Efficient cluster based intrusion detection in homogeneous and heterogeneous WSN

M. VASIM BABU

Latha Mathavan Engineering College, Madurai (India).
Email: vasim_babu@yahoo.co.in

(Received: August 12, 2010; Accepted: September 19, 2010)

ABSTRACT

Wireless Sensor Networks (WSNs) offer an excellent opportunity to monitor environments, and have a lot of interesting applications in warfare. Intrusion detection in Wireless Sensor Network (WSN) is of practical interest in many applications such as detecting intruder. The intrusion detection is defined as a mechanism for a WSN to detect the existence of inappropriate, incorrect, or anomalous moving attackers. In this paper, I consider the cluster based architecture according to two WSN models: homogeneous and heterogeneous WSN. Furthermore, I derive the detection probability by considering two sensing models: single-sensing detection and multiple-sensing detection. In this Intrusion detection model we are going to track and detect Intrusion in a Homogenous and Heterogeneous Wireless Sensor Networks (WSN) using the intrusion distance and detection probability with various Tracking and Detection models.

Keywords: Intrusion detection, node clustering, node class, node heterogeneity, sensing range, Wireless Sensor Network (WSN).

INTRODUCTION

Wireless sensor networks (WSNs) have become a hot research topic in recent years. Applications include military, rescue, environment monitoring, and smart homes. A WSN is composed of hundreds or even thousands of small, cheap sensors nodes which communicate with one another wirelessly. Sensor nodes typically do not have very much computational power, limiting the kinds of networking protocols and security mechanisms they can employ. Because WSNs are composed of so many nodes, which may be deployed in a hostile environment, replacing batteries is not feasible. Sensor nodes must therefore survive on the small amount of energy in the batteries they are deployed with (typically about 6 amp-hours). This creates a need to conserve energy. Because of the wireless nature of WSNs, security is a fairly difficult issue. Adversaries can easily listen to all the traffic and inject their own, especially if the WSN is deployed in a hostile environment. It is also important that the WSN be robust to losing some of the sensor nodes, because it can be very easy for an adversary to capture any given node. The general network topology is a dense collection of nodes, randomly distributed over some geographic area. Traffic typically goes from all the sensor nodes to a single sink, called the base station (BS), or broadcast traffic. A lot of work is currently being done on routing protocols, and not all of the details are figured out and agreed upon but, in general, routing is multi-hop like an ad-hoc wireless network. Cluster-based routing is a popular idea, because it is possible to exploit the fact that nearby nodes has highly correlated data. In cluster-based routing, the network is divided up into clusters, which consist of a cluster head (CH) and member nodes (MNs). The MNs send their data to the CH, which aggregates the data before sending it out of the cluster toward the base station. In this paper, I will assume that the WSN is in a hostile...
environment, where the deployers have no physical contact with the nodes, but attackers may. An attacker is someone who tries to disturb the functionality of the network in any of the ways. I will also talk about “intrusion detection”. This is defined as identifying an intruder, which is an attacker who has gained control of a node, or injected falsified or repeated packets into the network. This is not to be confused with other “intrusion detection” systems using WSNs, which monitor a physical environment, looking for intruders using a WSN for sensing and collecting information. In this paper, I will explain the special circumstances presented by WSNs, summarize work that has been done in security for WSNs and propose a new cluster-based approach for intrusion detection. In section 3, I will classify the types of attacks on WSNs. In section 4 I will develop the intrusion detection modal. In section 5, analyze the intrusion detection in homogeneous WSN, and section 6 examine the intrusion detection in heterogeneous WSN. In section 7 I will discuss the overall steps for cluster based architecture. In section 8 I will show the simulation of one part of this paper. Finally, I will conclude the paper in section 9.

