CLUSTER BASED SECURE DATA TRANSMISSION IN WSN

A.S.Syed Navaz, J.Antony Daniel Rex, S.Jensy Mary

Abstract— In wireless sensor networks, services may fail due to various reasons, including radio interference, de-synchronization, battery exhaustion, or dislocation. Such failures are caused by software and hardware faults, environmental conditions, malicious behavior, or bad timing of a legitimate action. In general, the consequence of such an event is that a node becomes unreachable or violates certain conditions that are essential for providing a service, for example by moving to a different location,

The node can no further provide sensor data about its former location. In some cases, a failure caused by a simple software bug can be propagated to become a massive failure of the sensor network. This results in application trials failing completely and is not acceptable in safety critical applications. The open nature of the wireless communication, the lack of infrastructure, the fast deployment practices, and the hostile deployment environments, make them vulnerable to a wide range of intrusions and security attacks.

The motivation for attacking a sensor networks could be, for example, to gain an undeserved and exclusive access to the collected data. There has been a multitude of attacks described in the literature: probabilistic data packet dropping, topology manipulation, routing table manipulation, prioritized data and control packet forwarding, identity falsification, medium access selfishness etc.

Index Terms— WSN, OSI, ELMO, GUI, SWS, Mobility, Nodes.

1 INTRODUCTION

The protection system of a sensor networks usually relies on the following two mechanisms: Authentication and secure protocols and (ii) intrusion and attack (misbehavior) detection. As the experience from the Internet shows, the weaknesses in authentication and secure protocols are frequently exploited. These protocols alone are in general considered being insufficient to provide the necessary level of protection.

Therefore, there has been a lot of effort invested in providing networks with means for a timely detection of an attack or intrusion. Such detection is often based on methods and algorithms known from the field of machine learning.

Additionally, after sensors get deployed in the monitored area, the access to them can be difficult. For example, a sensor network, with the goal to monitor conditions in the sewer system of a large city, might be inaccessible for maintenance, software updates or battery exchange. Therefore, a special focus has been put on designing energy efficient protocols at all layers of the OSI (Open Systems Interconnection protocol stack.

Additionally to addressing energy constraints, these protocols should impose a high degree of robustness in order to minimize the need for human intervention. The use of these sensor networks in hostile environments means that providing quality of service is essential and requires the implementation of fault-tolerant mechanisms

That can ensure availability and continuity of service. For example, the maximum coverage of the regions monitored by the network and connectivity of the various nodes of the network must be maintained. However in an environment where each node can fail unexpectedly resulting in the isolation of some parts of the network, this guarantee is neither automatic nor easy to achieve.

For all this problems, the integration of mechanisms for monitoring wireless sensor networks, for the reason of topology control, fault tolerance and security are crucial for the effective use of wireless sensor networks.
There are many current management approaches, but each provides only partial solutions to the problems of monitoring and fault tolerance, and they do not adapt to the properties and constraints of many wireless sensor networks.

In summarize there are many papers tried to tackle monitoring methodologies in wireless sensor networks. In this chapter we will try to give an overview on the use of monitoring mechanisms to supervise wireless sensor networks. Then we detailed the description of some research using monitoring mechanisms for reasons of security, topology control or fault tolerance in wireless sensor network, and we will describe our contribution in this field.

- ***A.S.SYED NAVAZ*** *working as an Assistant Professor in the Department of Computer Science at Muthayammal College of Arts & Science, Namakkal, India. E-Mail: a.s.syednawaz@gmail.com*

- ***J.ANTONY DANIEL REX*** *working as an Assistant Professor in the Department of Computer Science at St. Joseph's College of Arts & Science (Autonomous), Cuddalore, India.*

  S. JENSY MARY, *working as an Assistant Professor in the Department of Computer Science at St. Joseph's College of Arts & Science (Autonomous), Cuddalore, India*

2 System Analysis

2.1 Existing System

Local monitoring is a collaborative detection strategy where a node monitors the control traffic going in and out of its neighbors.. Many techniques have been introduced that use the framework of local monitoring to achieve specific tasks such as intrusion detection, building trust and reputation among nodes, protecting against control and data traffic attacks, and building secure routing protocols. Though local monitoring has been demonstrated as a powerful technique for enhancing security of WSNs, it results in a high energy cost since it requires the monitoring nodes to be constantly awake to oversee network activity.

