Packet Traffic: A Good Data Source for Wireless Sensor Network Modeling and Anomaly Detection

Qinghua Wang, Aalto University

Abstract

The wireless sensor network (WSN) has emerged as a promising technology. In WSNs, sensor nodes are distributedly deployed to collect interesting information from the environment. Because of the mission of WSNs, most node-wide as well as network-wide activities are manifested in packet traffic. As a result, packet traffic becomes a good data source for modeling sensor node as well as sensor network behaviors. In this article, the methodology of modeling node and network behavior profiles using packet traffic is exemplified. In addition, node as well as network anomalies are shown to be detectable by monitoring the evolution of node/network behavior profiles.

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, small, yet reasonably efficient wireless sensor nodes. These tiny sensor nodes, consisting of sensing, data processing, communicating, and power source components, make a new technological vision possible: wireless sensor networks (WSNs).

WSNs combine short-range wireless communication, minimal computation facilities, and some kinds of sensing functions into a new form of network that can be deeply embedded in our physical environment. They involve deploying a large number of tiny sensor nodes in either hostile or non-hostile environments. The nodes then sense environmental changes and report them to other nodes (usually sink nodes connected to the end user) over a flexible network architecture. Figure 1 shows the architecture of a WSN. Because there is no, or only limited, infrastructure, WSNs are usually self-organized.

Based on the vision of WSNs, new types of applications become possible. Possible applications include environmental monitoring such as wildfire and climate change monitoring; structural health monitoring; patient health monitoring; and so on. Due to the mission of WSNs and the low cost of sensor nodes, individual sensor nodes must forward their sensed data to the base station for final processing. This means most node-wide as well as network-wide activities are manifested in packet traffic. As a result, packet traffic is a good data source for modeling the behaviors of both individual sensor nodes and the whole network. In this article, the methodology of modeling sensor node as well as sensor network behaviors using packet traffic is exemplified. Being different from the Internet, telecommunication networks, and mobile ad hoc networks, the behaviors of sensor nodes as well as sensor networks are not controlled by humans during their runtimes. Although humans can interact with a WSN through predefined interfaces, the impact of the subjective uncertainty in node and network behaviors is minimized in WSNs. Therefore, node and network behaviors are relatively more stable than those in traditional networks. As for sensor network/node behavior modeling, we get stable network/node profiles. Because WSNs face security and reliability threats [1], and many of these threats could cause changes to otherwise stable network/node behavior profiles, any violation of security and reliability could be detected as an anomaly by monitoring the evolution of network/node behavior profiles. In this article, detecting anomalies according to the evolution of (packet-traffic-based) network/node behavior profiles is shown to enhance the security and reliability of WSNs.

Basics of Packet Traffic Modeling

Packet traffic modeling and classification are found to be important in many areas such as quality of service (QoS) provisioning, traffic analysis, traffic simulation, traffic prediction, and network anomaly detection. The task of traffic modeling is to find statistically invariant properties of packet traffic, which subsequently can be used to identify the types of packet traffic.

The types of different traffic can be differentiated by their associated applications, locations, times, and so on. To model a certain type of packet traffic, traffic features like packet train size and packet train length, interpacket times, and payload size are used to characterize packet traffic. According to [2], traffic features can be mainly grouped into *basic features, time-based features*, and *connection-based features* according to their underlying implementation. Table 1 gives a summary of the frequently used traffic features in packet traffic modeling.

One or a combination of traffic features can be used to statistically model a certain type of packet traffic. In WSNs, packet traffic associated with a sensor node can be used to model the behavior of that specific node; routing traffic can be used to model the behavior of routing protocols; and so

 0890-8044/11/$25.00

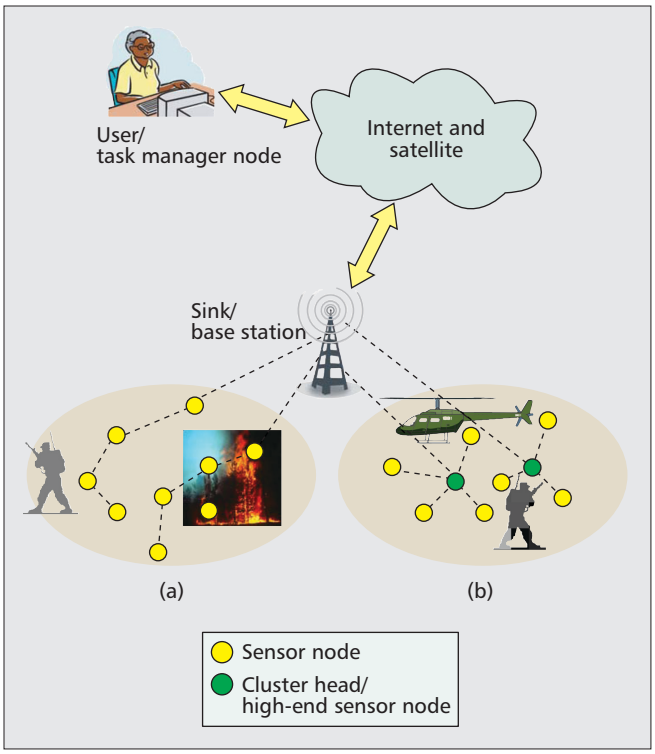

Figure 1. *The architecture of WSNs.*

on. In the following, examples of modeling the behaviors of sensor nodes and sensor networks using packet traffic are presented.