Related Works

Intrusion detection is one of the critical applications in WSNs and recently several approaches for intrusion detection in homogeneous WSNs have been presented\textsuperscript{8-10}. The focus of these approaches aims at effectively detecting the presence of an intruder. First, the problem is investigated from the aspect of the network architecture. Kung \textit{et al}.\textsuperscript{9} take advantage of a hierarchical tree structure to effectively track the movement of an intruder. The hierarchical tree consists of connected sensors, and is built upon expected properties of intruder mobility patterns (i.e., its movement frequency over a region). Based on the hierarchical tree, it allows efficient record of the intruder's moving information and supports fast querying from the base station. Another tree structure for tracking an intruder, called as a logic object tracking tree, is developed by Lin \textit{et al}.\textsuperscript{10}. The logic object tracking tree reduces the communication cost for data updating and querying, by taking into account the physical network topology. In particular, the logic object tracking tree targets to balance the cost of updating and querying so as to minimize the total communication cost.

Secondly, the intrusion detection problem has been considered from the constraint of saving network resources. For example, Chao \textit{et al}.\textsuperscript{11} have addressed the issue of tracking a moving intruder by power-conserving operations and sensor collaboration. To achieve this, the authors defined a set of novel metrics for detecting a moving intruder, and developed two efficient sleep-awake schemes called PECAS and MESH, to minimize the power consumption. Ren \textit{et al}.\textsuperscript{8} further studied the trade of between the network detection quality (i.e., how fast the intruder can be detected) and the network lifetime. Therefore, the sensor coverage can be carefully designed according to the detection probability with respect to specific application requirements on operating time. The authors then proposed three wave sensing scheduling protocols to achieve the bounded worst-case detection probability. Rather than a static WSN architecture as the above approaches, Liu \textit{et al}.\textsuperscript{12} have modeled the intrusion detection problem in a mobile WSN where each sensor is capable of moving. The authors have given the optimal strategy for fast detection and shown that a mobile WSN improves its detection quality due to the mobility of sensors.

Classification of Attacks

There are four aspects of a wireless sensor network that security must protect

1) Confidentiality
2) Data Integrity
3) Service Availability
4) Energy

The first three are addressed by security systems in wired networks and non-energy-constrained wireless networks, but the fourth is unique to the sensor network application.

I will thus classify the attacks that can be launched by which of these aspects they attack. In this section I will briefly describe the kinds of attacks in each of these categories and the ways in which the nature of the wireless network causes trouble.

Stealing Data (Confidentiality)

When we think of electronic security, this is the first kind of attack that comes to mind. We
want to be able to send messages without enemies being able to figure out the contents. Because of the wireless nature of the WSN, it is easy for an attacker to listen in on all the messages sent in the network, so to maintain confidentiality, the network must encrypt all the messages.

One of the biggest ideas in encryption today is public-key encryption. This is very powerful because it allows one to receive encrypted messages without even sharing a secret key with the sender. But this asymmetry comes at a cost. RSA public key encryption involves exponentiation the message, which can be quite computationally expensive, and is not really feasible in WSNs, where the nodes typically are not capable of doing such computations in a time- and energy-efficient way. But symmetric key encryption mechanisms pose the problem that if any node is apprehended by the attacker, he can look in its memory to find the key and effectively be able to masquerade as any other node (compromising the data integrity) and listen in on any other conversation.

### Altering/Generating False Data (Data Integrity)

Because sensor networks are used to monitor some environment, data integrity is even more important than confidentiality. This is because applications may include tracking objects that physically move through the environment and since attackers can typically see the physical environment, the only thing they could gain from listening in on the data is a sense of where the sensors are located. On the other hand, if they are able to alter to make the data collected by the WSN incomplete or incorrect, the deployed of the WSN will not know what is really going on in the environment he is trying to monitor.

In other networks, the same asymmetric key system that is used for encryption can be used for digital signatures, but this requires a lot of additional overhead. The signature may consist of a lot of additional bytes of data added on to a transmission (which takes additional energy), and verifying the signature can be very computationally expensive. Clearly, different techniques are needed for WSNs.

### Attacks on Service Availability

This class of attacks is not at all concerned with the actual data that is begin sent. Rather, the goal is to make the network not function properly. This can be done by sending bogus routing information (for example advertising a route that does not exist). It can also be done by flooding the network with packets (denial of service attack), or even jamming the frequency at the physical layer. Another interesting type of attack is homing. In a homing attack, the attacker looks at network traffic to deduce the geographic location of critical nodes, such as cluster heads or neighbors of the base station. The attacker can then physically disable these nodes. This leads to another type of attack: the “black hole attack”. In a “black hole” attack, the attacker compromises all the neighbors of the base station, making it effectively a black hole. A final kind of attack on service availability is a desynchronization attack, where the attacker tries to disrupt a transport-layer connection, by forging packets from either side.