In previous work, we partially addressed the problem of combining local monitoring (to support security) and sleep-wake scheduling (to conserve energy) under the assumption that a malicious node does not have the ability to control its transmission power level. Another limitation is that we did not consider scheduling of the monitoring nodes with the goal of additional energy savings. To the best of our knowledge, in this paper we provide the first methodology for enabling energy-efficient local monitoring in multihop wireless sensor networks.

2.2 Proposed System

We propose the Energy Aware Local Monitoring in Sensor Networks (ELMO) methodology, which consists of a set of mechanisms that significantly reduce the node wake time required for monitoring. The accepted approach for achieving maximum energy savings in sensor networks is to put nodes to sleep and wake them based on schedules that can be timed, event triggered, or some combination of the two.

The contributions of this paper can be summarized as follows:

1. We introduce a technique for conserving energy in WSNs while performing local monitoring without significantly degrading security performance. We believe that this is fundamental for deploying local monitoring in energy conscious networks.

2. We propose a generic on-demand sleep-wake algorithm for network monitoring in scenarios where either no application-specific sleep algorithm exists or the existing sleep-wake algorithm supports an arbitrary communication pattern.

3. We provide an analytical proof that OD-ELMO does not add any vulnerability to a nominal local monitoring technique.

4. We conduct extensive simulation experiments on an existing local monitoring technique with and without OD-ELMO, showing that OD-ELMO achieves a significant reduction in the monitoring cost with negligible degradation in the quality of local monitoring as compared with the baseline SWS.

3 Problem Definiton

A large wireless sensor network consisting of thousands of tiny sensor nodes distributed over a large geographical area can perform the monitor task. Sensor networks have applications in many areas, such as military, environment, agriculture and manufacturing. These applications require great care in the utilization of power which is provided by batteries. In many scenarios, the batteries are neither replaceable nor rechargeable. Hence in the wireless sensor networks the power conservation is paramount.

Local monitoring is a widely used technique to provide security services for WSNs. However, the application of local monitoring interferes with sleep-wake scheduling. In our previous work, we partially addressed the problem of combining local monitoring (to support security) and sleep-wake scheduling (to conserve energy) under the assumption that a malicious node does not have the ability to control its transmission power level. Another limitation is that we did not consider scheduling of the monitoring nodes with the goal of additional energy savings.

The design phase is a multi step process which focuses on system creation with the help of user specifications and information gathered in the above phases. It is the phase where the system requirements are translated to operational details. System has to be designed for various aspects such as input, output etc.

3.1 Module Description

Sensor Network setup or Topology Formation

We contribute to a more systematic understanding and treatment of sensor deployment issues. For this purpose, we studied the existing literature on deployment experience and present a classification of common problems encountered during deployment of sensor networks. A wireless network that is temporarily installed alongside the actual sensor network during the deployment process.

Parameters considered during sensor network formation

- **Transmission range**: nodes communication depends under transmission range which is placed nearly close to each other thus gets better link.
- **Local monitoring**: Nodes must be grouped under specific feature like battery power, processing capability; bandwidth, memory etc. so according to those, nodes are partitioned using driver methods.

- **Mobility**: Mobility refers the node movement procedure so need to consider the mobility options with limitation in maximum and minimum speed.

According to the ELMO process sensor network formed under local monitoring system with sleep wake scheduling.

3.2 Protocol Design

A routing protocol is a protocol that specifies how routers communicate with each other, disseminating information that enables them to select routes between any two nodes on a computer network, the choice of the route being done by routing algorithms. Each router has a prior knowledge only of networks attached to it directly. A routing protocol shares this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network. Design a Routing protocol named as ELMO,

Which is going to implement in OSI layer that need to get and deliver the messages from other layers for that make some more changes in supported layers. The routing protocol is implemented in the layered architecture of the GloMoSim simulator.

- **Sleep wake Scheduling**
- **ELMO**

3.3 ELMO Process

The Energy Aware Local Monitoring in Sensor Networks (ELMO) methodology, which consists of a set of mechanisms that significantly reduce the node wake time required for monitoring. ELMO does not need to do anything since a node and its monitoring neighbors will be automatically awakened by the baseline SWS itself.. ELMO requires each node to have a passive or a low-power wake-up antenna in addition to the usual antenna.

A node that is not involved in network activities such as data forwarding is ordinarily asleep according to the baseline SWS. However, for monitoring purposes, it is awakened on demand by a neighboring node using the wake-up antenna. On-demand sleep-wake protocols use either special purpose. Low-power wake-up antennas. These antennas are responsible for receiving an appropriate beacon from a neighbor node and waking up its node.

