lossy.

Adversarial Model: We adopt a global adversarial model, similar to the one assumed in [2], [20], [23], [29]. The adversary deploys a set of eavesdropping devices $\mathcal{A}$ that passively monitor *all* WSN transmissions. An eavesdropper $e \in \mathcal{A}$, located at ℓ_e, has a reception area C_e, which could have any shape (reception areas could be heterogeneous and need not follow the unit-disc model). We emphasize that this global adversarial model is a relevant one even when a fraction of the WSN transmissions can be intercepted. In the absence of eavesdropper location information, one has to account for all possible eavesdropping locations to provide *privacy guarantees*, which is equivalent to a global adversarial model. The adversary collectively analyzes the eavesdropped traffic at a fusion center to infer the following information: (a) the location of a physical event, (b) the occurrence time of that event, and (c) the sink location.

To formally define the information at the disposal of the adversary, we introduce the notions of a *transmission set* and an *observation set*. The transmission set is a truthful representation of all WSN transmissions taking place over a period of time. The observation set represents the *actual* information that is captured by the adversary for a specific eavesdropper deployment and assumed capability. Specifically, each packet p_i is associated with a unique signature $\sigma(p_i) = \{h(p_i), t(p_i), \ell(p_i)\}$, where $h(p_i)$ is a hash digest of p_i, $t(p_i)$ is the transmission time of p_i, and $\ell(p_i)$ is the location of the originating sensor. The signature $\sigma(p_i)$ constitutes the *ground truth* for the transmission of p_i. This ground truth may differ from the observation of p_i by an eavesdropper e, who tags p_i with $tag_e(p_i) = \{h(p_i), t(p_i), \ell_e\}$. A $tag_e(p_i)$ differs from $\sigma(p_i)$ in the location attributed to the source of p_i. Instead of $\ell(p_i)$, an eavesdropper e could at least attribute p_i to its own location ℓ_e and approximate $\ell(p_i)$ with accuracy equal to e's reception area C_e. Using the packet signatures and tags, we define the transmission set and observation set as follows.

Definition 1 (Transmission Set): For a sensor $v \in \mathcal{V}$, the transmission set $\Theta_v(W)$, defined over an epoch W is:

$$\Theta_v(W) = \{\sigma(p_i) : \ell(p_i) = \ell_v, t(p_i) \in W\}.$$

The transmission set for the entire network over W is:

$$\Theta(W) = \{\Theta_v(W) : v \in \mathcal{V}\}.$$

Definition 2 (Observation Set): For an eavesdropper e, the observation set $\mathcal{O}_e(W)$ over W, is:

$$\mathcal{O}_e(W) = \{tag_e(p_i) : t(p_i) \in W\}.$$

The observation set captured by $\mathcal{A}$ over W is:

$$\mathcal{O}(W) = \{\mathcal{O}_e(W) : e \in \mathcal{A}\}.$$

We are interested in evaluating the privacy maintained under the analysis of $\mathcal{O}(W)$. We quantify this privacy as the distance between the inferred location based on $\mathcal{O}(W)$ and the location of the source. We call this measure *privacy distance* and formally define it as follows.

Definition 3 (Privacy Distance): Let $\epsilon \in \mathbb{R}^n$ be some private information of interest, estimated as $\Xi \in \mathbb{R}^n$ based on eavesdropping. The *privacy distance* of ϵ is

$$\Pi = \int_{\xi \in \Xi} s(\xi)\, P(\xi)\, \mathrm{d}\xi$$

where $s(\xi)$ is the Euclidean distance between ϵ and $\xi \in \Xi$, and $P(\xi)$ a probability measure over the points in Ξ.

We note that Euclidean distance is a natural measure for evaluating location privacy as it yields the straight-line distance between the source location and its estimate in any dimensional space. As sn example, $\epsilon, \Xi \in \mathbb{R}^2$ for when location privacy is measured and sensors are deployed in two dimensions. For the WSN of Fig. 1, v_1 reports the occurrence of event Ψ during epoch W by transmitting $\Theta_{v_1}(W)$ to the sink. Eavesdroppers e_1 and e_4 capture $\mathcal{O}_{e_1}(W)$ and $\mathcal{O}_{e_4}(W)$ respectively. By jointly analyzing the collected observation sets, the adversary localizes the event source to $\Xi = C_{e_1} \bigcap C_{e_4}$. All points within Ξ are assumed equally likely event sources (there is no further information to bias the event location within Ξ). Therefore $P(\xi) = 1/\text{area}(\Xi)$. For this case,

$$\Pi = \frac{1}{\text{area}(\Xi)} \int_{\xi \in \Xi} s(\xi)\, \mathrm{d}\xi.$$

For a more meaningful evaluation of Π, it must be normalized to some application parameter. We leave the normalization function up to the application designer. In our evaluations, we have opted to normalize Π with the sensor communication range.

4 Traffic Analysis

In this section, we propose a general traffic analysis method for inferring contextual information. Our method is meant as a baseline for evaluating the performance of protection mechanisms with varying underlying assumptions. Therefore, it relies on minimal information, namely the packet interception times and eavesdroppers' locations. Our method is agnostic to the network topology (though it is inferred) and to the specific mechanism used to counter traffic analysis, so that it can be broadly applied. We emphasize that our goal is not to create the most sophisticated attack. Such an attack is highly-dependent on the protection mechanism and may require additional a priori knowledge. As an example, the methods in [2], [23] use sophisticated statistical inference methods to detect real events. However, these are specific to statistical anonymity approaches and assume the a priori knowledge of the probability distribution used to draw inter-packet times. Our method proceeds in the two stages: a traffic cleansing stage followed by a contextual information inference stage. Since our method is applied on a per-epoch basis, we omit the W notation when it is redundant.

4.1 Traffic Cleansing

The observation sets recorded by the scattered eavesdroppers are likely to contain duplicate tags. This is because more than one eavesdroppers may overhear the same packet transmission. In the traffic cleansing stage, the adversary uses duplicate tags in the observation set $\mathcal{O}$ to obtain a better estimation of transmission set Θ.

In Algorithm 1, we present a process for attributing tags to different sensors and eliminating duplicate tags. Specifically, for two eavesdroppers a and e with overlapping reception areas, we divide their respective observation sets to tags intercepted in $C_a \cap C_e$, $C_a \backslash C_e$, and $C_e \backslash C_a$. Each tag set is associated with a *sensor label* that represents the transmissions within the respective area. The location of each sensor label is approximated by the area intersection (difference) between C_a and C_e. Details are described in Algorithm 1.

Algorithm 1: Tag Cleansing

Step 1: For each eavesdropper e, set $\hat{\Theta}_v = \mathcal{O}_e$, $\hat{\ell}_v = C_e$, and $NS_e = \{v\}$. Here, v is a label for any sensor in C_e, $\hat{\ell}_v$ is the approximation area of v's location, and NS_e is the estimated sensor neighborhood of e.

Step 2: For each $\hat{\Theta}_v$ and $a \in \mathcal{A}, a \neq e$, if $\hat{\Theta}_v \cap \mathcal{O}_a \neq \emptyset$ **and** $\hat{\Theta}_v \backslash \mathcal{O}_a \neq \emptyset$, replace $\hat{\Theta}_v$ with

$$\hat{\Theta}_u = \hat{\Theta}_v \cap \mathcal{O}_a, \quad \hat{\Theta}_w = \hat{\Theta}_v \backslash \mathcal{O}_a$$

The intersection and complement set operations are defined based on the packet hash/timestamp dual contained in the tags. Labels u and w represent new sensor labels in e's reception range, i.e., $NS_e = \{u, w\}$.

Step 3: Approximate the locations of u and w by $\hat{\ell}_u = \hat{\ell}_v \cap C_a$ and $\hat{\ell}_w = \hat{\ell}_v \backslash C_a$, respectively.

Step 4: Compute $\mathcal{O}$ and an estimate $\hat{\mathcal{V}}$ of set $\mathcal{V}$ as:

$$\hat{\mathcal{V}} = \{v : v \in NS_e,\ \forall e \in \mathcal{A}\}, \quad \mathcal{O} = \{\hat{\Theta}_v : v \in \hat{\mathcal{V}}\}.$$

Step 5: To eliminate duplicates from $\mathcal{O}$ and $\hat{\mathcal{V}}$, find $\hat{\Theta}_v$, $\hat{\Theta}_u$, with $\hat{\Theta}_v = \hat{\Theta}_u$. Discard $\hat{\Theta}_u$ and update $\hat{\mathcal{V}} = \hat{\mathcal{V}} \backslash \{u\}$.

Consider the application of Algorithm 1 on the WSN of Fig. 2. Sensor v_1 reports an event by sending packets p_1, p_2, and p_3, which are relayed by v_2 (as p_4, p_5, and p_6) and later by v_3 (as p_7, p_8, and p_9). Traffic is eavesdropped by e_1 and e_2, which create sets $\mathcal{O}_{e_1} = \{tag_{e_1}(p_1), \ldots, tag_{e_1}(p_6)\}$ and $\mathcal{O}_{e_2} = \{tag_{e_2}(p_4), \ldots, tag_{e_2}(p_9)\}$.