### Denial of Sleep Attacks (Energy)

The constrained energy of WSNs adds a new element that can greatly complicate security issues. Because there is a limited amount of energy available and no way to replenish it, it is not sufficient to make sure that bad data is not used. We need to make sure that we do not waste energy listening to or re-transmitting bad packets. This introduces a whole new set of possible attacks. These include constantly sending RTS packets to stop nodes from going to a low power “sleep” state, sending falsified or repeated packets so that nodes waste energy re-transmitting them, or draining the power of a node by forcing it to do excessive computations.

### Detection Model

There are two detection models in terms of how many sensors are required to recognize an intruder: single-sensing detection model and multiple-sensing detection model. It is said that the intruder is detected under the single-sensing detection model if the intruder can be identified by using the sensing knowledge from one single sensor. On the contrary, in the multiple-sensing detection model, the intruder can only be identified by using cooperative knowledge from at least $k$.
sensors ($k$ is defined by specific application requirements). For simplicity of expression, multiple-sensing and $k$-sensing are interchangeable in the following discussion.

In order to evaluate the quality of intrusion detection in WSNs, we define three

**Metrics as follows**

**Intrusion Distance**

The intrusion distance, denoted by $D$, is the distance that the intruder travels before it is detected by a WSN for the first time. Specifically, it is the distance between the point where the intruder enters the WSN and the point where the intruder gets detected by any sensor(s). Following the definition of intrusion distance, the Maximal Intrusion Distance (denoted by $\varepsilon$, $\varepsilon > 0$) is the maximal distance allowable for the intruder to move before it is detected by the WSN.

**Detection Probability**

The detection probability is defined as the probability that an intruder is detected within a certain intrusion distance (e.g., Maximal Intrusion Distance $\varepsilon$).

**Average Intrusion Distance**

The average intrusion distance is defined as the expected distance that the intruder travels before it is detected by the WSN for the first time.

**Intrusion Strategy Model**

We consider two intrusion strategies for the movement of the intruder in a WSN as shown in Fig. 1 and Fig. 2. If the intruder (say, a panzer) already knows its destination before entering the network domain, it follows the shortest path to approach the destination. In this case, the intrusion path is a straight line ($D_1$) from the entering point to the destination as illustrated in Fig. 1. The main idea behind this strategy is that the straight movement causes the least risk for the intruder due to the least area that it has to explore by following a straight line (i.e., the shortest path) toward the destination. The corresponding *intrusion detection area* $S_1$ is determined by the sensors sensing range $r_s$ and intrusion distance $D_1$ as shown in Fig. 1. It is because the intruder can be detected within the intrusion distance $D_1$ by any sensor(s) situated within the area of $S_1$.

On the contrary, if the intruder does not know its destination, it moves in the network domain in a random fashion. We consider the intruder aims to minimize the overlapping on its path. Thus, the intrusion path of the intruder can be regarded as a non-overlapping curved line ($D_2$) and the intrusion area accordingly is a curved band $S_2$ as illustrated in Fig. 2.

In the above two strategies, if the intruder travels the same distance i.e., $D_1 = D_2$, the corresponding intrusion detection areas approximately satisfy $S_1 = S_2$. Therefore, we adopt a straight path in the following discussion, and the analytical results can be directly applied to the case of the curved path. Furthermore, the start point of the intruder can be located in the network boundary.
or a random point inside the network domain. For example, the intruder can be dropped from the air and starts from any point in the network domain.

**Intrusion detection in homogeneous WSN**

In this section, we present the analysis of intrusion detection probability of a heterogeneous WSN in single-sensing detection and multiple-sensing detection models.