Modeling Node Profile

In WSNs, sensor nodes cooperate to finish a communication task. Each sensor node functions as a sensing data source as well as a relay for the others. But no sensor node has a full view of network-wide communication. This is partially due to the fact that each sensor node has its unique sets of relay nodes and child nodes. This is also due to the fact that the sensing operation is fully controlled by individual sensor nodes and may be different from sensor node to sensor node. With this insight, we can uniquely model a sensor node using the unique packet traffic set observed at that specific sensor node.

Modeling Node Profile Based on Packet Sequence

Sequence relations exist among some types of packets. For example, a Routing Reply (RREP) message always comes after a Routing Request (RREQ) message, which is specified by a routing protocol. In [3], the authors propose to use a finite state machine (FSM) to specify the correct ad hoc on demand vector (AODV) routing behavior. The authors in [4] also use an FSM to model the correct routing behavior for another routing protocol.

In addition, the sequence relations among some special types of packets can be modeled according to protocol specifications, and the sequence relations among general kinds of packets can also be learned through training. In our former work [5], a methodology of automatically learning sequence relations for packets arriving at a sensor node has been presented. The general idea of this methodology is described in the following. Due to the unique traffic set observed at each sensor node, the learned set of packet sequence relations is unique for each sensor node. Therefore, a sensor node is uniquely modeled by its corresponding packet sequence relations.

Basic features	Protocol type, source/destination address, flags, payload size
Time-based features	Interpacket times, frequency, packet sequence
Connection-based features	Packet train size, packet train length

Table 1. *Traffic features used for packet traffic modeling.*

Packet Classification — To learn the sequence relations among packets, those packets must first be classified properly. Otherwise, either the class set has an unmanageable size, or the learned sequence relations have no practical use. We propose to classify the packets in such a way that the whole set of packet categories can be mapped to a set of single-byte ASCII characters, and the sequence relations learned based on the classified packets can reflect the unique behavior of the node of interest.

As a demonstration, we are going to classify packets according to the combinations of the two traffic features *Packet Type* and {*Src*, *Dest*} (i.e. the abbreviation for the pair of source and destination addresses). In order to control the number of packet categories and make the packet classification scheme scalable, we further map the real node address space to an abstracted address space. The abstracted address space has only five entries: {me; neighbor; local; unlocal; and sink/cluster header}, which are classified from the point of view of the node of interest. In concrete terms, "me" is the node of interest, "neighbor" represents all those nodes within one hop distance of the node of interest, and "local" represents all those nodes that are already known by the node of interest through learning of the source and destination nodes of all its previously observed packets. During the packet sequence learning, no node is classified as "unlocal." Once a stable set of all learned packet sequence relationships is acquired, the observation of a packet with its source or destination node classified as "unlocal" is usually a sign of anomaly.

Packet Translation — For simplicity, the classified packets can be further mapped to a set of single-byte ASCII characters. Figure 2 shows the process of packet classification and the process of mapping the classified packets to a character set. Finally, the sequence of packets arriving at a node of interest can be viewed as a large (or asymptotically infinite) string of characters.

Pattern Extraction — To learn the sequence relationships among the arriving packets, we must extract patterns from the large (or asymptotically infinite) string of characters. The pattern extraction algorithm first proposed by Forrest *et al.* in [6, 7] for intrusion detection in a Unix system is used in this case. During pattern extraction, the arriving sequence of the abstracted packet events (i.e., the character string) is scanned for all given length, k, unique sub-sequences. Simultaneously, a set of all such unique sub-sequences that have been found (i.e., patterns) is built. Once a stable set of patterns has been constructed, the process of pattern extraction is completed, and a behavior profile for the node under consideration is acquired.

The construction of the pattern set is best illustrated with an example. For $k = 4$ and the sample sequence AABBDCC, we obtain the following pattern set: AABB, ABBD, BBDC, BDCC.