In Step 1, label u_1 is associated to $\hat{\Theta}_{u_1} = \mathcal{O}_{e_1}$ and u_2 to $\hat{\Theta}_{u_4} = \mathcal{O}_{e_2}$. Steps 2 and 3 define new labels for e_1, based on the tag intersection and difference of $\hat{\Theta}_{u_1}$ and $\mathcal{O}_{e_2}$.

$$\hat{\Theta}_{u_3} = \{tag(p_1), tag(p_2), tag(p_3)\}, \quad \hat{\ell}_{u_3} = \hat{\ell}_{u_1} \cap C_{e_2},$$
$$\hat{\Theta}_{u_4} = \{tag(p_4), tag(p_5), tag(p_6)\}, \quad \hat{\ell}_{u_4} = \hat{\ell}_{u_1} \backslash C_{e_2}.$$

Similarly for e_2,

$$\hat{\Theta}_{u_5} = \{tag(p_4), tag(p_5), tag(p_6)\}, \quad \hat{\ell}_{u_5} = \hat{\ell}_{u_2} \cap C_{e_1}$$
$$\hat{\Theta}_{u_6} = \{tag(p_7), tag(p_8), tag(p_9)\}, \quad \hat{\ell}_{u_6} = \hat{\ell}_{u_2} \backslash C_{e_1}.$$

Also, $NS(e_1) = \{u_3, u_4\}$ and $NS(e_2) = \{u_5, u_6\}$. In Step 4 the observation and sensor sets are consolidated as:

$$\mathcal{O} = \hat{\Theta}_{u_2} \cup \hat{\Theta}_{u_3} \cup \hat{\Theta}_{u_5} \cup \hat{\Theta} u_6, \text{ and } \hat{\mathcal{V}} = \{u_3, u_4, u_5, u_6\}.$$

Finally, in Step 5, $\hat{\Theta}_{u_5}$ and u_5 are eliminated from $\mathcal{O}$ and $\hat{\mathcal{V}}$, respectively, since they are duplicates of u_4. This reduces $\hat{\mathcal{V}}$ to $\{u_3, u_4, u_6\}$. Note that sets $\hat{\Theta}_{u_3}$, $\hat{\Theta}_{u_4}$, and $\hat{\Theta}_{u_6}$ form a partition of $\mathcal{O}$. Also in this scenario, e_2 cannot distinguish between v_3's and v_4's transmissions.

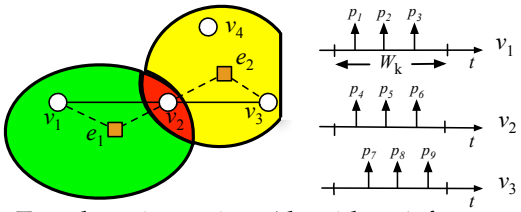

Fig. 2: Tag cleansing using Algorithm 1 for two eavesdroppers with heterogeneous reception areas.

4.2 Contextual Information Inference

In the second stage, the adversary performs timing analysis on $\mathcal{O}$. The adversary takes advantage of the bursty nature of traffic in event-driven WSNs to link traffic streams with physical events. We organize tags in $\mathcal{O}$ into disjoint sets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots$, where $\mathcal{Y}_j$ is attributed to event Ψ_j ($j = 1, 2, \ldots$). The division depends on the temporal and spatial tag correlation. For instance, consider packets p_1 and p_2, from v and u in $\hat{\mathcal{V}}$. These packets are assigned to the same event if $|t(p_1) - t(p_2)|$ is between certain bounds dependent on the distance between u and v. Details are given in Algorithm 2.

Algorithm 2: Event Filtering

Step 1: Sort $\mathcal{O}$ in ascending order according to the tag timestamps.
Step 2: Associate two consecutive packets p_1 and p_2 of $\mathcal{O}$, from sensor labels u and v, with the same set $\mathcal{Y}_j$ if

$$\beta_l(d_{\min}(\hat{\ell}_u, \hat{\ell}_v)) < t(p_2) - t(p_1) < \beta_h(d_{\max}(\hat{\ell}_u, \hat{\ell}_v)).$$

where $\beta_l(d_{\min}(\hat{\ell}_u, \hat{\ell}_v))$ and $\beta_h(d_{\max}(\hat{\ell}_u, \hat{\ell}_v))$ are lower and upper bounds, depending on the minimum and maximum distance between areas $\hat{\ell}_u$ and $\hat{\ell}_v$, respectively.
Step 3: Otherwise, associate p_1 with $\mathcal{Y}_j$ and p_2 with $\mathcal{Y}_{j+1}$.
Step 4: Associate tags in set $\mathcal{Y}_j$ to event Ψ_j.

Thresholds $\beta_l(d_{\min}(\hat{\ell}_u, \hat{\ell}_v))$ and $\beta_h(d_{\max}(\hat{\ell}_u, \hat{\ell}_v))$ reflect bounds on the minimum and maximum delays for relaying packets from $\hat{\ell}_u$ to $\hat{\ell}_v$. The bounds are calculated as a function of the minimum and maximum distance between two areas measured in hops and the per-hop relay delay. For instance, the lower bound is defined as

$$\beta_l(d_{\min}(\hat{\ell}_u, \hat{\ell}_v)) = T_h \left\lfloor \frac{d_{\min}(\hat{\ell}_u, \hat{\ell}_v)}{\gamma} \right\rfloor,$$

where γ is the sensor communication radius and T_h is an estimate of the packet transmission delay between two hops. The lower threshold prevents false association of tags recorded at distant parts of the WSN, due to event concurrence. Similarly, the upper bound is defined as

$$\beta_h(d_{\max}(\hat{\ell}_u, \hat{\ell}_v)) = T_h \left\lfloor \frac{d_{\max}(\hat{\ell}_u, \hat{\ell}_v)}{\gamma} \right\rfloor.$$

The upper bound associates transmissions that occur close in time and in space with the same event. Both thresholds were chosen assuming dense deployments in which paths can be approximated by straight lines.

Algorithm 2 outputs sets of the type $\mathcal{Y}_j = \hat{\Theta}_{u_1} \cup \ldots \cup \hat{\Theta}_{u_y}$ for y labels in $\hat{\mathcal{V}}$. The set $\mathcal{Y}_j$ is used for the inference of event Ψ_j. Moreover, the number of tags in each set $\hat{\Theta}_{u_i} \in \mathcal{Y}_j$ identifies the number of packets transmitted by u_i. Sensors that relay an event report over W should transmit the same number of packets (or approximately the same, if packet retransmissions are present). Thus, the adversary can also obtain an estimate of the number of packets x triggered by event Ψ_j. The accuracy of this estimation depends on the number of sensors associated with each label. For example, in Fig. 2, after applying Algorithm 1, the adversary obtains $\hat{\mathcal{V}} = \{u_2, u_3, u_6\}$, and assigns the same label to v_3 and v_4.

The adversary can utilize the size of each $\hat{\Theta}_{u_i} \in \mathcal{Y}_j$ to estimate the number of sensors associated with each label and the number of packets x that report Ψ_λ. The latter is estimated as the size of the smallest transmission set in $\mathcal{Y}_j$. Once the number of sensors per label is found, a topology approximation is obtained by establishing links between labels. Details are given in Algorithm 3.

Algorithm 3: Topology Approximation

Step 1: Let $c_1, c_2, \ldots$ be counters associated with each label v in Y_j, where $c_v = |\hat{\Theta}_v|$. Estimate the number of packets sent by the source to report Ψ_λ as

$$\hat{x} = \min(c_1, c_2, \ldots).$$

Step 2: Estimate the number of sensors $\hat{n}_v$ associated with label v and counter c_v as $\hat{n}_v = \left\lfloor \frac{c_v}{\hat{x}} \right\rfloor$.
Step 3: Establish a link (u, v) between labels u and v if

$$d_{\min}(\hat{\ell}_u, \hat{\ell}_v) \leq \gamma.$$

The value of $\hat{n}_v$, as estimated in Step 2, is incorrect if there are inactive sensors in the label area (e.g., v_4 in Fig. 2), or if a single sensor concurrently relays traffic of more than one events. Finally, based on sets $\mathcal{Y}_j$, the adversary can infer the source and sink location, the routing path to the sink, and the event's occurrence time associated with event Ψ_j, using Algorithm 4.

Algorithm 4: Contextual Information Inference

Step 1: Estimate the event location for Ψ_j as $\hat{\ell}_{v^*}$, where:

$$v^* = \arg\min_{v \in \hat{V}} \{t(p_i) : tag(p_i) \in \mathcal{Y}_j\}.$$

The event location of Ψ_j is estimated to be the area of the tag with the earliest transmission time in $\mathcal{Y}_j$.
Step 2: The occurrence time for $\mathcal{Y}_j$ is estimated as

$$\hat{t}(\mathcal{Y}_j) = \min_{v \in \hat{V}} \{t(p_i) : tag(p_i) \in \mathcal{Y}_j\},$$

i.e., the earliest transmission time recorded in $\mathcal{Y}_j$.
Step 3: The path $p(v, s)$ from the source v to the sink s is estimated as the label sequence of tags in $\mathcal{Y}_j$ (sorted in ascending order, based on transmission time).
Step 4: The sink location is estimated to the location area $\hat{\ell}_s$ of the last label in path $p(v, s)$.

The sink location estimation can be iteratively improved when multiple events are reported to the sink.