### Single Sensing Homogeneous Network

Since all the nodes are identical, the main design objective is to guarantee a certain network lifetime (in terms of number of data gathering cycles), and at the same time ensure that all the nodes expire at about the same time so that there is very little residual energy left behind when the network expires. Hence LEACH uses random and periodic rotation of the cluster heads for load balancing. Role rotation also ensures that a node which is located near the periphery of a cluster is nearer to the cluster head at some other time. Since each node has to be capable of acting as a cluster head, it is necessary for each node to have the hardware capable of performing long range transmissions to the remote base station, complex data computations (if required), and co-ordination of MAC and routing within a cluster. Since all the nodes are capable of acting as a cluster head, the failure of a few nodes does not seriously working of the scheme. Thus the system is robust to node failures.

In the single-sensing detection model, the intruder can be recognized once it moves into the sensing coverage disk of any sensor(s). According to the intrusion strategy, the intruder may access the network domain from any point of the network boundary or a random point in the network domain. When the intruder starts from a point of the network boundary as shown in Fig. 3, given an intrusion distance $D_0$, the corresponding intrusion detection area $SD$ is almost an oblong area. This area includes a rectangle area with length $D$ and width $2rs$, and a half disk with radius $rs$ attached to it. It has $SD = 2 \cdot D \cdot rs + \pi rs^2 / 2$.

### K-Sensing Detection in Homogenous WSN networks

In the k-sensing detection model, an intruder has to be sensed by at least k sensors for intrusion detection in a WSN. The number of required sensors depends on specific applications. For example, at least three sensors’ sensing information is required to determine the location of the intruder. Let $p_k[D=0]$ be the probability that an intruder is detected immediately once it enters a WSN with node density and sensing range $rs$ in k-sensing detection model. It has

$$p_k[D = 0] = 1 - \sum_{i=0}^{k-1} \frac{\pi rs^2 \lambda^i}{2^i i!} e^{-\pi rs^2 \lambda}.$$

Let be the probability that the intruder is detected within the maximal intrusion distance $\xi$ in a K-sensing detection model for the given homogeneous WSN. Then $p_k[D \leq \xi]$, can be calculated as

$$p_k[D \leq \xi] = 1 - \sum_{i=0}^{k-1} \frac{(S_{\xi} \lambda)^i}{i!} e^{-S_{\xi} \lambda},$$

where $S_{\xi} = 2\xi rs + \pi \lambda \xi^2 / 2$.

is the intrusion detection area with respect to the maximal intrusion distance $\xi$. If there are at least k sensors in the area $S_{\xi}$, the intruder can be sensed by the k sensors, and the k sensors could collaborate with each other to recognize the intruder. From (3),

$$P(i, S_{\xi}) = \frac{(S_{\xi} \lambda)^i}{i!} e^{-S_{\xi}}.$$
Denotes the probability that $i$ sensors are located in the area of $S_i$. Then
\[ \sum_{i=0}^{k-1} P(i, S_i) = \sum_{i=0}^{k-1} \frac{(S_i \lambda)^i}{i!} e^{-S_i \lambda}, \]
is the probability that less than $k$ sensors are located in the area. Thus, the complement of
\[ \sum_{i=0}^{k-1} P(i, S_i), \quad 1 - \sum_{i=0}^{k-1} \frac{(S_i \lambda)^i}{i!} e^{-S_i \lambda} \]
is the probability that there are at least $k$ sensors located in the area $S_i$. If this is the case, the intruder can be sensed by at least $k$ sensors from the WSN with probability
\[ 1 - \sum_{i=0}^{k-1} \frac{(S_i \lambda)^i}{i!} e^{-S_i \lambda} \]
before it travels a distance of $\xi$. Finally, the probability that the intruder is detected within the maximal intrusion distance in k-sensing detection model can be derived as
\[ p_k[D \leq \xi] = 1 - \sum_{i=0}^{k-1} \frac{(S_i \lambda)^i}{i!} e^{-S_i \lambda} \]

**Intrusion detection in Heterogeneous WSN**

**Single Sensing Heterogeneous Networks**

Heterogeneous sensor networks use two or more types of nodes with different functionalities. For example, the authors in [4] propose using two types of nodes; type 0 nodes which act as pure sensor nodes, and type 1 nodes which act as the cluster head nodes. Some of the salient features of such networks are:

1. Since the cluster head nodes are predetermined, and the sensor nodes use single hop communication to reach the cluster head nodes, the sensor nodes near the periphery of the cluster have the highest energy expenditure among all the sensor nodes. It is this worst case energy expenditure that has to taken into account in battery energy dimensioning. Thus there is a waste of energy due to the residual battery energy of the sensor nodes that are near the cluster heads.

