Adaptive Trust Management Protocol Based on Fault Detection for Wireless Sensor Networks

Laura Gheorghe, Răzvan Rughiniş, Răzvan Deaconescu, Nicolae Ţăpuş
Politehnica University of Bucharest
Bucharest, Romania
{laura.gheorghe, razvan.rughinis, razvan.deaconescu, ntapus}@cs.pub.ro

***Abstract*—Trust management is an important issue in self-configurable and autonomous networks such as Wireless Sensor Networks. Sensor nodes need to determine if other nodes are trustworthy, in order to decide whether to cooperate with them in completing the sensing and communication tasks. Therefore, adaptive trust management assures an appropriate level of security to the critical services provided by Wireless Sensor Networks. In this paper, we present the Adaptive Trust Management Protocol for Wireless Sensor Networks, which is able to compute data trust based on fault detection. The adaptive trust management protocol operates cycles in which reputation values are computed and penalty values are exchanged periodically. A spanning tree is generated for the sensor network, after which nodes evaluate their children using the fault detection mechanism and then exchange penalty values with their neighbors. The protocol has been implemented in TinyOS and evaluated in a test scenario using TOSSIM.**

Keywords-reputation, trust, fault detection, wireless sensor networks

I. INTRODUCTION

Wireless Sensor Networks are an emerging technology, becoming a fundamental method in monitoring various environments [1]. A sensor network consists of a large number of sensor nodes that are able to perform sensing, processing and communicating tasks in a collaborative manner, in order to detect specific events that take place in the monitored environment [2].

The sensor network can be seen as a service provider for user applications. The services provided by the WSN are data collection and data delivery. A service-oriented approach fills the semantic gap between high level application requirements and the low level operations provided by the sensor network [3] [4].

Because sensor networks are used in critical applications such as battlefield surveillance, homeland security and medical monitoring, a critical task when designing a Wireless Sensor Network is to ensure security against malicious attacks and faulty nodes [5].

A sensor network can be protected against external malicious nodes through the use of authentication methods [6]. However, for internal faulty nodes, another method must be used to ensure protection against false reports. A solution against false reports relies on using a trust management scheme and enforcing a trust policy that sensor nodes must respect [7].

While the aspects presented in this paper are general and can be applied to any kind of wireless network, the special aspect specific to Wireless Sensor Network considered is the need to minimize the consumption of energy. Therefore, we aim to reduce the number of packets being sent and received and we use simple algorithms to compute the trust values.

We propose an Adaptive Trust Management Protocol (ATMP) that determines trust by computing reputation values based on fault detection techniques. The protocol operates in three phases: setup, learning and exchange phase. In the setup phase, a spanning tree is built, while in the learning phase, the local penalty value is modified on the basis of the fault detection techniques. In the exchange phase, nodes exchange reputation values, re-compute them and determine trust. The protocol is adaptive because the reputation values are modified on each cycle, according to the detected faults and to the penalty values received from neighboring nodes. The protocol is collaborative because sensor nodes interact on every cycle in order to update reputation values.

The rest of this paper is organized as follows: Section II presents the problem of false reports and the proposed solution. Section III contains definitions of "trust" and "reputation" and introduces "trust management". Section IV describes related work. Section V introduces the Adaptive Trust Management Protocol. Section VI presents implementation details of the protocol in TinyOS. Section VII describes the test scenario and simulation results. Section VIII discusses advantages and potential problems. Section IX concludes this paper.

II. FALSE REPORTS

The main application of Wireless Sensor Networks consists in environment monitoring and event detection. However, malicious or faulty nodes can generate and send incorrect data to the base station. Incorrect data can disrupt normal data fusion, and event detection. It can trigger false alerts by generating false alert data, or it can hide important events by generating false normal data instead of alert data [8].

Attackers could physically capture and compromise a sensor node. They could maliciously inject invalid data into the network in order to disrupt normal functionality, especially event detection. Authentication and cryptographic

 ISBN: 978-1-61208-105-2

methods are not a solution to this problem, because once they have captured a sensor node, the attackers can have access to the cryptographic information stored on the sensor and they can successfully authenticate themselves [7].

Besides malicious attacks, hardware and software faults can also cause incorrect data to be generated and sent to the base station. For example, the sensing unit or the radio can malfunction, altering packets and generating inaccurate sensor readings. This problem, also, cannot be solved by using authentication and cryptographic mechanisms.