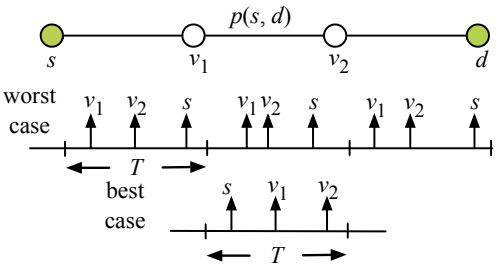

Fig. 3: Randomization of traffic patterns.

5 Efficient Traffic Normalization

To counter traffic analysis, most existing solutions *introduce bogus traffic at every sensor* [4], [20], [23], [29]. This is because all sensors are potential sources and the eavesdroppers' locations are unknown. Moreover, the normalized traffic patterns can lead to the accumulation of packet delay on a per-hop basis. For instance, consider the path $p(s,d)$ shown in Fig. 3. Assume that the traffic rate of every sensor is normalized to one packet per T. The worst-case forwarding delay is equal to $|p(s,d)|T$, where $|p(s,d)|$ is the path length in hops. This delay occurs when downstream sensors transmit earlier than upstream ones within each interval. In the best case, the forwarding delay reduces to T, when upstream sensors transmit earlier than downstream. Proposition 1 shows that a packet will be forwarded over less than two hops per T, on average.

Proposition 1: When sensors transmit one packet uniformly per interval T, the average number of hops that a packet can traverse per T is 1.72.

Proof: The proof is provided in Appendix A. □

To address the inefficiencies of prior traffic normalization methods, we first reduce the number of bogus traffic sources. To do so, we divide time into epochs and partition the set of sensors $\mathcal{V}$ into subsets. Only one subset is active at a given epoch, and subsets are periodically rotated in a round-robin fashion. A sensor is allowed to send traffic (bogus or real) only if a subset it belongs to is active. Each subset forms a subgraph designed to satisfy the following properties: (a) it is connected, (b) it can deliver packets to any vertex of the original graph, and (c) the subgraph size is minimal.

Properties (a) and (b) guarantee that an active subgraph can deliver a real packet to the sink. Moreover, the sink location remains hidden because all sensors can be reached by the subgraph. Finally, property (c) minimizes the number of active bogus traffic sources per epoch required to satisfy properties (a) and (b). For instance, Fig. 4 shows the partition of a small WSN into $\mathcal{D}_1 = \{v_1, v_2, v_3\}$ and $\mathcal{D}_2 = \{v_4, v_5, v_6, v_7, v_8\}$ that satisfy properties (a)-(c). Nodes in both $\mathcal{D}_1$ and $\mathcal{D}_2$ can deliver a packet to any node in $\mathcal{V}$. When $\mathcal{D}_1$ becomes active, v_1, v_2 and v_3 have a routing path to the sink v_6. To further reduce the forwarding delay, we loosely coordinate sensor transmissions based on tree structures. Our traffic normalization scheme consist of a network partition and a transmission coordination phase.

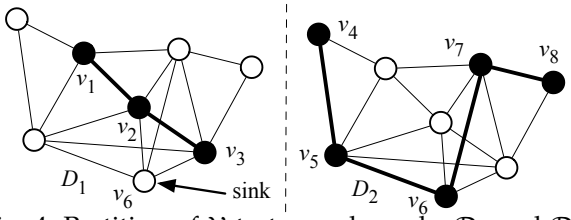

Fig. 4: Partition of $\mathcal{V}$ to two subgraphs $\mathcal{D}_1$ and $\mathcal{D}_2$.

5.1 Network Partition Phase

In the network partition phase, we partition $\mathcal{V}$ into subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_z\}$, which are activated in a round-robin fashion. Sensors of an active subset transmit dummy packets at a fixed packet rate. The dummy packets are replaced with real ones, when a sensor of an active subset reports an event. To satisfy design properties (a)-(c), we reduce the problem of partitioning $\mathcal{V}$ to the problem of finding a partition of connected dominating sets (CDSs). A CDS is formally defined as follows [11].

Definition 4 (Connected Dominating Set): For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a subset $\mathcal{D} \subseteq \mathcal{V}$ is a dominating set (DS) if any vertex $v \in \mathcal{V}$ either belongs to $\mathcal{D}$, or is adjacent (within one hop) to some vertex in $\mathcal{D}$. If $\mathcal{D}$ induces a connected subgraph on $\mathcal{G}$, then $\mathcal{D}$ is a connected dominating set (CDS). Moreover, the CDS with the smallest cardinality is called a minimum connected dominating set (MCDS).

The partition of $\mathcal{V}$ to disjoint MCDSs satisfies properties (a)-(c). Property (a) is satisfied, as the set of MCDSs spans $\mathcal{V}$. Hence, each sensor can transmit real traffic when its CDS becomes active. By design, the traffic pattern of an active sensor is not altered when dummy packets are replaced by real ones. For property (b), a CDS ensures that any sensor in $\mathcal{V}$ is either part of D_j or within one hop from a sensor in D_j. Moreover D_j forms a connected graph. Hence, a real packet transmitted by a sensor in D_j can be forwarded to any sensor in $\mathcal{V}$ using only D_j. Finally by definition, an MCDS minimizes the size of each subgraph. However, we note that MCDSs do not necessarily include shortest paths to the sink. This could increase the forwarding delay. We further investigate the construction of CDSs that contain the shortest paths from any CDS sensor to the sink.

5.1.1 Network Partition–Sets of Minimum Size

We first consider the partition of $\mathcal{V}$ into MCDSs. Such a partition is not guaranteed to exist for arbitrary graphs (e.g., a topology with a minimum vertex cut of one). Moreover, the problem of computing a single MCDS is NP-complete [9]. To address these limitations, we relax the partition requirement and allow nodes to be part of more than one MCDSs. We denote the *appearance frequency* of node v to MCDSs as $f(v)$. We propose a heuristic algorithm that computes an approximation of $\mathcal{V}$'s partition by balancng between the appearance frequency, the number of MCDSs that span $\mathcal{V}$, and the MCDS size. These parameters are used to control the tradeoff between the end-to-end delay and the communication overhead. Note that, if $\mathcal{V}$ is partitioned to a

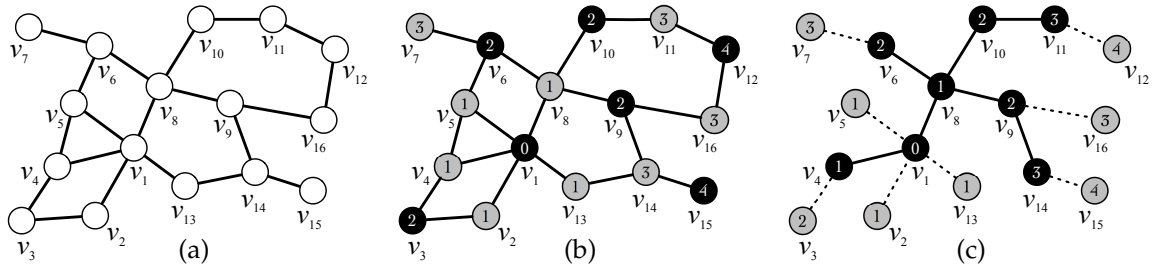

Fig. 5: (a) Original WSN graph, (b) a DS generated in Stage 1, (c) a MCDS approximation generated in Stage 2.

large number of MCDSs, the number of epochs until each MCDS becomes active increases, thus increasing the end-to-end delay. However, a small number of MCDSs results in larger communication overhead because more sensors are active per epoch.

Algorithm 5: MCDS approximation–We generate a CDS partition in three stages. In Stage 1, we construct a minimum DS based on a well-known approximation (the problem of computing a minimum DS is also NP-complete [9]). In Stage 2, we connect the DS to generate a CDS. The nodes selected to connect the DS minimize the CDS size in a greedy fashion. In Stage 3, we repeat stages 1 and 2 to obtain a partition of $\mathcal{V}$ to CDSs.

For each $v \in \mathcal{V}$, let $m(v)$ be a marker, which can take the values *white*, *black*, or *gray*. $\mathcal{N}_v^k$ and $[\mathcal{N}_v^k] = \mathcal{N}_v^k \cup v$ are v's open and closed k-hop neighborhoods, respectively. Let $\delta(v) = |\mathcal{N}_v|$ be the degree of v, and $\delta^*(v)$ the effective degree of v. The latter is defined as the number of v's neighbors marked as *white*. Let $r(v)$ be the rank of v, defined as the order that v changed its marker relative to a leader node. Finally, $\rho(v)$ is the dominator node of v. Initially, the appearance frequency of every node is set to $f(v) = 0$. We base the marking process for generating a DS to the algorithm presented in [13].