We then propose an approach to find and select routes, which accounts for the expected data transfer time over the path and allows reducing the energy consumption of ELMO routing protocols.

The nodes are maintain in three states to establish Sleep wake scheduling which are

- Active state-Node in routing process
- Ideal state-Node which are in ready state
- Sleeping state-Node which is in slow sleeping state by reducing its transmission range

3.4 Initial Configuration Setup

We need to configure some attributes which is supported to execute our routing protocol like Number of nodes, Mobility, Mac protocol, Simulation time, Band width, Transmission range etc… by setting these kinds of attributes we execute out routing protocol with layers interaction. We setup the layer wise results in the configuration process.

The sequence of events at run time:

- The main function in driver.pc is run. This is the C main function, where GloMoSim starts.
- The main function calls parsec main () to start the Parsec simulation engine, initialize the simulation runtime variables and create the driver entity. The parsec main function is used when the user wants to write his own main and is found at PCC DIRECTORY/include/pc api.h .
- Since the function is part of the Parsec runtime system, it is not possible to access the source for it.
- When the simulation ends, parsec main () returns, and the rest of the main function is executed.

In GloMoSim, the driver entity (in ./main/driver.pc) reads the input file descriptor, establishes partitions, allocates memory for node information, calls appropriate functions depending on the read input values such as simulation time and node placement, and finally starts simulation by sending a StartSim message to the partitionEntityName instance of the GLOMOPartition entity type (defined in the glomo.pc file).

The ./main/glomo.pc file performs the following steps:

1. Receive information from driver entity.
2. Find out nodes which belong to current partition and initialize all the layers for these nodes by calling the following entities: GLOMO PropInit (), GLOMO RadioInit(), GLOMO MacInit(), GLOMO NetworkInit(), GLOMO TransportInit(), GLOMO AppInit() and GLOMO MobilityInit().
3. Go into an loop trying to receive messages. When a message is received it retrieves information about the receiving node and calls the appropriate layer. It displays current simulation time during program execution.
4. When the simulation ends, it goes to the finalize code. This code will call the Finalize function for all the la ers of

all the nodes in this partition. The developer can thus collect any needed statistics.

The finalize functions are: GLOMO Radio Finalize(), GLOMO MacFinalize(), GLOMO NetworkFinalize(), GLOMO TransportFinalize(), GLOMO AppFinalize() and GLOMO MobilityFinalize().

3.5 GUI Design or Visualization Tool

A **graphical user interface** (**GUI**) is a type of user interface item that allows people to interact with programs in more ways than typing. Designing the visual composition and temporal behavior of GUI is an important part of simulation results. So design the GUI phase suite to our routing process using JAVA. The primary purpose of the Java VT is to help network designers debug their protocols. It is written in Java to provide portability across multiple platforms.

The GloMoSim simulation can be run with or without the VT. If run without the VT, it can just be executed from the command line. However, if run with the VT, it must be executed through the GUI provided by the VT rather than from the command line.

Finally measure the performance of our routing protocol by some metrics like end to end delay, throughput, collision and energy consumption which are by values and also by graph.

The performance of the proposed algorithm is evaluated via glomosim simulator. Performance metrics are utilized in the simulations for performance comparison:
1. Delivery ratio—the ratio of the number of packets delivered to the destination to the number of packets sent out by a node averaged over all the nodes in the network;
2. Percent wakeup time—the time a node has to be awake specifically to do monitoring, averaged over all the nodes as a percentage of the simulation time;
3. Average end-to-end delay—the time it takes a data packet to reach the final destination averaged over all successfully received data packets;
4. Isolation latency—the time between when a malicious node performs its first malicious action to the time of isolation, averaged over all isolated malicious nodes.

4 System Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. The process of putting the developed system in actual use is called system implementation. Implementation is the final phase. It involves user training, system testing and successful running of developed system.

4.1 SIMULATION PROCEDURE

First, we need to specify the necessary input parameters in the Config.in file as said above. For our simulation procedure, we have been specific about certain parameters as mentioned below to enable hassle free simulation

Terrain range – (800,800)

Number of nodes – 19 (This is a scalable simulator. Hence number of nodes can be increased at will.)