The method of pattern extraction can be more complicated. If we consider the fact that packets are usually sent based on connections, we can ignore the sequence relations among

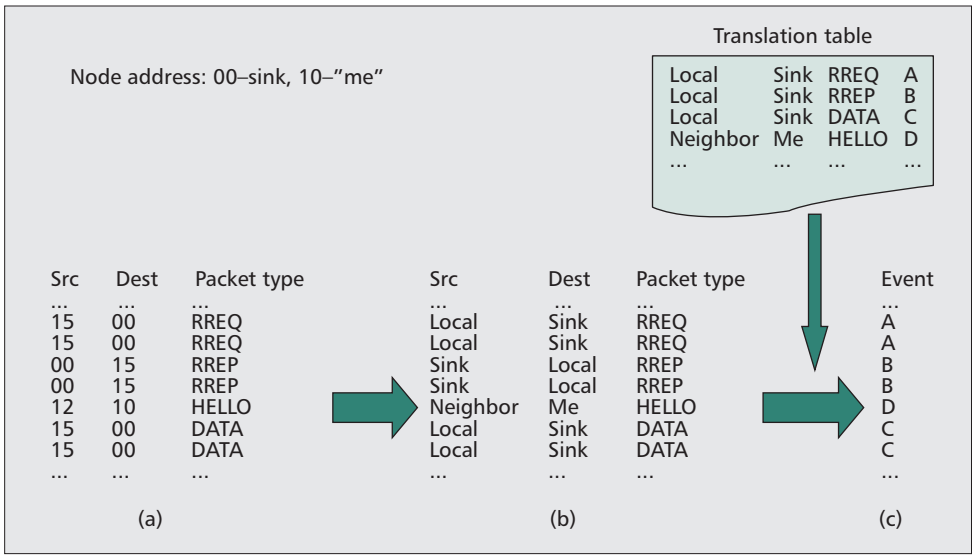

Figure 2. *Translation of packet arriving events to characters.*

packets belonging to different connections. The packet arriving interval roughly indicates whether two successive packets belong to the same connection.

By considering multiple k, we can acquire patterns with variable lengths. However, doing this may significantly increase the size of the pattern set. The size of the pattern set can be somewhat reduced by only considering those patterns that have been observed at least twice or more times. Furthermore, only maximal length patterns should be considered when a pattern contains a sub-sequence pattern that has the same number of occurrences.

Modeling Node Profile Based on Source Traffic

Event-driven data collection and dissemination have been adopted by many proposed WSN scenarios. In an event-driven WSN, bursty source traffic may arise from any corner of the sensing area if an event or interesting phenomenon is detected by the local sensor nodes. A Poisson process has been used to model the traffic arrival process in an event-driven WSN [8]. However, there are no solid grounds to support the use of a Poisson process in this situation. Instead of using Poisson processes, we propose the use of an ON/OFF model to capture the burst phenomenon exhibited in the source traffic of an event-driven WSN [9]. Because each sensor node's sensing operation is independent and relies on the physical property of its unique environment, models learned based on source traffic can be used to specify the behavior profile of a sensor node.

ON/OFF Model — An ON/OFF model [10] can be used to capture the burst phenomenon exhibited in the source traffic of a sensor node, where the source traffic is generated due to sensing operation. In the ON/OFF model, each ON interval corresponds to a time span when an interesting physical phenomenon appears within the sensing range, and each OFF interval represents a time span when there is no interesting phenomenon within the sensing range. Figure 3 shows the state transition diagram of the ON/OFF model, and Fig. 4 shows an example of how we can build on the ON/OFF model. The ON/OFF model includes two states: ON and OFF. The default state is OFF when the model starts training. Each time a source packet is observed, the state is turned ON if it is currently OFF, and an ON timer is started. The state is kept ON until the ON timer expires. If another source packet arrives before the ON timer expires, the ON timer is restarted. If no source packet arrives during the ON timer window, the state is turned OFF. The total amount of time the state remains ON is designated as the ON period. Similarly, the time duration when the state remains OFF is designated as the OFF period. Each ON period indicates an event when there is an interesting physical phenomenon in the neighborhood of the considered sensor node, while the duration of an ON period states the duration of that interesting phenomenon.

Properties of the ON/OFF Model — There are many properties of the ON/OFF model. Often used ones include the number of ON periods observed in a unit time span, the distribution of the duration of ON periods, and the distribution of the duration of OFF periods. All these properties can be learned when the ON/OFF model is applied to model the source traffic associated with a specific sensor node. In the end, the ON/OFF model together with its learned properties specify a sensor node's source traffic profile.

Case Study: Source Traffic Modeling in WSNs for Target Tracking — A typical WSN for target tracking consists of spatially distributed sensor nodes monitoring a mobile target collaboratively. When the target enters the surveillance area, any sensor node with appropriate sensing ability will discover this target as long as the target is within its sensing range. When the target remains in the sensing range, the alerted sensor node will report the target discovery event regularly to a remote base station by means of multihop communication. When the target is out of the sensing range, there will be no event reporting. Obviously, this kind of event reporting will generate bursty source traffic at individual sensor nodes.