Stage 1: DS generation

Step 1: Each $v \in \mathcal{V}$ initializes and broadcasts the values of $m(v) =$ *white*, $\delta^*(v) = \delta(v)$, and $r(v) = 0$.
Step 2: A randomly chosen leader s sets $m(s) =$ *black* and broadcasts $m(s), r(s)$, and $f(s)$ to $\mathcal{N}_s$.
Step 3: A *white* node u receiving $m(v) =$ *black* is dominated by v, sets $m(u) =$ *gray*, $\rho(u) = v$, and $r(u) = r(v) + 1$. It then broadcasts $m(u)$ and $r(u)$ to $\mathcal{N}_u$.
Step 4: A *white* node v receiving $m(u) =$ *gray* from $u \in \mathcal{N}_v$, decreases $\delta^*(v)$ by one, updates $r(v) = r(u) + 1$ if $r(v) \leq r(u)$, and broadcasts $\delta^*(v)$ and $r(v)$ to $\mathcal{N}_v$.
Step 5: A *white* node v changes $m(v)$ to *black*, if

$$v = \arg\max_{u \in [\mathcal{N}_v]} \left\{ \frac{\delta^*(u)}{\delta^*_{\max}(v)} \times \frac{1}{f(u)+1} \right\}, \quad (1)$$

where $\delta^*_{\max}(v) = \max_{u \in [\mathcal{N}_v]} \delta^*(u)$. Ties are broken arbitrarily. Node v becomes a "dominator" and broadcasts $m(v) =$ *black* and $r(v)$ to $\mathcal{N}_u$.
Step 6: Repeat Steps 3-5 until all nodes are marked as *black* (dominator) or *gray* (dominated).

With the termination of stage 1, the set of *black* nodes forms a DS. Note that, the domination metric in Eq. (1) tradeoffs between two competing factors: (a) the CDS size and (b) the number of CDSs in the partition of $\mathcal{V}$. By maximizing $\frac{\delta^*(v)}{\delta^*_{\max}(v)}$, we include in the DS nodes that dominate the largest fraction of their neighbors within their closed neighborhood. Therefore, the size of the DS is reduced in a greedy fashion. On the other hand, the factor $\frac{1}{f(v)+1}$ favors the selection of nodes with the lowest appearance frequency. Because $\frac{\delta(v)}{\delta_{\max}(v)} \leq 1$, the factor $\frac{1}{f(v)+1}$ guarantees that every node will be part of a CDS, and thus Algorithm 4 terminates. In Fig. 5(b), we show the DS generated during Stage 1 for the graph of Fig. 5(a). The color and rank of the nodes is also shown.

In Stage 2, we approximate the MCDS by adding *gray* nodes that connect the greatest number of *black* nodes. Let $b(v)$ be the number of higher ranked *black* neighbors of v.

Stage 2: MCDS Approximation

Step 1: Each *gray* node $v \in \mathcal{V}$ broadcasts $b(v)$ to $\mathcal{N}_v^2$ **Step 2:** Starting with the leader's neighborhood, a *gray* node v becomes *black* if,

$$v = \arg\max_{u \in Z} \left\{ \frac{b(u)}{b_{\max}(v)} \times \frac{1}{f(u)+1} \right\}, \quad (2)$$

where $Z = \{u : u \in [\mathcal{N}_v^2], r(u) = r(v)\}$, $b_{\max}(v) = \max_{\{u \in [\mathcal{N}_v^2], m(u) = gray\}} b(v)$, and $b(u), b_{max}(v) > 0$. Node v broadcasts $m(v)$ in $\mathcal{N}_v^2$. Ties are broken arbitrarily.
Step 3: A node $w \in \mathcal{N}_u$ overhearing the change of u's marker from *gray* to *black*, with $m(w) =$ *black* and $r(w) = r(u) + 1$ sets and broadcasts $\rho(w) = u$ to $\mathcal{N}_w$.
Step 4: A *gray* node u overhearing $\rho(w)$ from a *black* node w with $r(u) = r(w) + 1$ broadcasts $b(u) = b(u) - 1$ to $\mathcal{N}_u^2$.
Step 5: Steps 2-4 are iterated for all *black* nodes in the DS, until all *gray* nodes have a $b(v)$ value equal to zero.
Step 6 (Pruning): If a *black* node v with $f(v) > 0$ does not dominate at least one *gray*, it changes $m(v) =$ *gray*.

After the execution of Step 5, each *gray* node has a $b(v) = 0$, thus all *black* nodes of Stage 1 are dominated. Moreover, these nodes are dominated by a *gray* node of lower rank, that turns *black* in Step 3. Since the process is initiated in the leader's neighborhood, with the change of a *gray* node into *black*, each dominated *black* node is connected to the leader. Therefore, with the termination of Stage 2, all *black* nodes have a path to the leader, forming a CDS. Similarly to Stage 1, the metric used in Eq. 2 tradeoffs between the CDS size and the number of CDS in the partition. Maximizing the fraction $\frac{b(v)}{b_{\max}(v)}$

benefits the nodes that connect the highest number of *black* nodes in the DS, while $\frac{1}{f(v)+1}$ favors the selection of nodes with the lowest appearance frequency. Finally, in the pruning step, we eliminate all black nodes that do not dominate any gray nodes, provided that these black nodes have appeared at least at one CDS ($f(v) > 0$).

Fig. 5(c) shows the CDS generated in Stage 2 from the DS of Fig. 5(b). In the final stage, the MCDS generation process is iteratively applied until all sensors become part of one MCDS.

Stage 3: MCDS Update

Step 1: Increment $f(v)$ by one unit for all nodes in $\mathcal{D}_j$.
Step 2: Repeat Stages 1 and 2 until $f(v) > 0,\ \forall v \in \mathcal{V}$.

Proposition 2: Stage 3 terminates in at most $\delta_{\max} + 1$ iterations, where $\delta_{\max} = \max\{\delta(v) : \forall v \in \mathcal{V}\}$.

Proof: The proof is provided in Appendix B. □

5.1.2 Network Partition–CDSs with Shortest Paths

We now consider the problem of partitioning $\mathcal{V}$ to CDSs that contain the shortest paths from any CDS node to the sink. We call this CDS type as a Single-destination Shortest-path CDS (SS-CDS), defined it as follows.

Definition 5 (Single-destination Shortest-path CDS): Let $p(s, \mu)$ be the shortest path between s and μ in $\mathcal{G}$. Let also μ be a unique destination (sink). Set $\mathcal{D} \subseteq \mathcal{V}$ is a single-destination shortest-path CDS if for each $s \in \mathcal{D}$, $p(s, \mu)$ belongs to $\mathcal{D}$. The set $\mathcal{D}$ with the smallest cardinality is called a single-destination shortest-path minimum connected dominating set (SS-MCDS).

We first show that the problem of constructing SS-MCDSs is NP-complete by reducing it to the Minimum Shortest-path Steiner arborescence problem.

Proposition 3: The problem of finding an SS-MCDS in arbitrary graphs is NP-complete.

Proof: The proof is provided in Appendix C. □

We propose a heuristic algorithm that approximates the SS-MCDS in a greedy fashion and balances between the appearance frequency of nodes in an SS-MCDS and the number of SS-MCDSs that partition $\mathcal{V}$. We now describe our algorithm in detail.

Algorithm 6: SS-MCDS approximation–We modify Algorithm 5 to generate an SS-MCDS partition of $\mathcal{V}$. In Algorithm 6, we run stage 1 of Algorithm 5 to generate a DS, forcing the sink to be the leader. Then, we modify Stage 2 of Algorithm 5 to restrict the selection of "bridge nodes" (*gray* nodes turning *black* to connect DS nodes) to those found in the shortest path to the sink. Initially, for each $v \in \mathcal{V}$ we set the appearance frequency to $f(v) = 0$.

Algorithm 6: SS-MCDS approximation

Step 1: Set the leader node s to the sink μ and execute stage 1 of Algorithm 5 to generate a DS.
Step 2: Each $v \in \mathcal{V}$ with $m(v) = gray$ broadcasts $b(v)$ to $\mathcal{N}_v^2$, where $b(v)$ is the number of *black* neighbors with hop count $|p(v, s)|+1$ (at least one such neighbor exists).
Step 3: A *gray* node v becomes *black* if,

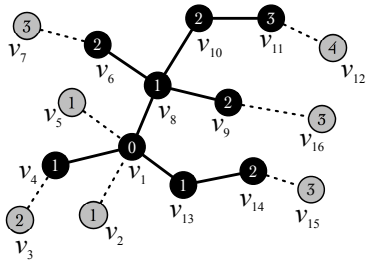

Fig. 6: SS-MCDS obtained by Algorithm 6.

$$v = \arg\max_{u \in Z} \left\{ \frac{b(u)}{b_{\max}(v)} \times \frac{1}{f(u)+1} \right\}, \tag{3}$$

where $Z = \{u : u \in \mathcal{N}_v, |p(u, s)| = |p(v, s)|, m(u) = gray\}$, $b_{\max}(v) = \max_{\{u \in [\mathcal{N}_v^2], m(u)=gray\}} b(u)$, and $b(v), b_{max}(v) > 0$. Node v broadcast $m(v)$ in $\mathcal{N}_v^2$. Ties are broken arbitrarily. If $|p(w, s)| \geq |p(v, s)|$ for $w = \rho(v)$, node v sets $\rho(v) = nill$ and broadcasts it.
Step 4: A node $w \in \mathcal{N}_u$ overhearing the change of marker of u, with $m(w) = black$ and $r(w) = r(u) + 1$ sets and broadcasts $\rho(w) = u$ to $\mathcal{N}_w$. If $m(w) = gray$, $r(w) = r(u) - 1$ and $\rho(u) = nill$, w increases $b(w)$ by one.
Step 5: A *gray* node u overhearing $\rho(w)$ from a *black* node w with $r(u) = r(w) + 1$ broadcast $b(u) = b(u) - 1$ to $\mathcal{N}_v^2$.
Step 6: Steps 3-5 are iterated for all *black* nodes in the DS, until all *gray* nodes set $b(v) = 0$.
Step 7 (Pruning): If a *black* node v with $f(v) > 0$ does not dominate at least one *gray*, it sets $m(v) = gray$.
Step 8: Run Stage 3 of Algorithm 5 until $\mathcal{V}$ is partitioned.