Routing protocol – ODELMO

These parameters were adhered to for the whole process of experimentation with the new protocol. A copy of the config.in file used for the simulation is given below for reference.

config.in:

```
SIMULATION-TIME           : 60S
SEED                      : 5
TERRAIN-DIMENSIONS        :( 800, 800)
NUMBER-OF-NODES           : 19
NODE-PLACEMENT-FILE       : ./nodes.input
MOBILITY                   : MOBILITY TRACE
MOBILITY-TRACE-FILE ./mobility.in
PROPAGATION-LIMIT           :-111.0
PROPAGATION-PATHLOSS      : TWO-RAY
NOISE-FIGURE                : 10.0
TEMPARATURE                 : 290.0
RADIO-TYPE                  : RADIO-NONOISE
RADIO-FREQUENCY           : 2.4e9
RADIO-BANDWIDTH           : 2000000
RADIO-RX-TYPE              : ./ber_bpsk.in
RADIO-RX-SNR-THRESHOLD    : 9.1
RADIO-TX-POWER            : 15.0
RADIO-ANTENNA-GAIN        : 0.0
RADIO-RX-SENSITIVITY      :-91.0
RADIO-RX-THRESHOLD        :-81.0
MAC-PROTOCOL              : 802.11
NETWORK-PROTOCOL          : IP
ROUTING-PROTOCOL          : ODELMO
APP-CONFIG-FILE            : /cbr.in
APPLICATION-STATISTICS   : YES
TCP-STATISTICS            : YES
UDP-STATISTICS             : YES
```

```
ROUTING-STATISTICS              : YES
NETWORK-LAYER-STATISTICS  : YES
MAC-LAYER-STATISTICS            : YES
RADIO-LAYER-STATISTICS         : YES
CHANNEL-LAYER-STATISTICS  : YES
MOBILITY-STATISTICS              : YES
GUI-OPTION                              : YES
GUI-RADIO                              : YES
GUI-ROUTING                          : YES
```

If we mention that the node placement should be according to a placement file, then we need to mention the co ordinates in a separate file called nodes.input.
A sample of how that file would look like is given below

Nodes.in
Format: nodeAddr 0 (x, y, z)
The second parameter is for the consistency with the mobility trace format.

```
0 (650 750)
1 (100 100)
2 (150 50)
3 (200 250)
4 (300 250)
5 (250 250)
6 (230 200)
7 (350 450)
8 (400 450)
9 (450 450)
10 (390 390)
11 (200 100)
12 (550 700)
13 (600 700)
14 (600 200)
15 (550 250)
16 (600 250)
17 (650 250)
```

5 Conclusion

In this paper, we have introduced the protocol ELMO which provides high-performance energy-efficient local monitoring for wireless sensor networks. ELMO has three manifestations, which correspond to the three classes of sleepwake schemes identified in the paper. For the first class (synchronized sleep-wake WSNs), we determined that localmonitoring needs no modification. For the second class (continuously acting WSNs),

we found that local monitoring can call the baseline sleep-wake scheme (SWS) with modified parameter values. We found that the third class (triggered WSNs) presents the most demanding case in terms of the needed adaptation of local monitoring. Here, hardware support such as low-power or passive wake-up antennas is required. We proposed a scheme called ODELMO whereby a node awakens the guards responsible for monitoring the next hop from that node. We proved analytically that OD-ELMO does not weaken the security of local monitoring. Additionally,

simulation experiments over a wide range of conditions show that the performance of local monitoring with ELMO is comparable to that without ELMO, while realizing listening energy savings on the order of 20-100 times, depending on the network parameters. We are currently researching extensions that will facilitate providing security guarantees in mobile ad hoc networks and building a trust framework for such networks.

.

5.1 Future Work

Our future work would be further investigating the applicability of the proposed technique and routing algorithm to more general wireless networks. The main challenge of our future work is to find an efficient way of distributing keys and keying materials to sensor nodes prior to deployment with better QOS.

We found that local monitoring can call the baseline sleep-wake scheme (SWS) with modified parameter values. We found that the third class (triggered WSNs) presents the most demanding case in terms of the needed adaptation of local monitoring. Here, hardware support such as low-power or passive wake-up antennas is required. We proposed a scheme called ODELMO whereby a node awakens the guards responsible for monitoring the next hop from that node. We proved analytically that OD-ELMO does not weaken the security of local monitoring. Additionally,

It is well-known that Energy Efficient Trusted Authority (EE-TA) architecture enables better resource allocation and helps to improve power control and QOS. It also scales well to different network sizes and node densities under energy constraints.