A possible solution relies on the fact that, when an event takes place in a specific area of the sensor network, sensor nodes that are in the proximity should have similar collected data [9]. However, if a node is malicious or faulty, it can generate data sets that do not match its neighbors' data. Therefore, an incorrect data reading can be detected by comparing the data collected by sensor nodes from the same area.

In the data aggregation process, the values received from the children nodes are combined and one single value is forwarded toward the base station [7]. In order to prevent the transmission of faulty information, before the aggregation operation takes place, the node waits for all the children to send data, and after that, it checks whether proximal nodes send similar data. The incorrect values will be detected and they will not be forwarded, while the source nodes will be penalized.

III. TRUST MANAGEMENT

Trust can be defined as the level of confidence a decident has in the performance of a person or object. Trust has always played an important role in social environments, and recently it started to be considered in various kinds of networks, such as peer-to-peer, ad-hoc and sensor networks.

Trust is associated with the ability to provide the expected service. In sensor networks, trust is associated with the accuracy of event detection and undisturbed network and protocol functionality.

From a networking perspective, a node can evaluate and use trust in order to decide whether another node is uncooperative, malicious or faulty.

Trust is especially critical in networks that rely on collaborative event detection and environmental monitoring, where nodes cooperate permanently in order to provide accurate data collection that characterizes the monitored environment.

Trust management was first defined by Blaze et al. in [9]. They propose a framework for security policy, credentials and trust relationships.

In Wireless Sensor Networks, trust management is a challenging task because they are autonomous and self-configurable, without any central point of management. In such networks, trust management is a cooperative process, rather than a local node oriented process.

Three types of trust evaluation have been defined in Wireless Sensor Networks: communication trust, data trust and energy trust [10]. Communication trust consists in computing reputation values based on successful and failed transactions. Data trust is the assurance of fault tolerance and data consistency. Energy trust depends on the level of existing energy and on the threshold level needed in order to perform sensing and communication tasks.

In this paper, we focus on data trust, and we present a data management protocol that is able to enforce trust on the basis of fault detection methods, and to provide data consistency for Wireless Sensor Networks.

Adaptive Trust Management Protocol (ATMP) uses cooperative trust management and has a hierarchical view over the network. The parent nodes obtain information about their children and then exchange penalty values with their neighbors in order to compute reputation values.

IV. RELATED WORK

The most important trust management schemes suited for Wireless Sensor Networks are: Reputation-based Framework for Sensor Networks (RFSN), Agent-based Trust and Reputation Management (ATRM), Parameterized and Localized trUst management Scheme (PLUS), Group-based Trust Management Scheme (GTMS) and Trust-aware Query Processing.

A. RFSN

Reputation-based Framework for Sensor Networks (RFSN) is a trust management framework in which each node maintains a trust value for each neighbor node [12]. RFSN uses statistical and decision theory methods in order to predict the future behavior of nodes and to identify misbehaving nodes.

Trust values are computed based on reputation. Bayesian formulation is used to represent the reputation of a node, but also for updates, integration and trust evolution. A level of confidence is computed for each data reading, through consensus-based outlier detection schemes.

RFSN is not suited for sensor networks with high mobility, because in this case, the reputation values will not converge. A node must have constant neighbor interactions in order for the reputation to stabilize.

B. ATRM

Agent-based Trust and Reputation Management (ATRM) is based on mobile agents that are generated by a single trusted authority [13]. It assumes that the information carried by the agents will not be accessed or modified by the malicious nodes present in the sensor network.

The major advantage of this trust management scheme is that it takes into consideration the power and bandwidth constrains and tries to reduce communication overhead and delay.

C. PLUS

Parameterized and Localized trUst management Scheme (PLUS) is built on top of the PLUS_R routing scheme. It uses a localized distributed algorithm in which trust is computed using direct and indirect observations [14].

In PLUS, the control messages generated by the BS contains a hashed sequence number (HSN). When a judge node receives a packet from another node, it uses the HSN to

 ISBN: 978-1-61208-105-2

check the integrity of the received packet. If the integrity has been compromised, the trust in node i is decreased. However, if node i had just forwarded the packet and it is not a malicious node, it is penalized without being guilty.

D. T-RGR

Trust management scheme for Resilient Geographic Routing (T-RGR) is a non-adaptive scheme in which sensor nodes observe the behavior of their one-hop neighbors [15]. T-RGR is vulnerable to collaborative attacks because it uses direct observations in order to compute trust values.