With the termination of Algorithm 6, the set of black nodes forms a SS-CDS. To ensure that the shortest paths to the sink are included, the sink is chosen as the leader in the DS generation. Moreover, a *black* node v with hop-count to the sink $|p(v, s)|$ can only be dominated by a *gray* node with hop-count of $|p(v, s)|-1$ (Steps 2 and 3 of Algorithm 6). After Step 6, all *black* nodes are dominated, and are connected to the sink via the shortest path. The application of Algorithm 6 on the topology of Fig. 5(a) results in the SS-MCDS shown in Fig. 6.

5.1.3 Message Complexity Analysis

In this section, we analyze the message complexity for partitioning the WSN to MCDS and SS-MCDSs (algorithms 5 and 6, respectively).

Proposition 4: The message complexity for partitioning the WSN to MCDSs using Algorithm 5 is $O(\delta_{\max}^3 |\mathcal{V}|)$. Partitioning the WSN to SS-MCDSs (Algorithm 6) yields the same complexity.

Proof: The proof is provided in Appendix D. □

We observe that algorithms 5 and 6 have linear message complexity to the size of the WSN. The network partition overhead is of the same order as the recurring overhead for normalizing traffic patterns. The WSN has to transmit $|\mathcal{V}|$ bogus messages periodically to normalize the traffic patterns at each sensor, whereas the WSN partition to subgraphs has to be applied only once.

5.1.4 Privacy Analysis

In this section, we analyze the privacy achieved by the MCDS partition. This analysis is performed assuming that the adversary is fully aware of the application of the MCDS partition, the MCDS rotation, and the normalization of the traffic in active sensors. Let an event Ψ occur at time $t(\Psi) \in W$ and be reported by a sensor $v \in D_i$ who is located at ℓ_v.

Source location and occurrence time privacy: To report Ψ, sensor v replaces dummy packets with real ones, while maintaining its transmission schedule. Note that real packets are indistinguishable from dummy ones due to the application of per-hop packet re-encryption. Downstream sensors receiving v's report continue to forward it by substituting dummy packets with real ones. By applying Algorithm 1, the eavesdropper can reduce the locations of the dummy transmissions to location approximation areas of the sensors in $\mathcal{D}_i$. However, events cannot be meaningfully distinguished by the application of Algorithm 2. Moreover, the set of candidate sources cannot be reduced below the set of sensors in $\mathcal{D}_i$.

The partition of the observation set $\mathcal{O}(W)$ to disjoint sets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots$ according to Algorithm 2 depends on the selection of the bound functions β_ℓ and β_h. Selecting a small upper bound to model the immediate relay of real packets practically divides $\mathcal{O}(W)$ to disjoint sets containing one or very few tags. Every sensor transmission in $\mathcal{O}(W)$ is assumed to be a distinct event, thus hiding the source location and occurrence time of the real event Ψ. If the adversary loosens the upper bound to account for the traffic normalization applied by the CDS sensors, $\mathcal{O}(W)$ is divided to few sets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots$ with large cardinality, which are mapped to events $\Psi_1, \Psi_2, \ldots$. The adversary cannot identify which set contains the real event. Moreover, the source location and time occurrence of the first tag in the set that contains $t(\Psi)$, is uncorrelated to $t(\Psi)$, but depends on the random transmission times of each sensor. We note that the adversary could apply other statistical analysis methods, such as those reported in [20], [23]. These methods fail to detect Ψ, since the transmission patterns of sensors in $\mathcal{D}_i$ do not change when real traffic is introduced.

Sink location privacy: After all CDSs have been active, the adversary can apply Algorithm 3 to approximate the topology of each $\mathcal{D}_i$ and obtain an estimate $\hat{\mathcal{V}}$ of $\mathcal{V}$. When $\mathcal{V}$ is partitioned to MCDSs, the MCDS structure is unrelated to the location of the sink and hence, the sink location privacy is protected. When $\mathcal{V}$ is partitioned to SS-MCDSs, the adversary can take advantage of the shortest-path property and localize the sink. This can be achieved by Algorithm 7. Let, $p^*(u,v)$ and $p(u,v)$ be the shortest path between u and v in $\mathcal{D}_i$ and $\hat{\mathcal{V}}$, respectively.

Algorithm 7: Sink location inference

Step 1: For each $u \in \mathcal{D}_i$, obtain paths $p(u,v)$ and $p^*(u,v)$ to each $v \in \mathcal{D}_i$. Obtain the cumulative path difference as

$$\Delta_u^i = \sum_{v \in \mathcal{D}_i} |p^*(u,v)| - |p(u,v)|$$

Step 2: Repeat Step 1 for each $\mathcal{D}_i$, and calculate the average cumulative path difference as,

$$\Delta_u = \frac{1}{f(u)} \sum_{u \in D_i} \Delta_u^i$$

Step 3: Identify the sink as $\hat{\mu} = \arg\min_{u \in \hat{\mathcal{V}}} \Delta_u$.

The intuition behind Algorithm 7 is to exhaustively test every $u \in \hat{\mathcal{V}}$ as a candidate sink by computing the average cumulative path difference Δ_u. A low difference indicates that u can be reached by all nodes in the $\mathcal{D}_i$'s it belongs to via shortest paths. Since each SS-MCDS is constructed to contain the shortest paths to the sink, the sensor $\hat{\mu}$ with the lowest score over all the sets in the partition is considered to be the sink. Thus, the sink location privacy distance is the average distance between $\ell(\hat{\mu})$ and $\ell(\mu)$. The location of the sink can be hidden by selecting a sensor near the sink to serve as the unique destination of the SS-MCDS. The location of the fake sink can be selected to satisfy any desired privacy distance.

5.1.5 Privacy and Retransmissions due to Packet Loss

In realistic conditions, packets could be corrupted due to fading or noise, leading to retransmissions. If only real packets are retransmitted, the privacy of events could be breached. To remedy the effect of retransmissions on privacy, we adopt an implicit acknowledgment scheme. Assuming that a communication link between v and u is bidirectional, when v forwards a packet p to u, sensor v can receive an implicit acknowledgement when u forwards the real packet p downstream. If v does not overhear the transmission of p by u (because it got corrupted by the channel), it repeats the transmission of p by following its normalized transmission schedule. For scheme correctness, the sink must transmit the real packet it received to acknowledge it to the last relay.

5.2 Transmission Coordination

As we showed in Proposition 1, when sensors transmit in an uncoordinated fashion, the forwarding delay for reporting a real event can increase significantly. This is because a downstream sensor may transit earlier than an upstream one on a given interval. To reduce the forwarding delay we schedule sensors to transmit relative to their depth in the CDS tree, when the tree is assumed to be rooted at the sink. For an interval T, if downstream nodes are scheduled to transmit after upstream ones, a real transmission is guaranteed to reach the sink within T. However, this simple scheme reveals the sink location due to the use of the sink as the tree root. To preserve the sink location privacy, we propose the direction-free assignment scheme (DFAS) that guarantees the traversal of a minimum number of hops per T, while hiding the traffic directionality and the sink location.

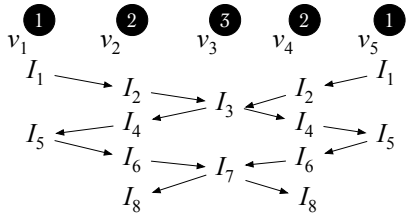

Fig. 7: DFAS assignment for two subpaths ($h = 3, \kappa = 1$).

5.2.1 Direction-Free Assignment Scheme (DFAS)

Consider a CDS $\mathcal{D}_i$. We first divide $\mathcal{D}_i$ into several subpaths. Let h be a control parameter of the subpath length and κ a control factor of the packet rate of each node. Our algorithm uses h and κ to compute the transmission intervals for each node in $\mathcal{D}_i$.

Algorithm 8: Direction-Free Assignment Scheme (DFAS)

Step 1: Epoch W that $\mathcal{D}_i$ remains active is divided into sub-epochs $I_1, I_2, \ldots, I_{\kappa \times S}$, where $S = 4h - 4$.
Step 2: Randomly select a node μ' as the pseudo-sink.
Step 3: A node v is labeled with,

$$id_v = \begin{cases} q + 1, & \text{if } q < h \\ 2h - q + 1, & \text{if } q \geq h \end{cases}$$

where $q = \mod(|p(v, \mu')|, 2h - 2)$.
Step 4: If $id_v = 1, h$, node v with id_v transmits at random in sub-epochs

$$I_{id_v + qS}, I_{2h + id_v - 2 + qS},$$

if $2 \leq id_v \leq h - 1$ in transmits in subepochs

$$I_{id_v + qS}, I_{2h - id_v + qS}, I_{2h + id_v - 2 + qS}, I_{4h - id_v - 2 + qS}.$$

In DFAS, we have assumed that $h > 1$. If $h = 1$, the transmission assignment degenerates to uncoordinated transmissions within W. To demonstrate the operation of DFAS, consider the example of Fig. 7 where $h = 3$ and $\kappa = 1$. In Step 1, epoch W is divided into 8 sub-epochs. In Step 2, v_1 is selected as the pseudo-sink. In Step 3, we label the network as a tree rooted at v_1, and obtain $id_{v_1} = 1$, $id_{v_2} = 2$, etc. In Step 4, sensors are assigned to transmit at random within the designated sub-epochs, according to their ids. Sensors with ids 1 and 3 transmit two packets per epoch, while all other sensors transmit four packets per epoch. As shown in the privacy analysis, the symmetry in this assignment hides the traffic direction. We now show that DFAS guarantees $2h$ relay operations per any $(4h - 4)$ sub-epochs.