2. Since only the cluster head nodes bear the responsibility of transmitting to the distant base station, the rest of the nodes can be designed with simple hardware that enables short range communication. Thus the hardware complexity is limited to only a few nodes.

3. A cluster head node serves as the fusion point, as well as the command center of its cluster. As a result when a cluster head node fails, all the sensor nodes in that cluster have to be re-assigned to other neighboring clusters. In the extreme case, it is possible that the entire cluster head nodes Might fail, thereby bringing down the entire network. Thus the system is less robust to node failure as compared to a homogeneous sensor network.

**K-Sensing in a Heterogeneous WSN**

In the k-sensing detection model of a heterogeneous WSN with two types of sensors, at least $k$ sensors are required to detect an intruder. These $k$ sensors can be any combination of Type I and Type II sensors.

- Type I sensor that has a larger sensing range $r_s$, as well as a longer transmission range $r_t$.
- Type II sensor that has a smaller sensing range $r_s^2$, as well as a shorter transmission range $r_t^2$.

For instance, if three sensors are required to detect an intruder for a specific application, the intruder can be detected by any of the following sensor combinations:

1. Three Type I sensors,
2. Three Type II sensors,
3. One Type I sensor and two Type II sensors,
4. Two Type I sensors and one Type II sensor.

Let $p_k[D \leq \xi]$ be the probability that an intruder can be immediately detected once it enters the given heterogeneous WSN in the k-sensing detection model. It has
\[ p_k[D_k = 0] = 1 - \sum_{i=0}^{k-1} \frac{(S_i \lambda)^i}{i!} e^{-S_i \lambda} \]

Let $p_k[D \leq \xi]$ be the probability that the intrusion distance is less than $\xi (\xi < 0)$ in the k-sensing detection model, $\xi$ is the maximal intrusion distance allowable for an intruder to move in the given heterogeneous WSN. It has
Let \( E_k(D_h) \) be the average intrusion distance under the k-sensing detection model in the given heterogeneous WSN. Then

\[
\hat{\gamma}_k(D_h, \xi) = 1 - \sum_{i=k}^{k-1} \sum_{j=0}^{i} P_3(j, S_j) P_3(m-j, S_m),
\]

where \( S_j = 2\xi \rho_k + \frac{\psi_k^2}{2}, (1 = 1, 2) \).

Design Steps for Cluster Based Architecture

1. Topology Design
   - 400 Nodes Topology
   - Cluster based Architecture
   - BS control
   - MICA Node
2. Cluster Based Architecture (LEACH)
   - LEACH Implementation (Low-energy adaptive clustering hierarchy)
3. Change MAC Layer
   - Change Buffering Time (Normal Node)
   - Route Table Refreshment time (Normal Node)
4. Increase Route table size (Normal Node, Watch Dog)
   - Neighbors Selection Algorithm (K-neighbors)
   - Select Best node from group of neighbors node
5. Neighbors Filtering (Bloom Filter)
   - Filter unselected bandwidth nodes (Geometrical Based) (Watch Dog)
6. Generate Scenery
   - Generate Movement and Application flow
7. Trace and Draw Graph
   - End – End Delay
   - Watch Dog Probability
   - Performance