E. Group-based Trust Management Scheme (GTMS)

Group-based Trust Management Scheme (GTMS) is a method for clustered Wireless Sensor Networks that evaluates the trust of a group of sensor nodes [16]. This approach reduces the memory used to store trust values for each observed entity.

F. Trust-aware Query Processing

Trust-aware Query Processing is a new approach to efficient trust-aware routing in data intensive WSNs [17]. The trust metric is based on subjective logic that includes properties of deployment area, sensor design, and properties of the transmission channels. The approach optimizes energy consumption and provides reliability to data intensive sensor networks.

V. ADAPTIVE TRUST MANAGEMENT PROTOCOL

Wireless Sensor Networks are used to collect data about the environment in which they are placed. This data may refer to temperature, humidity, pressure, light, sound, and advanced properties such as air or water quality, or other specific object attributes.

We assume that nodes in the same range will gather similar measurement data regarding a given environmental property. The optimal range will be determined experimentally because it depends on the deployed network and application.

We also assume that nodes have the capability to determine the distance between them by using ranging techniques such as TOA-based or RSS-based ranging. This topic goes beyond the purpose of this paper. We assume that data packets contain the localization of the source packet.

We define the reputation of a node as a measure of confidence in ability of that node to correctly collect and transmit sensor readings.

Every node computes the penalty values for neighbor nodes on the basis of the packets it has to forward to the base station. After that, the nodes exchange penalty values, and the final reputation for a specific node is computed using the local penalty values and the received ones.

Each sensor node uses reputation values to determine whether it can trust a certain node or not. The trust is represented as a binary value. The trust values are used in order to select which messages will not be forwarded or aggregated.

The Adaptive Trust Management Protocol for Wireless Sensor Networks consists in the following three phases:

A. The setup phase

1) The base station broadcasts a Hello packet.

2) The nodes that receive a Hello packet re-broadcast it in order to reach the whole network.

3) Every node stores the address of the node from which it has received the first Hello packet. This node will be called the parent. This way, a spanning tree overlay will be contructed.

B. The learning phase

1) The learned trust for each neighbor is set to the default value.

2) The nodes start collecting data and sending it to the base station. Every node will forward data towards the base station by using the parent node.

3) The nodes perform error detection using the following algorithm:

a) Leaf nodes just transmit the raw collected data

b) Every other node within the spanning tree waits to receive data from children nodes for a specific period of time. The packets are stored in a list.

c) After the waiting period, based on the location of each source of data, the nodes are grouped in clusters, so that the distance between nodes within a cluster is less that a constant ε.

d) Each cluster of nodes is represented by a list of measurement values generated by member nodes. Each list of nodes is sorted in an ascending manner.

e) For each list of values, the median value is computed. For error detection purposes, the median is a better measure of the central tendency than the average, because it is less sensitive to outliers.

f) For each list, the values are compared with the median value. If the difference between the considered value and the median is greater than a constant deviation γ, the value will be considered erroneous. The constant deviation γ is defined as a percent of the median value. The actual value depends on the application.

4) For each node that is the source of an erroneous data value, the local penalty value will be increased with a specific value. This values depends directly on the diference between the analized and the median values.

5) Each non-leaf node will have a list of associations between nodes and penalty values, called penalty associations.

C. The exchange phase

1) Each node sends the list of penalty associations using a broadcast message.

 ISBN: 978-1-61208-105-2

2) Each node waits to receive the lists of associations from their neighbors for a predefined period of time.

3) After the period of time has expired, the reputation value is recomputed using the current local penalty obtained through the learning phase, the previous reputation value and the penalty values received from the neighbours using the following formula:
Reputation(X)=Previous_reputation(X)-Local_penalty(X)-$\Sigma_Y(W_Y$*Received_penalty$_Y(X))$. The received penalty from node Y is weighted with W_Y, which represents the trust value that the current node has in node Y. The trust value is either 1 for trusted nodes or 0 for un-trusted nodes.

4) The trust value is recomputed using the following conditions:

Trust(X)=1 if Reputation(X)>=TRUST_LIMIT

Trust(X)=0 if Reputation(X)<TRUST_LIMIT

This computed trust value can be used by parent nodes in order to forward or aggregate data packets received from children that are trustworthy, and ignore packets from children in which they do not trust.

A complete trust management cycle consists in a learning phase and an exchange phase. At the end of a cycle, each node has updated their trust in other nodes, even if they are not reachable within one single hop.