Proposition 5: In DFAS, a packet is guaranteed to be forwarded $2h$ hops per $(4h - 4)$ sub-epochs, irrespective of the flow direction and the origin sensor.

Proof: The proof is provided in Appendix E. □

5.2.2 Privacy Analysis

Assume event Ψ's occurrence at $t(\Psi) \in W$, while CDS $\mathcal{D}_i$ is active. Let Ψ be sensed by $v \in \mathcal{D}_i$ located at ℓ_v.

Source location and occurrence time privacy: The DFAS coordination is applied irrespective of the occurrence of an event. Sensor v and all sensors downstream to the sink, will replace the dummy transmissions scheduled at different sub-epochs with real packets. Hence, the occurrence time of Ψ is concealed. Moreover, DFAS maintains the same source location privacy as the uncoordinated case. Without an estimate of $t(\Psi)$, the adversary cannot localize v.

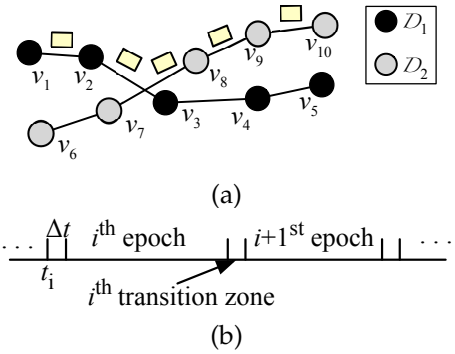

Fig. 8: (a) The MCFS operation, (b) the transition zone between epochs.

Sink location privacy: Assume that the adversary observes traffic during $4h - 4$ sub-epochs of a single interval ($\kappa = 1$). According to the transmission assignment of DFAS, nodes transmit either 2 or 4 packets within this interval. The packet rate of each node could by used to identify the node positions within subpaths and consequently, the labeling of DFAS. For instance, in Fig. 7, the adversary intercepts two transmissions from v_1, four transmissions from v_2, two from v_3, and so on. He can infer that $h = 3$ and discover the subpath labeling. The subpath labeling reduces the candidate nodes that could have been used as the pseudo-sink to nodes with $id = 1$ or $id = h$. However, the sink location remains hidden due to the random selection of the pseudo-sink.

We emphasize that the coordination imposed by DFAS conceals the traffic direction. Consider the transmission assignment shown in Fig. 7. Without loss of generality, assume that a packet m is relayed by v_1 in sub-epoch I_5, v_2 in I_6, and v_3 in I_7. However, the transmissions intercepted during these sub-epochs could be associated with packet relays in the opposite direction. The transmission in I_5 could be due to the relay of an m' by v_3, v_2, and v_1 in I_3, I_4, and I_5, respectively. Similarly, the transmission in I_7 could be due to the relay of an m' by v_5, v_4, and v_3, in I_5, I_6, and I_7, respectively. The symmetry in the assignment of the transmission sub-epochs over subpaths hides the traffic direction.

5.3 Routing Over Multiple CDSs

CDSs are rotated periodically per epoch to allow all sensors report events to the sink. A real packet m that originated from $v \in \mathcal{D}_i$, may be in transit while another CDS $\mathcal{D}_j$ becomes active. The CDS property guarantees that at least one node in $\mathcal{D}_j$ would overhear the last relay of m by a node in $\mathcal{D}_i$. We develop a simple routing scheme to forward packets over multiple CDSs. Here we assume that $\mathcal{D}_i$ is active during epoch W_k, and $\mathcal{D}_j$ in the

next epoch W_{k+1}. The steps of our scheme are as follows.

Algorithm 9: Multiple CDS Forwarding Scheme (MCFS)

Step 1: A real packet m originating from $v \in \mathcal{D}_i$ at epoch W_k is forwarded to μ via the shortest path $p(v, \mu)$ in $\mathcal{D}_i$.
Step 2: Any $u \in \mathcal{D}_j$ (next active CDS) overhearing m's transmission during W_k's last sub-epoch (i.e., sub-epoch $I_{\kappa \times S}$), forwards m to the sink when $\mathcal{D}_j$ becomes active in W_{k+1}. Nodes in $\mathcal{D}_j$ discard any duplicates of m.

The MCFS operation is depicted in Fig. 8(a). Sensor $v_1 \in \mathcal{D}_1$ sends a packet m to the sink during epoch W_k, using $\mathcal{D}_1$. The CDS is rotated to $\mathcal{D}_2$ while m is in transit. Sensor v_3 is the last one to transmit m in W_k. Sensors v_7 and v_8 overhear v_3's transmission. The relay of m is continued by v_7 and v_8 during W_{k+1}, using $\mathcal{D}_2$. The duplicate packet forwarded by v_7 is discarded at v_8. Packet m is delivered to the sink during $\mathcal{D}_2$.

The MCFS operation impacts the end-to-end delay for delivering a report to the sink in two ways. First, it introduces a *buffering delay* until a CDS that contains the source is activated. This delay depends on the number of CDSs and the frequency of appearance of the source to CDSs. Second, it may increase the forwarding delay. This is because a packet may be forwarded to the sink via multiple CDSs. We analyze the two delays with the following propositions.

Proposition 6: Let a sensor v belong to $f(v) \geq 1$ CDSs. Suppose that an event Ψ is detected by v at time $t(\Psi)$, where $t(\Psi)$ is uniformly distributed over z epochs. The delay until a CDS containing v becomes active is:

1) $d_{\min} = 0$ epochs.
2) $d_{\max} = z - f(v)$ epochs.
3) $d_{ave} = \sum_{k=1}^{z-f(v)} k \times \frac{C(z-k-1, f(v)-1)}{C(z, f(v))}$ epochs.

Proof: The proof is provided in Appendix F. □

Proposition 7: The number of hops traversed by a real packet m originating from v until it reaches the sink μ is upper-bounded by $|p(v, \mu)| + 2rot$, where $|p(v, \mu)|$ is the shortest path length between v and μ and rot is the number of CDS rotations until m reaches μ.

Proof: The proof is provided in Appendix G. □

5.3.1 Synchronization of CDS Rotations

The MCFS is a coordinated action which requires network-wide synchronization to a common time reference. The problem of time synchronization in WSNs has been extensively studied (e.g., [1], [26]). Given the rich literature in this domain, the specific method used for maintaining synchronization is beyond the scope of the present work. We assume that synchronization is maintained for purposes that extend beyond the privacy of contextual information including the implementation of well-known time-slotted protocols at the MAC layer and temporal analysis of sensor data at the sink.

For a maximum synchronization error Δt, the synchronous sensor activation at different epochs can be ensured by incorporating a "transition zone". The concept of a transition zone is demonstrated in Fig. 8(b). Two consecutive epochs i and $i+1$ are separated by a transition zone with a duration equal to Δt. Sensors that were active during the i^{th} epoch remain active (transmitting or receiving) during the i^{th} transition zone, whereas sensors of the following epoch are activated after the i^{th} transition zone has expired. The introduction of a transition zone ensures the following property.

Let the earliest sensor transition to epoch i at t_i. Let each epoch last for T. Because the synchronization error is at most Δt, any sensor active during the i^{th} epoch will transmit before t_i +T+ Δt, which is the expiration time of the i^{th} transition zone. Moreover, no sensor of the $i+1^{st}$ epoch will be active before the i^{th} transition zone is expired. This guarantees the synchronous activation of the sensors that belong to the same CDS.

6 Privacy and Performance Evaluation

In this section, we compare our traffic normalization method with a representative set of prior works. We show that the MCDS approach achieves the same privacy level as a global traffic normalization, but at lower cost. Moreover, we show that techniques designed to thwart local eavesdroppers leak information under global eavesdroppers.

6.1 Privacy Evaluation

We applied the traffic analysis algorithms outlined in Section 4 and measured the privacy distance achieved for different privacy types (source location, sink location, and event occurrence time). We compared our method, referred to as "MCDS", with (a) a base method that does not protect contextual information privacy, (b) the "STaR" scheme [15], (c) "phantom flooding" [22], and (d) the global traffic normalization method in [20].

Simulation setup: We randomly deployed 250 sensors within a 450m×450m area. We also deployed an eavesdropping network on a square grid to achieve homogeneous coverage of the WSN. We varied the square grid size α and the corresponding number of deployed eavesdroppers. The sensor transmission range and the eavesdropper reception range were set to 50m. We abstracted the PHY and MAC layers into a simple per-hop delay model, which consists of a fixed component representing the transmission and propagation delays at the PHY layer, and the contention delay at the MAC layer. This delay was set to 166ms for a packet transmission of 1280 bytes, according to the IEEE 802.15.4 protocol evaluation presented in [8]. No retransmissions due to collisions or impairments of the wireless medium were considered. This simple model was preferred to eliminate the randomness in different system realizations due to contention. Moreover, this model closely matches event-driven networks, which operate under low-contention conditions due to the sparsity of transmissions.