In our former work [9], a simulation was done to explore the dynamics of source traffic in a target tracking scenario. The simulation found that different sensor nodes experience different event reporting situations, which are determined by the locations of sensor nodes as well as the mobility model used by the mobile target. When the ON/OFF model is applied to model the source traffic associated with each sensor node, it was found that the source traffic at each sensor node exhibits its uniqueness in terms of the frequency of the observation of ON periods, the distribution of the duration of ON periods, and the distribution of the duration of OFF periods. Therefore, these properties of the ON/OFF model uniquely identify a sensor node and provide a profile for the node of interest.

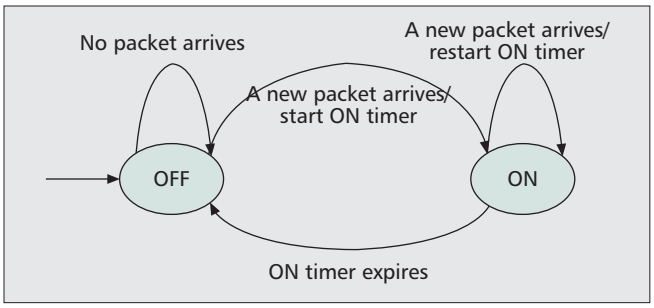

Figure 3. *ON/OFF state transition diagram.*

Modeling Network Profile

Besides individual sensor nodes, the whole sensor network can also be modeled by identifying the statistical invariants exhibited in network-wide packet traffic.

Monitoring Network-Wide Packet Traffic

Because communication is wireless inside a WSN, monitoring packet traffic is not a difficult task. A problem may arise when there is a need to monitor all network-wide packet traffic. Due to the lack of infrastructures, monitoring all network-wide packet traffic will require a deployment of monitoring nodes all over the network. This is not always realistic. Fortunately, there is a slightly downgraded but much easier plan B. In a typical WSN, the task is to collect sensed information from all over the network and forward it to a powerful base station for final processing. As a result, most data packets as well as routing packets are destined to the base station. Therefore, it is possible to monitor most network-wide packet traffic by simply attaching a monitor to the base station.

Classification of Network-Wide Packet Traffic

When there is access to network-wide packet traffic, appropriate classification of the observed packet traffic provides a behavior profile for the network as a whole.

We mentioned earlier that selected features can be used to represent one or one group of packets. If the features describing the same (group of) packet(s) is viewed as a data feature vector, we have a set of data feature vectors describing all network-wide packet traffic. To make a classification of network-wide packet traffic, a clustering algorithm can be applied to the set of data feature vectors. By organizing data feature vectors into groups whose members are similar in some way, a clustering algorithm finds the structure of the dataset or of the network. If it is not clear, the following case study gives more intuition.

Case Study: Grouping Sensor Nodes

In this example, we have three types of sensor nodes in a WSN. Each type of sensor node has a different sensing pattern. Carbon dioxide sensor nodes sense the gas periodically and report their data to the base station at a low data rate. Temperature sensor nodes also sense temperature periodically but report their data to the base station at a high data rate. There is a third type, motion sensor nodes, which constantly measure optical and acoustical changes in their fields of view but only report their data to the base station when there is a change (i.e., when motion is detected).

The problem here is that we know there are three types of sensor nodes, but we do not know which sensor node belongs to which type. What we need to do is to model this WSN by correctly grouping all these sensor nodes, and by classifying them according to their types. We can do this by attaching a monitor to the base station. Because the base station basically receives all data packets from the deployed sensor nodes, it is possible to classify sensor nodes according to the features of those data packets received from them. In this example, we consider the frequency of data packets received from each sensor node. CO_2 and temperature sensor nodes have periodical data reporting rates. Therefore, the frequency of data packets arriving at the base station from these two types of sensor nodes can be modeled by an addition of a constant and Gaussian noise. The constant is the true data reporting rate, while the Gaussian noise represents the uncertainty (due to delay, packet loss, packet retransmission, etc.) when a packet is sent from a sensor node to the base station.

