Dynamic Clustering to Detect Malicious Nodes in Wireless Sensor Network

Deepali Virmani a, Ankita Soni b, Manas Hemrajani c, Shringarica Chandel d
Bhagwan Parshuram Institute Of Technology, Delhi, India
deealivirmani@gmail.com, ankita24soni@gmail.com, manas.hemrajani@gmail.com, shringaricachandel37@gmail.com

Abstract—Wireless Sensor Networks are vulnerable to many types of attacks that disturb the normal behaviour of the network. These attacks occur because of the presence of malicious nodes in the network that advertise false information among the nodes and disrupt normal operations of the network. This paper proposes a mechanism to detect malicious nodes in the network. In this paper we overcome the limitations of conventional Watchdog Mechanism for detecting malicious nodes in the network. We propose a dynamic clustering technique to detect the malicious nodes. This technique improves network security, saves energy which leads to efficient use of power and helps in removal of malicious nodes from the network.

Index Terms—Wireless Sensor Network, Malicious node, Watchdog, Dynamic Cluster, Cluster head node.

I. INTRODUCTION

Wireless Sensor Networks consists of numerous sensors which monitor physical environmental conditions accurately and collectively pass data to the destination [1]. The sensors are randomly dispersed in the network and communicate using wireless transceivers [2]. Applications of Wireless Sensor Networks can be classified into many categories such as battlefield surveillance, flood detection, forest fire detection etc [3]. But they are vulnerable to many attacks like black hole attack, denial of service, hello flood attack, worm hole attack etc [2]. These attacks are caused by the malicious nodes in the network and pose serious threats to the security of Wireless Sensor Networks.

In this paper we attempt to detect the malicious nodes in the network using dynamically selected cluster head node. The head node is selected on the basis of highest available energy of the nodes and can sense the activities of all the sensor nodes within its cluster. Using this method we are able to overcome the limitations of Watchdog mechanism and hence enhance the security and reliability of the network.

The remainder of the paper is structured as follows: Section 2 presents related works on malicious node detection. Section 3 explains the Watchdog mechanism and its limitations. Section 4 provides a detailed description of our proposed model. Finally, we conclude the paper in Section 5.

II. RELATED WORK

In this section we briefly state some of the previously proposed mechanisms for detection of malicious nodes.

© Elsevier, 2013
D. Liu proposed a mechanism[4] where the beacon nodes first detects the malicious beacon signals and then the corresponding malicious beacon nodes. The scheme consists of two types of nodes: Detecting beacon node (that detects the malicious nodes) and Target beacon node (the node being detected). During the detection, the detecting node uses a Detecting ID. This ID is recognized as a non-beacon node ID so that the attacker cannot distinguish between the detecting beacon node and non-beacon node. When the target node receives a request message from the detecting node, it sends back a beacon packet that includes its own location. The detecting node estimates the distance to the target node using beacon signals and also calculates the distance based on its location and the target node’s location. If the difference between the estimated and the calculated distance is larger than the maximum distance error, the detecting node can tell that the received beacon signal is malicious.

In [5], a mechanism, particularly for Hello Flood attack and Wormhole attack was proposed for the detection of malicious nodes. This mechanism detects the malicious nodes based on the signal strength and demographics. The actual signal strength of the received signal is compared with the calculated one, and if the difference is greater than a given threshold, then the node is declared malicious. This scheme has its own limitations. First, it requires a large overhead for transmitting data. Second, nodes consume maximum power as all nodes monitor and process data all the time.

In [6], Atkali et al designed the Weighted Trust Evaluation Technology for detection of malicious nodes. This technology consists of three layer network architecture with sensor nodes (SN) at the lowest level, forwarding nodes (FN) at the next level and Access Points or base stations at the highest level. The sensor nodes can misbehave and transmit wrong information to the higher layers. To keep a check on the sensor nodes, FN calculates an aggregation result on the basis of weight assigned to each SN and the information provided by each SN. The aggregation result is then compared with the output of SN. If the aggregation result is not consistent with the output of SN, then that SN is regarded as malicious.

In [7], S.Marti et al proposed the Watchdog mechanism which works on the basis of overhearing the transmissions of the next hop neighbours. A node that comes under the radial transmission of another node can overhear all the transmissions of its neighbours. Watchdog is explained in detail in Section 3 along with its limitations.

In [8], an improved watchdog technique was proposed. In this technique the cluster head node was fixed and used a buffer to store the packets sent by the nodes. As the next node received the packet, it would be compared with the packet stored in the buffer. If the packets were same, first packet stored in the buffer would be deleted. Otherwise a warning signal would be issued signalling the maliciousness of the sending node. This technique could overcome four out of six limitations of Watchdog technique.