Cluster-Based Security

As we have seen in the G-MAC example, clusters can provide major advantages in sensor network security. In the case of G-MAC, we let the GS be the CH. The CH can also monitor the traffic coming from each MN and figure out if any of them have been compromised. It can then blacklist these nodes, isolating them from the network. In case a CH is compromised, MNs must also have the ability to decommission the CH if there are enough MNs that agree to do so. This will defend against homing attacks. It is critical that several nodes agree to decommission the cluster head, because if only a few nodes are compromised, they should not be able to take down the cluster head. When a node is removed, its transmissions will be ignored, and nothing will be sent to the node. However, it will still be able to hear and understand broadcast traffic. This can be fixed with a slight modification to iTESLA in which the function \( F(\cdot) \) is not fixed at deployment time, but is rather a function that takes two arguments: a cluster key and key for the broadcast Kn. We can then generate keys using \( KN = F(KN+1) \). (3) When a node is removed, the CH can generate a new, and inform each of the remaining nodes individually. An analysis of this protocol modification may be included in future work. When a node is monitoring another node, there are two ways in which it can detect misbehavior: anomaly detection and signature detection [4]. In anomaly detection, the monitoring node looks for deviations from typical behavior. This has a high probability of false alarm, so it is not often used [4]. Signature detection, on the other hand, looks for particular types of misbehavior. This leaves it susceptible to new creative types of attacks, but there are not really a lot of new different actions a misbehaving sensor node could take. The typical actions in this application would be dropping packets, duplicating packets, or causing collisions [6]. We discussed the mechanisms for intrusion detection in 5 and 6, but while we know how to verify packets, we need to determine the optimal place to do the checking. It turns out that the CH is the right place to check for authenticity, as discussed in [4]. The simplest idea for intrusion detection would be to have all nodes monitor each other in promiscuous mode, meaning they listen in on all transmissions, even if they are not the recipient, but this wastes way too much energy. The other extreme would be to only check the packets at the end (base station or its neighbors), but by waiting to check the packet, we run the risk of transmitting a bad packet far too many times, consuming valuable resources. The packet should be checking only a few times, and early in the path. [4] Suggests that this should occur at the cluster head. All packets should be first sent to the cluster head, which checks their authenticity before...
forwarding them out to the rest of the network. But a malicious node can avoid this by neglecting to send its packet to the cluster head first, instead opting to send it out to another cluster. To compensate for this, every node should have a probability, p, of sending a packet to its cluster head to be checked. The higher the p, the more energy consumed in checking, and the longer the route is made, but the more likely we are to catch malicious packets. Selecting the proper p may possible be done using the method.

The Clustering Algorithm

The clustering algorithm serves to mitigate the problems associated with a low cost wireless network. The algorithm begins with the assumption that each node has been assigned a unique identification number in the pre-deployment phase. These unique IDs will be appended to any message passed through the network so that each successive node knows which node sent the received message. A flow-diagram of the clustering algorithm is given in Appendix B. Before beginning a discussion of the clustering algorithm used, the reader should understand that the following outlines the actions that the wireless sensor network takes after event detection. The clustering algorithm is not, however, responsible for event detection itself, as this could be achieved in a variety of ways. The mission profile to which the WSN is tailored in the pre-deployment phase would ensure proper event detection.

Upon activation the nodes begin the declaration phase, in which each node broadcasts its unique ID to all other nodes within range. When a node receives the declaration transmission from its neighbors, it records the nodes within its immediate surroundings. Next, the nodes begin to establish a hierarchy for message handling by assigning themselves different classes of nodes.

LEACH: Low-Energy Adaptive Clustering Hierarchy

LEACH is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network. In LEACH, the nodes organize themselves into local clusters, with one node acting as the local base station or cluster-head. LEACH includes randomized rotation of the high-energy cluster-head position such that it rotates among the various sensors in order to not drain the battery of a single sensor. In addition, LEACH performs local data fusion to “compress” the amount of data being sent from the clusters to the base station, further reducing energy dissipation and enhancing system lifetime.

Sensors elect themselves to be local cluster-heads at any given time with a certain probability. Each node makes its decision about whether to be a cluster-head independently of the other nodes in the network and thus no extra negotiation is required to determine the cluster-heads. These cluster head nodes broadcast their status to the other sensors in the network. Each sensor node determines to which cluster it wants to belong by choosing the cluster-head that requires the minimum communication energy. Once all the nodes are organized into clusters, each cluster-head creates a schedule for the nodes in its cluster. This allows the radio components of each non-cluster head node to be turned off at all times except during its transmit time, thus minimizing the energy dissipated in the individual sensors. Once the cluster head has all the data from the nodes in its cluster, the cluster-head node aggregates the data and then transmits the compressed data to the base station.