Events were generated at randomly selected locations in the WSNs. The inter-event time followed a uniform distribution in the $[0, 60]$s interval. Events were reported

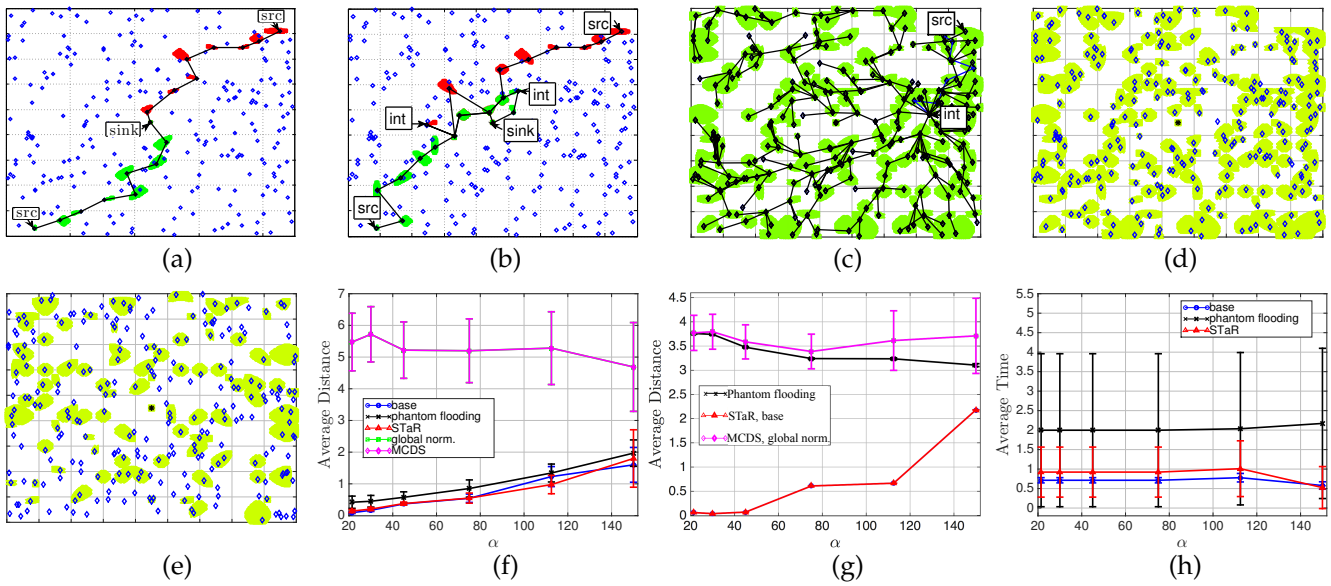

Fig. 9: Inferred sensor activity areas for the following schemes: (a) no protection, (c) STaR, (b) phantom flooding, (d) global norm., (e) MCDS. Privacy distance for: (f) source privacy, (g) sink privacy, and (h) temporal privacy.

by the closest active sensor to the event location. In the MCDS scheme, events were reported by the first active sensor that detected the event. The event is reported by transmitting one packet of 1,280 bytes to the sink.

Visual Representation of Privacy: We visualized the event privacy achieved by each scheme by representing the sensors' true locations as points and the approximated area estimated via traffic analysis as a shaded area around each point. We simulated two events and shaded the detected transmission activity of each event with the same color. Moreover, the positions of various critical nodes such as the source, the sink, and intermediate nodes are highlighted, if identified by the adversary.

Fig. 9(a) shows the inferred sensor activity for the base scheme. The adversary accurately localized the sources of two events and the paths to the sink. The sink location was also identified. Fig. 9(b) shows the inferred sensor activity for the STaR scheme. Despite the use of an intermediate node as a decoy, the global view of the adversary allowed him to pinpoint the source and sink locations, and the path to the sink. Moreover, the two events were clearly distinguishable. Fig. 9(c) shows the transmission activity inferred for phantom flooding. For this scheme, the adversary pinpointed the source location based on the earliest transmission time. Also, the adversary reconstructed the path from the source to the fake source. However, the sink location was protected by the application of the probabilistic flooding. Finally, Figs. 9(d) and 9(e) show the sensor activity inferred when the global norm. and MCDS methods are applied, respectively. Both methods successfully hid contextual information. However, in MCDS, fewer sensors were active. In Figs. 9(c), 9(d), and 9(e), we only show results for a single event to preserve visual clarity.

Source Location Privacy: In Fig. 9(f), we show the privacy distance for the source location as a function of the grid square size, normalized to the eavesdropping reception range (50m). Confidence intervals of 95% are also shown. We observe that the base, STaR, and phantom flooding schemes, the adversary can identify the source within one communication range ($\Pi \leq 1$), for sufficiently dense eavesdropper deployments ($\alpha \leq 80$). On the other hand, the global norm. and MCDS schemes maintain a relatively constant privacy distance that is 5-6 times larger than the eavesdropping reception range. The larger variance observed is due to the difference in the Euclidean distance between the random event locations and all dummy traffic sources.

Sink Location Privacy: In Fig. 9(g), we show the privacy distance for the sink location. In this figure, StaR has the same performance with the base method and MCDS has the same performance with the global norm. method. Therefore, only one curve is used to represent each pair of methods. We observe that for low α, the STaR method is unable to hide the sink location. However, for large α (small number of eavesdroppers), the sink privacy increases considerably. This is because the adversary loses trace of the path from the source to the sink, when some poorly eavesdropped area is traversed. The rest of the methods achieve high sink privacy. The privacy in phantom flooding is slightly lower than MCDS and global norm., as the path between the source and the intermediate fake source before the application of probabilistic flooding is identifiable. Finally, MCDS achieves the same privacy levels as global norm., despite the fewer active sensors in MCDS.

Temporal Privacy: In Fig. 9(h), we show the temporal privacy distance for the base, phantom flooding, and STaR schemes (results for global norm. and MCDS are not included because the event occurrence time is not identifiable in these schemes). The event occurrence time was estimated using Algorithm 4. We normalized the

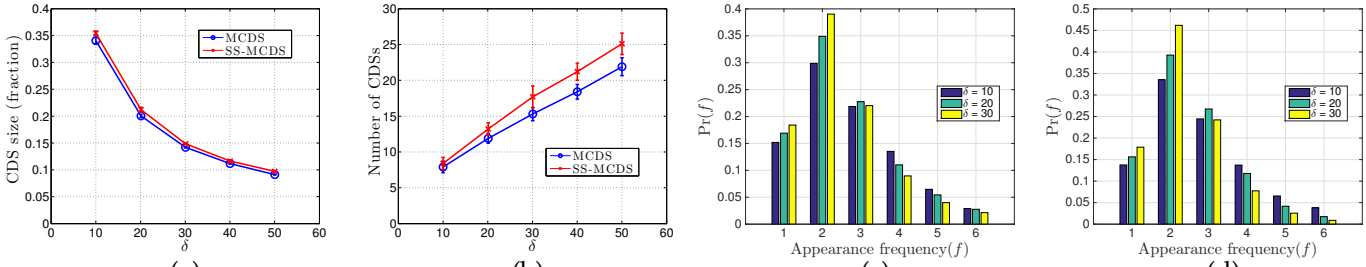

Fig. 10: (a) Average CDS size normalized over $|\mathcal{V}|$, as a function of δ, (b) average number of CDSs that span $\mathcal{V}$ as a function of δ. Empirical probability mass function of f for the (c) MCDS partition, and (d) SS-MCDS partition.

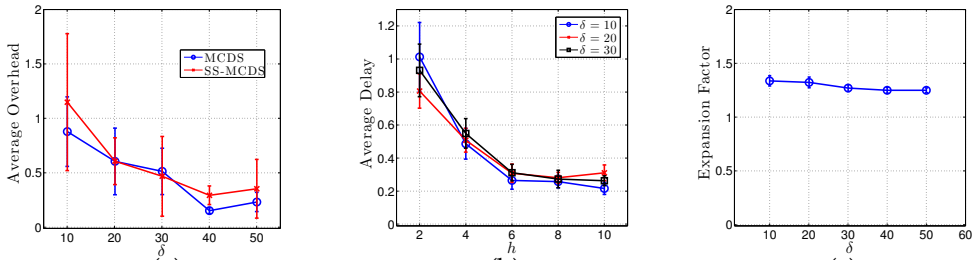

Fig. 11: (a) Average delay as a function of the hop count to the sink, (b) average delay as a function of the subpath size, (c) expansion factor between MCDS and SS-MCDS partition as a function of δ.

temporal privacy with respect to the end-to-end delay of reporting the event of interest. For base, STaR, and phantom flooding, the privacy distance takes values between 0.8 and 2, which shows that the accuracy of the adversary's estimations is in the order of the end-to-end delay. Phantom flooding exhibits larger variance compared to the other schemes, due to the probabilistic nature of the flooding operation.

6.2 Generation of a CDS Partition

In these experiments, we studied the performance of Algorithms 5 and 6 in partitioning the sensors set $\mathcal{V}$ to multiple CDSs. We deployed several WSNs at random and varied the average node degree δ by increasing the sensor density. We ran Algorithms 5 and 6 to obtain the MCDS and SS-MCDS partition. We evaluated the following metrics: (a) the average fraction of sensors that belong to a CDS (CDS size), (b) the number of CDSs needed to span $\mathcal{V}$, and (c) the probability mass function (pmf) of the appearance frequency $f(v)$ to a CDS.