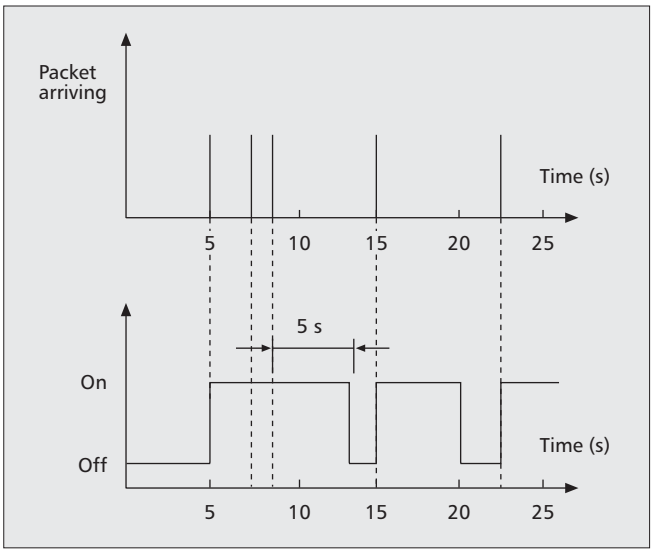

Figure 4. *ON/OFF state transitions with an ON timer of 5 s.*

For motion sensor nodes, they only report data when they have detected a motion in their fields of view. Therefore, they do not have a constant data reporting rate. In this example, we use the Poisson model to describe the dynamic data reporting rate of motion sensor nodes. In addition, the observed packet arriving frequency from a motion sensor node also suffers from Gaussian noise for the same reason described above.

With the packet arriving models, we can simulate the packet arriving at the base station. We assume there are 10 CO_2 sensor nodes, 10 temperature sensor nodes, and 10 motion sensor nodes in total. CO_2 sensor nodes have a constant data reporting rate of 12 times per hour. Temperature sensor nodes have a constant data reporting rate of 60 times per hour. Motion sensor nodes report their data based on the detection of motions, and the expected frequency of data reporting per hour is 15 (i.e., the mean in the Poisson model). We use a simple assumption that the Gaussian noise representing the difference between the packet arriving rate observed at the base station and the data reporting rate at sensor nodes is the same for all types of sensor nodes, and the noise follows $N(0, 4)$.

For each sensor node s_i, we record 100 samples $x_{i,1}, x_{i,2}, \ldots, x_{i,100}$ of their observed packet arriving rates at the base station. Each sample corresponds to the observed number of packets arriving within a one hour interval. From the samples, the data features of mean μ_i and standard variance v_i are extracted as a representation of sensor node s_i. Therefore, we have a two-dimensional data point $[\mu_i, v_i]$ associated with each sensor node s_i. The collection of data points associated with all 30 sensor nodes is shown in Fig. 5.

K-means has been used as the clustering algorithm to classify the dataset. The centroids found for the three clus-

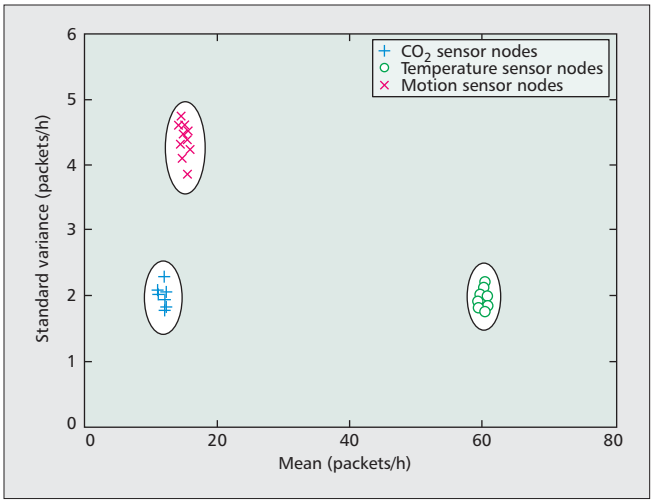

Figure 5. *Grouping sensor nodes according to their packet arriving features at the base station.*

ters shown in Fig. 5 are (12.01, 2.00), (60.05, 1.94), and (15.10, 4.40). Assigning each data point to its nearest centroid, we get three clusters. It is shown that all data points falling into the same cluster are associated with the same type of sensor node. Thus, it can be concluded that sensor nodes can be grouped correctly according to their observed packet traffic features. When there is no prior knowledge about the type differences of sensor nodes, a network profile learned like this provides some basic information about the structure of a WSN.

Detecting Anomalies Based on Node/Network Profiles

It has been shown that behavior profiles of individual sensor nodes as well as the whole sensor network can be modeled based on their associated packet traffic features. In this section it is shown that the built profiles can be used as the basis for anomaly detection.

Detecting Abnormal Packet Sequences

Above we have shown that the sequence relations among packets arriving at a sensor node can be learned. Actually, the learned set of sequence relations represents the unique behavior profile of the node of interest. Because a sensor node in a WSN has its role assigned, and only performs necessary and specified operations, the number of patterns exhibited in the observed packet sequence is limited. Once a stable pattern set has been built, any unacquainted new pattern or packet subsequence observed at the node of interest is highly suspicious and should signal an anomaly.