The system can determine, a priori, the optimal number of clusters to have in the system. This will depend on several parameters, such as the network topology and the relative costs of computation versus communication. If there are fewer than optimal number of cluster heads, some nodes in the network have to transmit their data very far to reach the cluster-head, causing the global energy in the system to be large. If there is more than optimal number of cluster heads, the distance nodes have to transmit to reach the nearest cluster-head does not reduce substantially, yet there are more.

Cluster-heads that have to transmit data the long-haul distances to the base station, and there is less compression being performed locally. In addition to reducing energy dissipation, LEACH successfully distributes energy-usage among the nodes in the network such that the nodes die randomly and the same rate.
Node Classes

Within the wireless sensor network, there are four (really three) different types of nodes: regular nodes, gateways, gateway pairs, and cluster heads (CHs). After the nodes have declared themselves and received declarations from their neighbors, they then “Vote” by broadcasting the highest node ID number received. Any node that receives a vote is promoted to cluster head. For example, if nodes 1, 3, 4 and 9 are all within range of each other, all four nodes would broadcast a vote for node 9 (including node 9 itself). It is important to note that these MicaZ nodes have a limited range, compared to the size of the field; thus in any given field, there will be multiple cluster heads. If any single node is within range of two or more cluster heads, it must be a Gateway. This is due to the fact that any node receiving a nomination must become a cluster head. Gateways are the nodes that link two cluster heads together that are out of range of each other.

Node Clustering

In a WSN, the communication cost is often several orders of magnitude higher than the computation cost. Therefore, minimizing energy consumption in communication and maximizing the system lifetime has been a major design goal for WSNs. Cluster-based hierarchical architecture has been emerging as an essential paradigm to minimize communication cost in WSNs.

Node clustering is to decompose the entire network into multiple clusters to form a cluster-based WSN. It actually results in a three layer hierarchical architecture in the WSN, in which each regular/member sensor (MS, in the bottom level) transmits data to its cluster head (CH, in the middle level) instead of the faraway base station (BS, in the top level). This communication paradigm hence reduces the number of sensors that participate in the long distance communication. It eventually improves the energy efficiency of the overall system. The basic idea of clustering is performed by assigning each sensor to a CH in order for optimizing the resource allocation and spatial reuse. Here, we investigate two popular clustering technologies: $k$-clustering and balanced clustering, and compare their energy efficiency in lattice WSNs.

It is a framework in which the WSN is divided into non-overlapping sub-networks (i.e., clusters), where each member sensor in the cluster is at most $k$ hops away from the CH.
Fig. 6 illustrates the cluster instances formed by 2-clustering technology for lattice WSNs. The dark/red nodes represent the CHs, the blank nodes denote its corresponding member sensors, and the grey nodes represent member sensors belonging to other clusters. In all the three cases, the CH is located in the center of each cluster, and the number of member sensors in each cluster for square, triangle, and hexagon pattern is 13, 19, and 10 respectively. Furthermore, the resulting cluster form for square pattern deployed WSN is still square-formed, while that of both triangle and hexagon pattern deployed WSNs is hexagonally shaped. All these regular shapes can also be repeated over a continuous WSN domain. Similar observations can be found in Fig. 7 for 1-clustering technology in lattice WSNs.

**Balanced Clustering**

The balanced clustering technology aims at grouping neighboring sensors such that each cluster is balanced in terms of the number of sensors.

The existing balanced clustering method is based on min-weight matching in optimizing the total spatial distance between the member sensors and the CH, and minimizing the total distance between them to reduce the energy consumption in communication. This scheme is also efficient in balancing the overall system load.

Suppose about $N_c = 25$ adjacent sensors should be grouped together to form a cluster in lattice WSNs. Fig. 2.9 shows the cluster instances resulting from balanced clustering with a cluster size of 25. It is clear that the resulting clusters are also in regular shape. That is square, rectangular and hexagon for the square, triangle, and hexagon pattern based deployment. They can be repeated in the entire network domain.

**Simulation and verification**

To validate the analytical and simulation model, we have carried out simulations using ns-2 simulator.