References

[1] M. Tubaishat and S. Madria, “Sensor Networks: An Overview,” IEEE Potentials, vol. 22, no. 2 pp. 20-23, Apr./May 2003.

[2] J. Yick, B. Mukherjee, and D. Ghosal, “Analysis of a Prediction- Based Adaptive Mobility Tracking Algorithm,” Proc. Second Int’l Conf. Broadband Networks, pp. 809-816, 2005.

[3] M.Y.S. Uddin and M.M. Akbar, “Addressing Techniques in Wireless Sensor Networks: A Short Survey,” Proc. Fourth Int’l Conf. Electrical and Computer Eng., 2006.

[4] Y. Wang, G. Attebury, and B. Ramamurthy, “A Survey of Security

Issues in Wireless Sensor Networks,” IEEE Comm. Surveys and Tutorials, vol. 8, no. 2, pp. 2-23, 2006.

[5] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, “Particle Filtering for Multi-Target Tracking and Sensor Management,” Proc. Fifth Int’l Conf. Information Fusion, 2002.

[6] Optimal Filtering, B.D.O. Anderson and J.B. Moore, eds. Prentice-Hall, 1979.

[7] V. Hasu and H. Koivo, “Decentralized Kalman Filter in Wireless Sensor Networks—Case Studies,” Proc. Int’l Joint Conf. Computer, Information, and Systems Sciences, and Eng., 2005.

[8] S.J. Julier and J.K. Unlmann, “A New Extension of the Kalman Filter to Nonlinear Systems,” Proc. 11th Int’l Symp. Aerospace/ Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II (AeroSense), 1997.

[9] J.H. Kotecha and P.M. Djuric, “Gaussian Particle Filtering,” IEEE Trans. Signal Processing, vol. 51, no. 10, pp. 2592-2601, Oct. 2003.

[10] Sequential Monte Carlo Methods in Practice, A. Doucet, N. de Freitas, and N. Gordon, eds. Springer, 2001

[11] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing, Imielinski and Korth, Eds. Boston, MA: Kluwer Academic, 1996, vol. 353 [Online]. Available: citeseer.nj. nec.com/johnson96dynamic.html

[12] V. Park and M. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in Proc. 16th Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM), 1997, pp. 1405–1413 [Online]. Available: citeseer.ist.psu.edu/article/ park97highly.html

[13] M. Joa-Ng and I.-T. Lu, “Apeer-to-peer zone-based two-level link state routing for mobile ad hoc networks,” IEEE J. Sel. Areas Commun., vol. 17, no. 8, pp. 1415–1425, Aug. 1999.

[14] N. Nikaein, C. Bonnet, and N. Nikaein, “HARP—Hybrid Ad Hoc Routing protocol,” in Int. Symp. Telecommunications (IST 2001), Tehran, Iran, Sep. 2001.

[15] V. Ramasubramanian, Z. Haas, and E. Sirer, “SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks,” in Proc. ACM MobiHoc, 2003.

BIOGRAPHY

A.S.SYED NAVAZ received M.Sc in Information Technology from K.S.Rangasamy College of Technology, Anna University Coimbatore, M.Phil in Computer Science from Prist University, Thanjavur, M.C.A from Periyar University, Salem and Pursuing Ph.D in the area of Wireless Sensor Networks. He researched and published in International journals and working as Editorial Board Member & Reviewer for International journals also Member in International Social Bodies. Currently he is working as an Assistant Professor in the Department of Computer Science at Muthayammal College of Arts & Science, Namakkal, India. His Research areas are Wireless Sensor Networks, Mobile Computing & Image Processing.

J.ANTONY DANIEL REX received M.C.A from Periyar University, Salem, M.Phil., in Computer Science from Government arts college ,Salem, He researched and published one International journals & presented one paper in national conference and one paper in international conference . Currently he is working as an Assistant Professor in the Department of Computer Science at St.Joseph's College of Arts & Science (autonomous),Cuddalore, India. His Research areas are Wireless Sensor Networks, Mobile-Adhoc network& Image processing.

S. JENSY MARY received M.Sc., from Holy Cross College, Trichy from Bharathidasan University, M.Phil in Computer Science from St.Joseph's College of Arts & Science (autonomous), Cuddalore; she researched and published one paper in national conference. Currently she is working as an Assistant Professor in the Department of Computer Science at St.Joseph's College of Arts & Science (autonomous), Cuddalore, India. Her Research areas are Multimedia Streaming & Image processing.