Pattern Matching and Alarm — When a node profile consisting of patterns is used for anomaly detection, pattern matching is used to find out whether a new packet sub-sequence is a known pattern or not. Pattern matching is similar to pattern extraction. A buffer window of length k is maintained across the sequence of arriving packets during runtime monitoring. Each time a new interesting packet arrives, the buffer window is moved forward by one position and checked for a *match* (i.e., whether there is a pattern that matches the sub-sequence in the buffer window). If no matching pattern exists, this is called a *mismatch*.

Let a and b be two sequences of length k. The expression a_i designates the character at position i. The difference $d(a, b)$ between a and b is defined as

$$d(a,b) = \sum_{i=1}^{k} f_i(a,b),$$
$$\text{where } f_i(a,b) = \begin{cases} 0 & \text{if } a_i = b_i \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

During pattern matching, we determine for each subsequence u of the arriving packet sequence the minimum distance $d_{\min}(u)$ between u and the entries in the pattern set,

$$d_{\min}(u) = \min \{d(u, p) \quad \forall \text{ patterns } p\}. \quad (2)$$

To detect an anomalous event, at least one of the observed sub-sequences affected by this event must be classified as anomalous. In terms of the above measure, there is at least one sub-sequence u for which $d_{\min}(u) > 0$.

In the ideal case, any $d_{\min}(u)$ value greater than 0 can be considered as a sign of an anomalous event. However, a complete match cannot always be achieved, especially for a network with mobility and a dynamic routing strategy. Therefore, a threshold can be defined such that only sub-sequences whose $d_{\min}(u)$ value is above this threshold are considered suspicious. Once a packet subsequence is detected as suspicious, an alarm is launched.

When variable-length patterns are used, multiple buffer windows with different lengths should be adopted. The pattern matching method is the same for all pattern lengths. If a packet sub-sequence is considered to be a pattern only after it has occurred for two or more times, the pattern matching should adopt the same strategy. That is, a mismatch of a packet sub-sequence would not be considered anomalous until the packet sub-sequence under consideration has been observed more than two or more times during the pattern matching stage.

Case Study: Detection of Sinkhole Attack — A node profile consisting of packet sequence patterns can be used to detect many kinds of attacks in a WSN. Examples of attacks that can be detected using this methodology include *ID spoofing*, *sinkhole,* and *wormhole* attacks [5]. Due to space limitations, only the detection of a sinkhole attack is depicted in the following.

In a sinkhole attack, a malicious node manages to attract routes from many other nodes to go through it, thus acting as a "sinkhole." This attack typically works by making the malicious node appear especially attractive to surrounding nodes, for example, by claiming a short or fast route to the destination. If the attacker succeeds, data traffic attacks can be launched, and these can prevent the discovery of other legitimate routes.

This attack would violate the normal profile of the malicious node. Since the malicious node attempts to attract routes that would never pass through it in a normal situation, many unlocal packets (with previously unknown addresses as the source and/or destination) would be observed at the sinkhole during the attack. The introduction of these unlocal packets would disrupt the normal packet arriving sequence, and many pattern mismatches would be expected. Accordingly, alarms would be launched.

Detecting Abnormal ON/OFF Periods

In an event-driven WSN, bursty source traffic due to the discovery of interesting events can originate from any corner of the sensing field. Earlier, an ON/OFF model was used to capture the burst phenomenon of event-driven source traffic. An ON period corresponds to a period of continuous observation

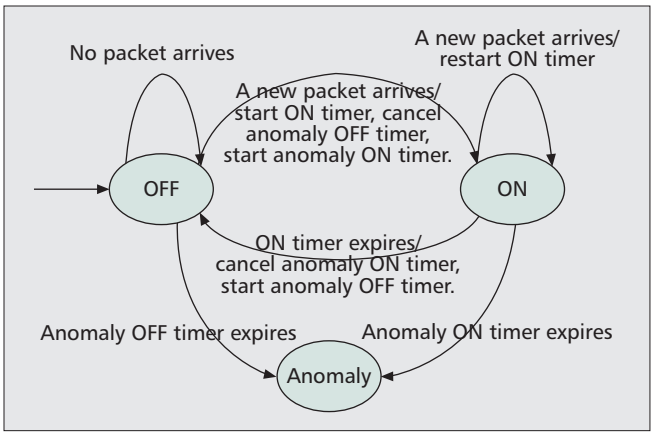

Figure 6. *Grouping sensor nodes according to their packet arrival features at the base station.*

of interesting events, while an OFF period is a silent period between two adjacent ON periods when there is no observation of interesting events. The duration of an ON/OFF period is not fixed. However, the probability of observing extremely long ON/OFF periods is low. When there is an unusually long ON period, it could be due to a compromised sensor node or the result of an energy exhaustion attack. When there is an unusually long OFF period, it could be the result of a link or node failure. Therefore, the appearance of an unusually long ON/OFF period is highly suspicious and should trigger an anomaly alarm to receive special attention. In the following, the methodology of detecting unusually long ON/OFF periods is shown.