The setup phase is repeated after a specific number of trust management cycles. This phase must be repeated because the topology may change and nodes could lose their parents, and therefore they would not be able to send data to the base station. The number of cycles after which a setup phase must take place depends on the duration of a cycle and on the dynamic of the network. The dynamic of the network depends on the frequency of the topology changes that may be caused by energy depletion and node mobility.

After a topology change, two nodes that were not neighbors in the previous cycle can become parent and child after a setup phase. The parent is now able to use the information it has previously obtained about the new child node.

VI. PROTOCOL IMPLEMENTATION

The protocol has been implemented in TinyOS, an open-source, component-oriented operating system designed especially for Wireless Sensor Networks [18].

A single component was used to implement the protocol and a wiring component is used to place the protocol component on top of the TinyOS Active Message Stack.

The messages used in our implementation of the protocol rely on the following Layer 2 header which corresponds to the TinyOS Active Message header. The real AM header contains other additional fields that are not relevant for the understanding of the Adaptive Trust Management Protocol. The Layer 2 header is represented in Figure 1. The source and destination addressed are AM addresses used for the hop-by-hop communication between nodes.

Hop_src	Hop_dest	Upper Layer data

Figure 1. Layer 2 header

In Figure 2, we present the Layer 3 header, which is specific to our protocol. The source and destination addresses are AM addresses that are used for the end-to-end communication between nodes. The Type field represents the type of message being sent: hello, data or penalty exchange message. The fields X and Y represent the coordinates of the source node used to compute the distance between the nodes in order to form clusters.

End_src	End_dest	Type	X	Y	Upper Layer Data

Figure 2. Layer 3 header

The component contains nine events implemented, from which the most important are the receive event that is used to manage received messages and the fired events for each of the four timers that are used to perform specific actions.

The component uses four timers in order to assure the proper functionality of the protocol: Hello timer, Collect timer, TrustAnalyse timer and TrustExchange timer.

The Hello timer is used only by the base station in order to periodically broadcast Hello messages that are used to build the spanning tree overlay, which corresponds to the Setup phase, step 1.

The *Hello_timer.fired* event is used to periodically send messages containing: the Hop_src and End_src equal to the base station identifier, the Hop_dest and End_dest equal to AM_BROADCAST_ADDR, the broadcast address, the type equal to 1 which represents Hello messages. Fields X and Y are not filled. The Application data contains a sequence number in order to keep track of the Setup phases.

The Collect timer is used by the nodes in the network to periodically collect data from the environment and send it towards the base station, which is the implementation of Learning phase, step 2.

The Collect_timer.fired event sends messages with the following fields: Hop_src and End_src equal to the node identifier that is generating the message, End_dest equal to the identifier of the base station, Hop_dest equal to the parent node identified in the Setup phase, Type equal to 2 which represents Data messages, and fields X and Y containing the coordinates of the source node. The data packet is sent to the parent of the source node.

The TrustAnalyse timer is used in Learning phase, step 3, to model the waiting period in which non-leaf nodes receive data packets from their children and forward them towards the base station.

The TrustAnalyse_timer.fired event implements the algorithm presented in Learning phase, step 3, in which clusters are formed, messages are sorted in lists for each cluster and data errors are detected using the median method. The local penalty values are modified according to the data errors detected and broadcasted to the neighbor messages. The packets used to broadcast the penalty associations contain the following fields: Hop_src and End_src is equal to the node identifier, Hop_dest and End_dest equal to AM_BROADCAST_ADDR, type is set to 3, representing a penalty exchange packet. Fields X and Y will not be filled. The payload contains only the penalty associations modified in the Learning phase.

 ISBN: 978-1-61208-105-2

The TrustExchange timer is used in the Exchange phase, step3, to represent the waiting period in which nodes receive penalty associations from their neighbors.

The TrustExchange_timer.fired event is used to re-compute reputation lists according to the local penalty values and the penalty associations received from the neighbor nodes. The trust binary values are determined by comparing the reputation values obtained with the threshold limit of the accepted reputation.

The Receive.receive event is used to react to every message received by the current node:

1. If the node receiving the message is the base station and the message has type equal to 2, the message contains collected data that reached destination.

2. If the message type is 1 and the node receiving the message has no parent, the Hop_src node becomes its parent. The message is re-broadcasted in order to reach other nodes from the network.