III. CONVENTIONAL WATCHDOG

A. Introduction

Watchdog Mechanism is a widely used mechanism for detection of malicious nodes in ad-hoc networks. This mechanism identifies the malicious nodes by eavesdropping on the transmission of the next hop [9]. Consider the path – A → B → C as shown in Fig. 1. A node, like node A transmits a packet to its neighbour B and keeps a copy of the sent packet in its buffer. Then, A overhears B to check whether it forwards the packet to its next hop neighbour C. If the packet overheard from B matches to that stored in the buffer then B has really forwarded the packet. If the packet is not overheard or the packet doesn’t match to that stored in the buffer within a limited time, then failure count for B is incremented. If the failure count exceeds threshold, then B is declared malicious.
B. Limitations of Watchdog Mechanism
In [7], S.Marti et al pointed out the limitations of Watchdog as stated below.

1. Ambiguous Collision: A may not be able to overhear B’s transmission to C if packets are sent to A simultaneously. This may misguide A to conclude that B is malicious, which may be wrong.
2. Receiver Collision: A can only overhear B and tell if it transmitted the packet, but A cannot determine whether C received the packet. In this case, if a collision occurs at C then malicious node B could skip retransmitting the packet or it could cause the transmitted packet to collide purposefully at C when C is transmitting.
3. False Misbehavior: A malicious node, A can falsely report that B is misbehaving, when B is actually not misbehaving.
4. Limited Transmission Power: A malicious node, like B can deliberately drop packets by limiting its transmission power in such a way that A can only overhear whether B transmitted the packet but cannot overhear whether C received it.
5. Partial Dropping: A malicious node, like B can drop packets at a lower rate than the threshold value of malicious nodes. So, even though B is malicious, it’s not considered as malicious.
6. Collusion: If two nodes A and B both are malicious, they can circumvent the source node. A can transmit the packet to B and B can drop the packet intentionally. Since the source node has limited transmission range, it will never know that B dropped the packet. So, A and B both deceives the source node.

IV. PROPOSED MODEL
In our proposed model, we consider a network that consists of clusters where each cluster contains a head node. The cluster head node has the highest energy in the cluster and is dynamically assigned based on the energy considerations of all the nodes in the cluster. After the Energy Timer, $T_e$ goes off, all the nodes broadcast their energy and the node that has the highest available energy is selected as the Cluster Head Node. A centralized table is maintained in memory that stores the node id, packet id, suspicion count and cluster head of each node (See Table 1). The cluster head node has access to this centralized table for limited time duration.

Let M be a Cluster Head Node that monitors the transmissions of nodes S, A, B and D which are within its transmission range. Source node S sends a packet to A and as soon as the packet is received by the first node A, the Forwarding Timer is initialized to some real time delay. The packet must be forwarded within the real time delay, otherwise the suspicion count for node A will be incremented and M will ask the source to resend the packet by sending a request to resend(RTR) to the source (See Table 1, Fig. 2).

If the node forwards the packet before the timer goes off, then the packet id, PID, will be checked at the next node. We consider only the Packet id and not the complete contents of the packet so as to avoid ambiguous collision. Comparing only the P.ID saves time and reduces probability that some other node starts transmitting simultaneously while the head node monitors other packets. If the node tries to change the contents of the packet then PID of the received packet will be different than the source PID. In this case, the suspicion count will be incremented for the node which sent the packet and M will again send request to resend (RTR) to the Source (See Fig. 3).

<table>
<thead>
<tr>
<th>NODE ID</th>
<th>S.PID</th>
<th>R.PID</th>
<th>SUSPICION COUNT</th>
<th>CLUSTER HEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>5</td>
<td></td>
<td>0</td>
<td>M</td>
</tr>
<tr>
<td>A</td>
<td>#5</td>
<td>1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>#6</td>
<td>0</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

Once the Suspicion count of a node reaches the threshold value, the node will be declared malicious and the message will be broadcasted in the cluster.

If the packet reaches its destination D successfully, an ACK is sent by M to S to inform it about the delivery of the packet (See Fig. 4).

A. Proposed Algorithm
Assumptions: The following assumptions are considered for the realization of our proposed model.
1. Source Node S is not malicious.
2. When a Node is selected as the Cluster head node then it can only overhear the transmissions and cannot receive packets.