As expected, Fig. 9 shows that the intrusion detection probability in the heterogeneous WSN increases at a much faster rate than in the homogeneous WSN, as the number of Type I sensors is increased. Especially in the more demanding multiple sensing detection case, the intrusion detection probability increase seven more quickly in heterogeneous WSN than in homogeneous case.
It can be observed from Fig. 10 that the average intrusion distance drops dramatically with the increase of type I sensors. This is because the increase of the more powerful type I sensors significantly enhances the network sensing coverage and improves the network detection probability. This further validates our intuition that the introducing of more powerful sensor can improve the network performance in heterogeneous WSNs at a much faster rate than in homogeneous WSNs.

CONCLUSION

This paper analyzes the intrusion detection problem in both homogeneous and heterogeneous WSNs with cluster based approach by characterizing intrusion detection probability with respect to the intrusion distance and the network parameters (i.e., node density, sensing range, and transmission range). Two detection models are considered: single-sensing detection and multiple-sensing detection models. The analytical model for intrusion detection allows us to analytically formulate intrusion detection probability within a certain intrusion distance under various application scenarios. Moreover, we consider the network connectivity and the broadcast reach ability in a heterogeneous WSN. This work provides insights in designing homogeneous and heterogeneous WSNs and helps in selecting critical network parameters so as to meet the application requirements.

REFERENCES

1. I.F. Akyildiz, W. Su, Y. Sankarsubramaniam and E.Cayirci, “Wireless Sensor Networks, a survey”, Computer Networks, 38: 393-422 (2002).
2. V.mhatre and c.rosenberg “Design guidelines for Wireless sensor networks: communication, clustering and aggregation”, to appear in adhoc network journal, Elsevier science available from http:/web.ics.purdue.edu/mhatre/adhoc.pdf
3. S.Bandyopadhyay and E. Coyle, An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks, IEEE Info-com, San Francisco, CA, (2003).
4. Chien-chung su ,ko-Ming Chang,Yau-Hwang Kuo,”The New Intrusion Prevention and Detection Approaches For Clustering Based Sensor”,IEEE communication Society/ WCNC (2005).
5. Yanchao Zhang, “Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks”, IEEE Journal on Selected Areas in Communications, 24(2): (2006).
6. Hu.w. Chou, C.T., Jha.s., and Bulusu,N:deploying long-lived and cost-
effective Hybrid sensor network. Elsevier ad-

7. C.-Y. Lin, W.-C. Peng, and Y.-C Tseng, "Efficient in-Network Moving Object Tracking in Wireless Sensor Networks," IEEE Trans. Mobile Computing, 5(8): 1044-1056 (2006).

8. S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, "Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks," IEEE Transaction on Parallel and Distributed Systems, 18(3): 334-350 (2007).

9. H. Kung and D. Vlah, "Efficient location tracking using sensor networks," in IEEE Wireless Communications and Networking Conference, ser. 3(3): 1954-1961 (2003).

10. C.-Y. Lin, W.-C. Peng and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," IEEE Transactions on Mobile Computing, 5(8): 1044-1056 (2006).

11. C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom) (2004).

12. B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in Proceedings of the 6th ACM International Symposium on Mobile Ad

13. K.-W. Fan, S. Liu, and P. Sinha, "On the potential of structure-free data aggregation in sensor networks," in 25th IEEE International Conference on Computer Communications (InfoCom), (2006).

14. W. B. Heinzelman, A. P. Chandrakasan, and H. Balkrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, 1(4): (2002).

15. M. Khan, B. Bhargava, S. Agarwal, L. Lilien, and Pankaj, "Self-configuring node clusters, data aggregation, and security in microsensor networks." [Online]. Available: http://raidlab.cs.purdue.edu/papers/self-con.pdf

16. S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multihop wireless networks," in Sensor Networks, (2008).

17. Y. Fernandess and D. Malkhi, "K-clustering in wireless ad hoc networks," in the Second ACM International Workshop on Principles of Mobile Computing, (2002).

18. S. Ghiasi, "Optimal energy aware clustering in sensor network," [Online]. Available: citeseer.ist.psu.edu/ghiasi_02optimal.html (2002).