3. If the message type is 2 but the current node does not have a parent yet, the following message is generated: "No route to base station, packet from X with value Y is dropped".

4. If the message type is 2 and the current node has a parent, the Hop-by-hop addresses are changed to reflect the current Hop source (Hop_src) and destination (Hop_dest) and the message is forwarded towards the base station, through the Hop_dest. The message is stored until analyzed in the TrustAnalyse_timer.fired event.

5. If the message type is 3, the received penalty associations are stored locally.

VII. TEST SCENARIO

The Adaptive Trust Management Protocol has been tested using TOSSIM, a simulator for TinyOS applications [19], which is particularly adequate for testing WSN protocols [20], [21].

We use a test scenario based on a simple topology in order to prove the functionality of the protocol. The topology is represented in Figure 3, and it contains ten nodes placed at the coordinates specified in the figure. The line between two nodes indicates that they are in the broadcast range of each other, and therefore they can directly communicate with each other. The three circles observed in the figure represent the clusters identified by the nodes using a specialized algorithm.

We present the output of TOSSIM for every step in the protocol. We choose to display only receive events in order to eliminate redundant data from the figure. Even for broadcast messages, the Hop_dest field in the received packet is the unicast AM address of the receiving node. This behavior is specific to TinyOS implementation. The broadcast address is equal to 65535, as the AM address is represented on 16 bits.

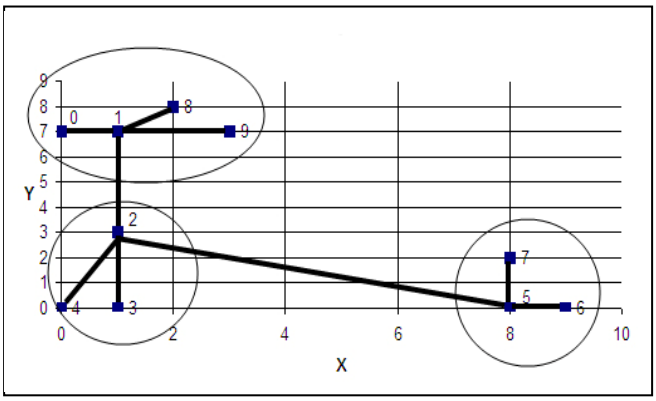

Figure 3. Simple topology

The Setup phase consists in flooding the network with Hello messages and building the spanning tree overlay. As it can be seen in Figure 4, each node learns its parent node when it first receives a Hello packet. For example, node 1 receives Hello messages from node 0, 8, 9 and 2, but it stores as parent, the node from which it has received the first Hello message.

After all nodes have learned their parent, the spanning tree is ready and the sensor nodes can start collecting data and send it towards the base station.

```
(1): Packet received (Hop_src=0 Hop_dest=1 End_src=0 End_dest=65535 type=1)
(1): Parent 0
(9): Packet received (Hop_src=1 Hop_dest=9 End_src=0 End_dest=65535 type=1)
(9): Parent 1
[…]
(3): Packet received (Hop_src=2 Hop_dest=3 End_src=0 End_dest=65535 type=1)
(3): Parent 2
(1): Packet received (Hop_src=2 Hop_dest=1 End_src=0 End_dest=65535 type=1)
(7): Packet received (Hop_src=5 Hop_dest=7 End_src=0 End_dest=65535 type=1)
(7): Parent 5
(6): Packet received (Hop_src=5 Hop_dest=6 End_src=0 End_dest=65535 type=1)
(6): Parent 5
```

Figure 4. Setup phase

The nodes receiving a data packet will forward it using the parent node. A four-hop data routing process is presented in Figure 5.

```
(5): Packet received (Hop_src=6 Hop_dest=5 End_src=6 End_dest=0 type=2
msg=14)
(2): Packet received (Hop_src=5 Hop_dest=2 End_src=6 End_dest=0 type=2
msg=14)
(1): Packet received (Hop_src=2 Hop_dest=1 End_src=6 End_dest=0 type=2
msg=14)
(0): Packet received (Hop_src=1 Hop_dest=0 End_src=6 End_dest=0 type=2
msg=14)
(0): Received from End_src=6 collected data=14
```

Figure 5. Collect and route data packets

The data packet is generated by node 6, which has the parent node 5. The Layer 2 AM addresses are changes at every hop as it can be observed in Figure 5. Each node on the path forwards the packet to its parent. Node 0 displays a message with the data received and the source node.