Algorithm
1. S is the source node from where the packet is sent and D is the destination where the packet is to be delivered.
2. $T_E$ is the Energy timer initialized to some value $\alpha$.
3. $T_F$ is the Forwarding timer which checks whether the packet has been forwarded by the node within a certain time. It is initialized to a real time delay $\beta$.
4. $SC_i$ is the suspicion count of the $i^{th}$ node and is incremented in two cases:
   a. When the forwarding timer goes off and the packet has not been forwarded
   b. When the packet has been modified by the intermediate nodes.
5. K is the threshold value for the suspicion count. For environments which cannot tolerate packets being dropped, K is kept less. When the $SC_i$ reaches K, the $i^{th}$ node is broadcasted as malicious.

6. S.PID is the Packet ID of the packet sent by the source. R.PID is the Packet ID of the packet received by the intermediate nodes. The S.PID and R.PID are kept in centralized memory buffer and compared when the packets are received. When the S.PID and R.PID are not same, SC of $i^{th}$ node is incremented by 1.

7. When the packet reaches Destination D then an Acknowledgement, ACK, is sent from the Head node to the Source S. When the packet gets dropped or the contents get modified then the Request to Resend, RTR, is sent to the Source S.

(See Fig. 5 for flowchart of proposed algorithm).

B. Advantages of the Proposed Algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INT $T_a = \alpha$</td>
</tr>
<tr>
<td>2.</td>
<td>S $\rightarrow A$ (S sends packet to A)</td>
</tr>
</tbody>
</table>
| 3.   | Packet reaches intermediate node  
|      | 3.1 IF packet is sent by Source THEN  
|      | Go to Step 5  
|      | 3.2 Else go to step 4 |
| 4.   | Compare packet ID  
|      | 4.1 IF S.PID == R.PID THEN  
|      | 4.1.1 Forward the packet  
|      | 4.1.2 Go to step 5  
|      | 4.2 IF S.PID \neq R.PID THEN  
|      | 4.2.1 Go to step 7 |
| 5.   | IF receiving node == D THEN  
|      | 5.1 M $\rightarrow S$ (M sends ACK to S)  
|      | ELSE  
|      | 5.2 Go to step 6 |
| 6.   | INT $T_b = \beta$  
|      | 6.1 IF packet forwarded by node && $T_b \neq 0$ THEN  
|      | 6.1.1 Go to step 3  
|      | 6.2 IF packet not forwarded by node && $T_b = 0$ THEN  
|      | 6.2.1 Go to step 7  
|      | 6.3 IF packet forwarded by node && $T_b = 0$ THEN  
|      | 6.3.1 Go to step 7 |
| 7.   | $SC_i^{++}$ (Increment $SC_i$ by 1 for $i^{th}$ node) |
| 8.   | IF $SC_i = K$ THEN  
|      | 8.1 Node i = malicious. Broadcast i as malicious.  
|      | ELSE  
|      | 8.2 Send RTR to S.  
|      | 8.3 Go to step 2 |
| 9.   | IF $T_a = 0$ THEN  
|      | 9.1 Request all nodes to broadcast $E_a$  
|      | 9.2 Node with max $E_a$ is selected as the new Cluster head node  
|      | 9.3 Go to Step 1 |
1. The proposed model detects malicious nodes by selecting a Head Node dynamically which overhears all the transmissions and checks PID after each transmission.

2. The model enhances the security by sending ACK and RTR to the source. It improves the power consumption by not relying only on a single node to perform overhearing. If a single node performs all the overhearing and comparisons then it will run out of battery very soon and network lifetime will reduce.

3. Overcoming False Misbehavior: The Head node reports whether the nodes are malicious or not and it is not left upon the intermediate nodes to judge it. Thus, it is unlikely that a single head node which is appointed for a small duration will be incrementing the suspicion count repeatedly for a single node and reporting it as malicious.

4. Overcoming Ambiguous Collision: The packet is overheard and dropped as soon as the P.ID is checked. This allows node to overhear other packets and avoid collisions. If the entire packet contents are checked it will lead to more time being spent on a single packet and increases probability that some other node might start forwarding.

5. Overcoming Partial Dropping: Each time a packet is sent after a real time delay by the node or not sent at all, the SC is incremented by 1. If the packets are repeatedly dropped then SC will reach K and node will be broadcasted as malicious.

6. Overcoming Collusion: The two intermediate nodes being malicious and not reporting a malicious node is avoided in the proposed model by delegating the detection to the Head Node.

Fig. 5. Flow Chart for proposed scheme
V. CONCLUSION

Security of Wireless Sensor Network is important to prevent attacks by malicious nodes and to increase the reliability of the network. In this paper we have attempted to detect the malicious sensor nodes by using dynamically selected cluster head node. Dynamic clustering overcomes the limitations of Watchdog mechanism, increases the network security and is efficient in terms of energy consumption.

REFERENCES