Figure 10(a) shows the average size of the CDSs that span $\mathcal{V}$, as a function of δ. Confidence intervals of 95% are also shown. The CDS size directly relates to the energy savings compared to the global norm. method, which requires all sensors to be dummy traffic sources [20]. We observe that the fraction of active sensors decreases with δ, which indicates higher energy savings. Furthermore, the size difference between MCDSs and SS-MCDSs is close to 2%, for all values of δ. This indicates that the SS-MDSS partition optimizes the paths to the sink without a significant penalty on the CDS size. Figure 10(b) shows the average number of CDSs needed to span $\mathcal{V}$, as a function of δ. This value is related the delay until a CDS containing a sensor with a real packet for transmission becomes active. We observe a linear increase in the number of CDSs with δ. Also note that the total number of SS-MCDSs required to span $\mathcal{V}$ is slightly higher than the number of MCDSs. This difference is attributed to the additional requirement of including shortest paths to the sink for the SS-MCDS case.

In Figs. 10(c) and 10(d), we show the empirical pmf for the appearance frequency $f(v)$ when constructing MCDSs and SS-MCDSs, respectively. The $f(v)$ represents the "quality" of the partition (ideally, $f(v) = 1, \forall v \in \mathcal{V}$). For both partitions types, more that 50% of sensors are part of one or two CDSs, while for 95% of the sensors, $f(v) < 5$. This indicates that Algorithm 6 favors the creation of disjoint CDSs to a large extend, thus reducing the per-sensor dummy traffic overhead.

6.3 Communication and Delay Overheads

In the last set of experiments, we studied the communication overhead and end-to-end delay for delivering real packets to the sink. We compared MCDS with the global norm. method, because only those two achieve the same privacy. In our experiments, each CDS remained active for one epoch. To reduce the event buffering delay due to CDS rotation, an event was reported by all sensors that sensed it. To provide a fair comparison, we selected a dummy packet rate in the global norm. method that achieves the same end-to-end delay as in MCDS.

Figure 11 (a) shows the average communication overhead as a function of δ. The overhead is normalized to the traffic volume introduced by the global norm. method. We observe that the SS-MCDS introduces an overhead of 120% when $\delta = 10$, while for the MCDS, the overhead drops to 90%. This difference is primarily due to the delay introduced by the CDS rotation. As a

partition to SS-MCDSs requires a larger number of SS-MCDSs to span $\mathcal{V}$, a sensor that has detected an event has to wait longer until its CDS becomes active. The communication overhead is drastically reduced with the increase of δ. This is due to the fact that under denser sensor deployments, more sensors can detect an event. These sensors likely belong to different CDSs because they are within the same neighborhood. Therefore, the buffering delay is reduced.

In Fig. 11(b), we show the average end-to-end delay achieved by DFAS as a function of the subpath size h. The delay is normalized to the delay incurred when transmissions are uncoordinated. To provide a fair comparison, each scheme was restricted to transmit the same number of dummy packets. We considered events reported by sensors located 7 hops away from the sink (which is the highest hop-count in the network). As expected, the delay reduces with the increase of h, as a larger number of hops is traversed per epoch. Finally, in Fig. 11(c) we present the average path expansion factor for the MCDS method relative to the SS-MCDS method, as a function of δ. The expansion factor is defined as the ratio between the path length to the sink, when MCDSs are constructed, relative to the shortest path. We observe that MCDS produces slightly longer routing paths to the sink (in the order of 30% when $\delta = 10$.) However, this difference reduces with δ due to the availability of more routes with short paths to the sink.

7 Conclusions

We addressed the problem of contextual information privacy in WSNs under a global eavesdropper. We presented a general traffic analysis method for collectively processing the packet interception times and eavesdropper locations at a fusion center. The method is agnostic to the protection mechanism and can be used as a baseline for evaluating different schemes. To mitigate global eavesdropping, we proposed traffic normalization methods that regulate the sensor traffic patterns of a subset of sensors that form MCDSs. We developed two algorithms for partitioning the WSN to MCDSs and SS-MCDSs and evaluated their performance via simulations. Compared to prior methods capable of protecting against a global eavesdropper, we showed that limiting the dummy traffic transmissions to MCDS nodes, reduces the communication overhead due to traffic normalization. We further proposed a loose transmission coordination scheme that reduces the end-to-end delay for reporting events.

Acknowledgments

This research was supported in part by the NSF under grant CNS-1409172 and ARO grant W911NF-13-1-0302. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] M. Akhlaq and T. R. Sheltami. RTSP: An accurate and energy-efficient protocol for clock synchronization in wsns. *IEEE Transactions on Instrumentation and Measurement*, 62(3):578–589, 2013.

[2] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Toward a statistical framework for source anonymity in sensor networks. *IEEE Transactions on Mobile Computing*, 12(2):248–260, 2013.

[3] F. Armknecht, J. Girao, A. Matos, and R. Aguiar. Who said that? privacy at the link layer. In *Proc. of the INFOCOM Conference*, pages 2521–2525, 2007.

[4] K. Bicakci, H. Gultekin, B. Tavli, and I. Bagci. Maximizing lifetime of event-unobservable wireless sensor networks. *Computer Standards & Interfaces*, 33(4):401–410, 2011.

[5] G. Chinnu and N. Dhinakaran. Protecting location privacy in wireless sensor networks against a local eavesdropper–a survey. *International Journal of Computer Applications*, 56(5):25–47, 2012.

[6] M. Conti, J. Willemsen, and B. Crispo. Providing source location privacy in wireless sensor networks: A survey. *Communications Surveys Tutorials*, 15(3):1238–1280, 2013.

[7] J. Deng, R. Han, and S. Mishra. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing*, 2(2):159–186, 2006.

[8] M. Fruth. Probabilistic model checking of contention resolution in the IEEE 802.15.4 low-rate wireless personal area network protocol. In *Proc. of the Symp. on Leveraging Applications of Formal Methods, Verification and Validation*, pages 290–297, 2006.

[9] M. Garey and D. Johnson. *Computers and Intractability*, volume 174. Freeman, 1979.

[10] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proc. of the ACM Conference on Mobile Systems, Spplications, and Services*, pages 40–53, 2008.

[11] J. Gross and J. Yellen. *Handbook of Graph Theory*. CRC, 2004.

[12] A. Jhumka, M. Leeke, and S. Shrestha. On the use of fake sources for source location privacy: Trade-offs between energy and privacy. *The Computer Journal*, 54(6):860–874, 2011.

[13] L. Jia, R. Rajaraman, and T. Suel. An efficient distributed algorithm for constructing small dominating sets. *Distributed Computing*, 15(4):193–205, 2002.

[14] Y. Li and J. Ren. Source-location privacy through dynamic routing in wireless sensor networks. In *Proc. of the INFOCOM Conference*, pages 1–9, 2010.

[15] L. Lightfoot, Y. Li, and J. Ren. Preserving source-location privacy in wireless sensor network using STaR routing. In *Proc. of the IEEE GLOBECOM conference*, pages 1–5, 2010.

[16] X. Luo, X. Ji, and M. Park. Location privacy against traffic analysis attacks in wireless sensor networks. In *Proc. of the IEEE Conference on Information Science and Applications*, pages 1–6, 2010.

[17] M. Mahmoud and X. Shen. A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1805–1818, 2012.

[18] M. Mahmoud and X. Shen. A novel traffic-analysis back tracing attack for locating source nodes in wireless sensor networks. In *Proc. of the IEEE ICC Conference*, pages 939–943, 2012.

[19] M. Mahmoud and X. Shen. Secure and efficient source location privacy-preserving scheme for wireless sensor networks. In *Proc. of the IEEE ICC Conference*, pages 1123–1127, 2012.

[20] K. Mehta, D. Liu, and M. Wright. Protecting location privacy in sensor networks against a global eavesdropper. *IEEE Transactions on Mobile Computing*, 11(2):320–336, 2012.

[21] D. N. Ngo. Deployment of 802.15.4 sensor networks for C4ISR operations. Technical report, DTIC Document, 2006.

[22] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proc. of the ACM SASN Workshop*, pages 88–93, 2004.

[23] M. Shao, Y. Yang, S. Zhu, and G. Cao. Towards statistically strong source anonymity for sensor networks. In *Proc. of the INFOCOM Conference*, pages 51–55, 2008.

[24] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, 2000.

[25] J. A. Stankovic, A. D. Wood, and T. He. Realistic applications for wireless sensor networks. In *Theoretical Aspects of Distributed Computing in Sensor Networks*, pages 835–863. 2011.

[26] Y.-C. Wu, Q. Chaudhari, and E. Serpedin. Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, 28(1):124–138, 2011.
[27] Y. Xi, L. Schwiebert, and W. Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *Proc. of the Parallel and Distributed Processing Symposium*, pages 1–8, 2006.
[28] W. Yang and W. Zhu. Protecting source location privacy in wireless sensor networks with data aggregation. In *Proc. of the UIC Conference*, pages 252–266, 2010.
[29] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic in sensor networks. In *Proc. of the ACM WiSec Conference*, pages 77–88, 2008.