ISSN:0975-9646

A Strategic Approach for Bypassing the Intruder and Sending Data to Destination using Three –Tier Framework in Sensor Networks

Veera RaghavaRao Atukuri , Venkata Pavani S,Vasanthi B ,Chandrika M,Naga Swetha M

Department of Computer Science and Engineering

Malineni Lakshmaiah Women's Engineering College, Pulladigunta (V), Vatticherukuru (M), Guntur (Dt) – 522017.

Abstract – In sensor networks, an intruder (i.e., compromised node) identified and isolated in one place can be relocated and/or duplicated to other places to continue attacks; hence, detection and isolation of the same intruder or its clones may have to be conducted repeatedly, wasting scarce network resources. Detecting a compromised sensor, whose memory contents have been tampered, is crucial in these settings, as the attacker can reprogram the sensor to act on his behalf. In the case of sensors, the task of verifying the integrity of memory contents is difficult as physical access to the sensors is often infeasible. Often There is solution to find and detect the intruder in sensor network using a three-tier framework, consisting of a verifiable intruder reporting (VIR) scheme, a quorum based caching (QBC) scheme for efficiently propagating intruder reports to the whole network, and a collaborative Bloom Filter (CBF) scheme for handling intruder information locally. Which is useful only to find the attacker in the network and if there is an attacker we didn't sent the data..In this paper, we propose a bypassing the data when the intruder is attacked in the sensor network. By using these three schemes and finding the different routes in the network and send the data to destination.

INTRODUCTION

A Sensor network is a group of specialized transducers with a communications infrastructure intended to monitor and record conditions at diverse locations. Commonly monitored parameters are temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, chemical concentrations, pollutant levels and vital body functions. Routing**.** Since a distributed network has multiple nodes and services many messages, and each node is a shared resource, many decisions must be made. There may be multiple paths from the source to the destination. The main performance measures affected by the routing scheme are throughput (quantity of service) and average packet delay (quality of service).

Routing methods can be fixed (i.e. pre-planned), adaptive, centralized, distributed, broadcast, etc. Perhaps the simplest routing scheme is the token ring [Smythe 1999]. Here, a simple topology and a straightforward fixed protocol result in very good reliability and precomputable QoS. A token passes continuously around a ring topology. When a node desires to transmit, it captures the token and attaches the message. As the token passes, the destination reads the header, and captures the message. In some schemes, it attaches a 'message received' signal to the token, which is then received by the original source node. Then, the token is released and can accept further messages. The token ring is a completely decentralized scheme that effectively uses TDMA. Though this scheme is very reliable, one can see that it results in a waste of network capacity. The token must pass once around the ring for each message. Therefore, there are various modifications of this scheme, including using several tokens, etc.

LITERATURE SURVEY

The proposed framework and Routing is composed of the following three tiers of entities and routing schemes:

On *the* ***top tier*** is a dedicated membership server (DMS), which aggregates and periodically disseminates intruder information to the whole network. Due to its critical role, the DMS may become an attractive target of attacks. Specifically, the adversary may locate the DMS and then either compromise the DMS directly or block the communication between the DMS and the rest of the network. To protect the DMS, it is not connected to the network all the time. Instead, it goes online every now and then at different places randomly. The protection makes it hard for the adversary to trace, attack, or isolate the DMS.

On *the middle tier* are intruder information caches (IICs), which are a small number of sensor nodes picked from all sensor nodes in the network. They temporarily cache

new intruder information when the DMS is offline. As ordinary sensor nodes, they could be compromised by the adversary. If compromised, the intruder information cached by these IICs may be removed or modified, which is addressed in our solution through (i) verifying intruder information to prevent faking or fabricating, and (ii) duplicating intruder information to maintain high availability of the information.

On *the bottom tier* are ordinary sensor nodes, which collaboratively identify intruders and report intruder information to IICs. Sensor nodes maintain their own intruder information based on the periodical updates disseminated by the DMS, and collaboratively determine the legitimacy of sensor nodes who join their neighborhoods; they may also query IICs to obtain latest intruder information when necessary. Fig. 1. Overview of the proposed Framework (Dark circles represent IICs and white circles represent other sensor nodes) To summarize, interactions between these entities include: Sensor nodes collaboratively generate intruder reports that can be verified by any other nodes, and send them to a certain set of IICs. Every time interval *l*, the DMS queries IICs to collect the reports for

intruders that have been identified since the previous query, and then disseminates the IDs of these intruders to all sensor nodes in a secure manner. Upon receiving it, every sensor node records these intruders; if the sensor node is also an IIC, it removes these intruders from its cache (since it is not necessary to cache the information). When a node joins a neighborhood, the neighbors can use their own knowledge about identified intruder to determine if the new arrival is intruder or not. If the neighbors need more accurate intruder information (i.e., the information about intruders detected since last dissemination by the DMS), they may query a certain set of IICs to obtain

Fixed routing schemes often use Routing Tables that dictate the next node to be routed to, given the current message location and the destination node. Routing tables can be very large for large networks, and cannot take into account real-time effects such as failed links, nodes with backed up queues, or congested links.