The Methodology of Abnormal ON/OFF Period Detection — The goal is to detect those unusually long ON/OFF periods so that they can be identified for further analysis. Given that the distributions of the duration of ON/OFF periods can be statistically acquired after training for a certain length of time, a probabilistic upper length limit (e.g., $x|F(x) = 0.99$, where x is the length of an ON/OFF period duration) for the duration of ON/OFF periods can easily be acquired for any node of interest. Our strategy is that an abnormal ON/OFF period is detected whenever there is an unusually long ON/OFF period of a duration longer than the specified upper length limit. We thus describe the new ON/OFF state transition diagram (an old diagram is shown in Fig. 3) for anomaly detection in Fig. 6, where the length of the anomaly ON/OFF timer is set to be a probabilistic upper length limit for any ON/OFF period. For a target tracking sensor network considered in [9], it has been found that the distributions of the duration of ON/OFF periods have short tails. A short tail of a distribution means that the support range of the variable is concentrated in a small region, and there is an extremely low probability that the variable will take a large value. With the short tail property, an anomaly regarding an unusually long ON/OFF period can be quickly detected with high confidence.

Detecting Changes of Network Profiles

The detection of network profile change is a method of anomaly detection as behavior change of individual nodes or links or other network objects is reflected in an updated network profile. By considering the similarity of the updated and old network profiles, network anomalies are detected. When a network profile is given by an appropriate classification of network objects, the change of network profile can be detected by checking out the similarity between an old classification and an updated classification.

	U_1	U_2	U_3
V_1	8	0	0
V_2	2	10	0
V_3	0	0	10

Table 2. *Contingency table.*

Measuring the Similarity between Two Classifications — Let S be a set of N sensor nodes (or other network objects that can be identified). Given two classifications of S, $U = U_1, U_2, \ldots, U_u$ with u clusters and $V = V_1, V_2, \ldots, Vv$ with v clusters, the information on cluster overlap between U and V can be summarized in the form of a $u \times v$ contingency table where each cell denotes the number of objects that are common to a cluster in U and a cluster in V (Table 2).

One method to evaluate the similarity of two classifications is to measure the mutual information. In this case, the entropy associated with classification U is $H(U) = -\sum_{i=1}^{u} P_U(i) \log P_U(i)$, where

$$P_U(i) = \frac{|U_i|}{N},$$

and the entropy associated with classification V is $H(V) = -\sum_{j=1}^{v} PV(j)\log PV(j)$, where

$$P_V(j) = \frac{|V_j|}{N}.$$

The joint entropy of U and V is $H(U, V) = -\sum_{i=1}^{u} \sum_{j=1}^{v} P_{UV}(i, j) \log P_{UV}(i, j)$, where $P_{UV}(i, j)$ denotes the probability that a sensor node (or another object) belongs to cluster U_i in U and cluster V_j in V and

$$P_{UV}(i,j) = \frac{|U_i \cap V_j|}{N}.$$

According to the definition of mutual information, we have

$$I(U;V) = H(U) + H(V) - H(U,V) = \sum_{i=1}^{u}\sum_{j=1}^{v} P_{UV}(i,j)\log\frac{P_{UV}(i,j)}{P_U(i)P_V(j)}. \quad (3)$$

The drawback with the mutual information as a measure of similarity is that it tends to be larger when the two classifications have a larger number of clusters (with a fixed N). To conquer this drawback, we can use a normalized variant. The Jaccard distance has an interpretation here:

$$D(U,V) = 1 - \frac{H(U \cap V)}{H(U \cup V)} = 1 - \frac{I(U;V)}{H(U,V)}. \quad (4)$$

The Jaccard distance is a normalized metric that has a value range [0,1]. The higher the Jaccard distance, the less similar the two classifications.

Case Study: Detecting Misbehaving Sensor Nodes — In the example given earlier, there are three types of sensor nodes with 10 sensor nodes for each type. If all sensor nodes are classified correctly, there are three clusters with 10 sensor nodes in each cluster. The entropy of this classification is 1.585 bits. If the network profile does not change, a new classification will end up the

same. In this case, the entropy of the mutual information between the new and old classifications is the same as the entropy of either the new or old classification. Because the joint entropy will also be equal to the entropy of either the new or old classification, the Jaccard distance will be 0 according to Eq. 4.