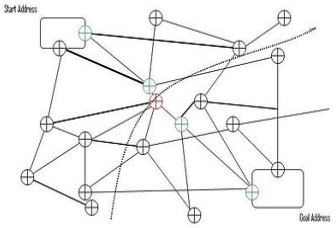
Adaptive routing schemes depend on the current network status and can take into account various performance measures, including cost of transmission over a given link, congestion of a given link, reliability of a path, and time of transmission. They can also account for link or node failures.

Routing algorithms can be based on various network analysis and graph theoretic concepts in Computer Science (e.g. A-star tree search), or in Operations Research [Bronson 1997] including shortest-route, maximal flow, and minimum-span problems. Routing is closely associated with dynamic programming and the optimal control problem in feedback control theory [Lewis and Syrmos 1995]. Shortest Path routing schemes find the shortest path from a given node to the destination node. If the cost, instead of the link length, is associated with each link, these algorithms can also compute minimum cost routes. These algorithms can be centralized (find the shortest path from a given node to all other nodes) or decentralized (find the shortest path from all nodes to a given node). There are certain well-defined algorithms for shortest path routing, including the efficient Dijkstra algorithm [Kumar 2001], which has polynomial complexity.

Bidirectional search is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: Routing schemes based on competitive game theoretic notions have also been developed [Altman et al. 2002]. **Bidirectional search** is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: one forward from the initial state, and one backward from the goal, stopping when the two meet in the middle. The reason for this approach is that in many cases it is faster: for instance, in a simplified model of search problem complexity in which both searches expand a tree with branching factor b, and the distance from start to goal is d, each of the two searches has complexity $O(b^{d/2})$ (in Big O notation), and the sum of these two search times is much less than the $O(b^d)$ complexity that would result from a single search from the beginning to the goal.

As in A* search, bi-directional search can be guided by a heuristic estimate of the remaining distance to the goal (in the forward tree) or from the start (in the backward tree).

Ira Pohl was the first one to design and implement a bi-directional heuristic search algorithm. Andrew Goldberg and other are explaining how the correct termination for the bidirectional Dijkstra's Algorithm has to be.

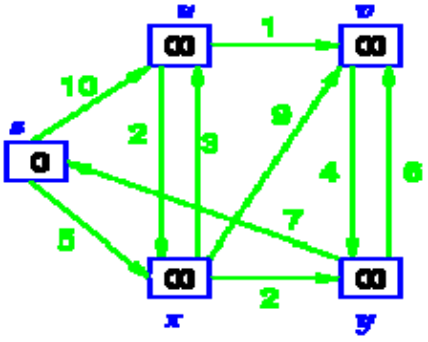


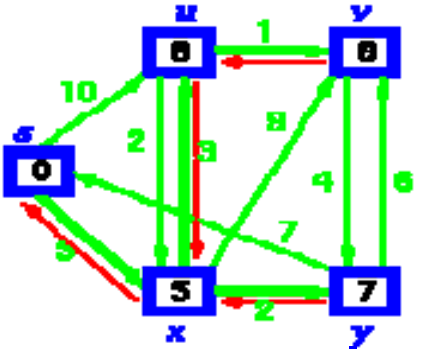
DIJKSTRA'S Algorithm

Dijkstra's algorithm solves the single-source shortest-path problem when all edges have non-negative weights. It is a greedy algorithm and similar to Prim's algorithm. Algorithm starts at the source vertex, s, it grows a tree, T, that ultimately spans all vertices reachable from S. Vertices are added to T in order of distance i.e., first S, then the vertex closest to S, then the next closest, and so on. Following implementation assumes that graph G is represented by adjacency lists.

DIJKSTRA (G, w, s)

1. INITIALIZE SINGLE-SOURCE (G,s)
2. S ← { } // S will ultimately contains vertices of final shortest-path weights from s
3. Initialize priority queue Q i.e., Q ← V[G]
4. while priority queue Q is not empty do
5. u ← EXTRACT_MIN(Q) // Pull out new vertex
6. S ← S□{u}
   // Perform relaxation for each vertex v adjacent to u
7. for each vertex v in Adj[u] do
8. Relax (u, v, w)





CONCLUSION

In this paper, we proposed a Routing schemes in a three-tier framework for intruder information sharing in sensor networks and bypass the data when the intruder is in the network. By finding the different routes in the network. The framework consists of a verifiable intruder reporting (VIR) scheme, a quorum-based caching (QBC) scheme for system-wide propagation of intruder information, and a collaborative Bloom Filter (CBF) scheme for local management of intruder information. For find the routing we are using Bidirectional Heuristic Search and Dijkstra algorithm. Extensive analysis and simulation are also conducted to verify the efficiency of the proposed framework as long as the system parameters are carefully chosen.

ACKNOWLEDEMENT

Thanks to Management of Malineni Lakshmaiah Group of Educational Institutions and to the Princial Dr.J.AppaRao, our Colleagues and Friends of Malineni Lakshmaiah Women's Engineering College.

REFERENCES

[1] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *ACM MobiCom*, pp. 255–265, August 2000.

[2] Bin Tong, Santosh Panchapakesan, and Wensheng Zhang The Department of Computer Science Iowa State University Ames, IA 50011

[3] G. Wang, W. Zhang, G. Cao, and T. La Porta, "On Supporting Distributed Collaboration in Sensor networks," *MILCOM*, pp. 752–757 Vol.2, October 2003.