However, the Jaccard distance will be greater than 0 if the network profile changes. For example, assume that there are two CO_2 sensor nodes which are misbehaving and they are disguising themselves as temperature sensor nodes. The result of this misbehavior is that an updated network profile (i.e. a new classification) end up with two CO2 sensor nodes being misclassified as temperature sensor nodes. In this case, there are 8 nodes in the CO2 sensor node cluster, 12 nodes in the temperature sensor node cluster and 10 nodes in the motion sensor node cluster. The contingency table is shown in Table 2 where U is used to represent the old classification and V is used to represent the new classification.

From the contingency table, we can get the joint probability distribution by dividing each cell with the total number of sensor nodes. Finally, it can be calculated that the entropy of the mutual information is 1.325 bits, while the joint entropy is 1.826 bits. From Eq. 4, we know the Jaccard distance between the new and old classifications is 0.274. Because the Jaccard distance is greater than 0 in this case, a network profile change is detected. By comparing the old and the new classifications, we can even identify which sensor nodes are misbehaving.

Conclusions

The packet traffic patterns of WSNs are much simpler and less dynamic than those of more traditional networks (e.g., the Internet). This makes it possible to build precise behavior profiles for individual sensor nodes as well as for the whole WSN based on observed packet traffic. In this article the methodology of building node/network profiles in a WSN based on packet traffic is presented with examples. It is shown that many packet traffic features can be extracted for profile building purposes. Packet arriving sequence, packet arriving interval, and packet arriving frequency are the ones used in this article. Node/network profiles based on different packet traffic features and packet traffic types reflect different aspects of node/network behaviors.

Because WSNs are basically simple networks in terms of node/network behaviors, node/network profiles evolve slowly over time. Once there is a sign of a change in node/network profiles, there is a high risk that something unexpected is happening. Based on this rationale, it is proposed that anomalies due to malicious threats and reliability defects can be detected by monitoring changes appearing in node/network profiles. The feasibility of this proposition has been demonstrated in this article through analysis and examples.

References

[1] Q. Wang and T. Zhang, "A Survey on Security in Wireless Sensor Networks," Ch. 14, *Security in RFID and Sensor Networks*, Y. Zhang and P. Kitsos, Eds. CRC Press, Taylor & Francis Group, 2009, pp. 293–320.

[2] I.-V. Onut and A. A. Ghorbani, "Features vs. Attacks: A Comprehensive Feature Selection Model for Network Based Intrusion Detection Systems," Ch. 2, *LNCS*, ser. 4779, J. G. *et al.*, Eds., Springer-Verlag, 2007, pp. 19–36.

[3] C. Tseng *et al.*, "A Specification-Based Intrusion Detection System for AODV," *Proc. 1st ACM Wksp. Security of Ad Hoc and Sensor Networks*, 2003.

[4] P. Yi *et al.*, "Distributed Intrusion Detection for Mobile Ad Hoc Networks," *Proc. 2005 Symp. Apps. and the Internet Wksps.*, 2005.

[5] Q. Wang and T. Zhang, "Detecting Anomaly Node Behavior in Wireless Sensor Networks," *Proc. 21st Int'l. Conf. Advanced Info. Networking and Apps. Wksps.*, May 2007, pp. 451–56.

[6] S. Forrest *et al.*, "A Sense of Self for Unix Processes," *Proc. 1996 IEEE Symp. Security and Privacy*, May 1996, pp. 120–28.

[7] S. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion Detection Using Sequences of System Calls," *J. Comp. Security*, vol. 6, 1998, pp. 151–80.

[8] S. Tang, "An Analytical Traffic Flow Model for Cluster-Based Wireless Sensor Networks," *Proc. 1st Int'l. Symp. Wireless Pervasive Computing*, 2006.

[9] Q. Wang and T. Zhang, "Source Traffic Modeling in Wireless Sensor Networks for Target Tracking," *Proc. 5th ACM Int'l. Symp. Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks* , Oct. 2008, pp. 96–100.

[10] X. Zhao *et al.*, "ON/OFF Model: A New Tool to Understand BGP Update Burst," tech. rep. 04-819, USC-CSD, Aug. 2004.

Biographies

QINGHUA WANG [M] (qinghua.wang@ieee.org) is a researcher in the Department of Communications and Networking, Aalto University, Finland. He received his B.Eng. degree in automatic control from Harbin Engineering University, China, in 2002, and Ph.D. (Tekn. Dr.) degree in computer and system sciences from Mid Sweden University in 2010. He was a doctoral student at Xi'an Jiaotong University, China, during 2002–2005. He received the Best Paper Award at the 2007 Annual Conference on Communication Networks and Services. He also received an ERCIM Alain Bensoussan Fellowship for his stay at the Norwegian University of Science and Technology during 2010–2011. He has held visiting positions at the University of Texas at Arlington and at Aalborg University. His research interests include computer networking, wireless sensor networks, wireless control, and network security.