 ISBN: 978-1-61208-105-2

```
(0): Received from End_src=2 collected data=8
(0): Received from End_src=3 collected data=9
(0): Received from End_src=4 collected data=12
(0): Received from End_src=7 collected data=15
(0): Received from End_src=1 collected data=4
(0): Received from End_src=2 collected data=9
(0): Received from End_src=3 collected data=10
(0): Received from End_src=5 collected data=14
(0): Received from End_src=6 collected data=15
(0): Received from End_src=9 collected data=6
```

Figure 6. Collected data

Figure 6 presents all data received by the base station in a round. Node 2 has received and stored the messages from nodes 3, 4, 5, 6, 7. Two clusters are identified, cluster1 containing nodes 2, 3, 4 and cluster 2 containing the nodes 5, 6, 7, as they are represented in Figure 3. Two sorted lists are built containing the measurement data from nodes within the two clusters: [8, 9, 12] and [14, 15, 15]. The median values for the two lists are 9 and 15. The value γ was set to 20%, therefore the value 12 collected and sent by node 4 is found to be erroneous.

Initially, the local reputation is set to a default value, for example 100, and local_penalty[4] is set to 0. After the error is detected, local_penalty[4]=3, the difference between the value sent by node 4 and the median value. Node 2 sends a broadcast message announcing that it has detected an error. The message contains the accused node identifier and the error found, as it can be seen in Figure 7. The broadcast message is received by nodes 5, 4, 1 and 3, and they compute the final reputation value based on the local penalty and the received penalty association. The value obtained by all receiving nodes is 97.

```
(2): Trust Packet sent (src=2 dest=65535 End_src=2 End_dest=65535 type=3
node=4 penalty=3)
(2): node=4 reputation=97
(5): Received from 2 dif_reputation=3 in node=4
(4): Received from 2 dif_reputation=3 in node=4
(1): Received from 2 dif_reputation=3 in node=4
(3): Received from 2 dif_reputation=3 in node=4
(3): node=4 reputation=97
(4): node=4 reputation=97
(5): node=4 reputation=97
```

Figure 7. Exchanging penalty associations

Unless the data packet from node 4 is dropped by node 2, the process is repeated by node 1, which also detects that the value sent by node 4 is erroneous and announces nodes 0, 8 and 9.

VIII. DISCUSSION

The TRUST_LIMIT value used to compute trust depends on the application and the behavior of the sensor nodes. If the reputation of a node continues to drop under a certain limit, the node should not be trusted anymore and the packets received from it should not be forwarded or used in the aggregation process. Therefore, the procedure for computing the reputation and trust has the advantage of eliminating both nodes that perform one serious error and nodes that generate many relatively small errors.

A problem arises if one of the non-leaf nodes that forwards data towards the base station starts modifying the data contained in the packets. This behavior could be determined by a failure in the node or because it is malicious. We found the solution to integrate a Message Authentication Code (MAC) into the message that would be computed with a secret key shared only between the source node and the base station. This way, if the data packet is modified on the way, the malicious node does not have the secret key of the source node, therefore it will not re-compute correctly the MAC.

Another problem could appear regarding the formula for computing the reputation, in which the reputation of a node can only descrease or stay constant, but can not increase or return to baseline. In some cases a redemption procedure is needed. The formula can be adjusted as follows: $Reputation(X)=Previous\_reputation(X)+Local\_penalty(X)+\Sigma_Y(W_Y*Received\_penalty_Y(X))$, where the penalty is negative and proportional to the detected error in the case of fault detection, and the penalty is positive and equal to a value determined experimentally if measured values are detected as normal.

The major advantage of this protocol is that it can detect data packets generated by faulty or malicious nodes and drop them before reaching the base station. Therefore, a number of useless send and receive operations are avoided and energy consumption is minimized.

The filtering of erroneous data is also very useful for the data aggregation process. Once aggregated, the base station would not be able to detect errors in the received data. Therefore, data values must be verified before being aggregated.

IX. CONCLUSIONS

Wireless Sensor Networks are deployed in order to provide a service to the end user. Medical and military monitoring consists in critical services provided that must be protected using an efficient security solution.

We developed the Adaptive Trust Management Protocol for Wireless Sensor Network, a protocol that computes reputation and trust based on fault detection in three phases organized in cycles.

One cycle contains a Setup phase and a number of Learning and Exchange phases. In the Setup phase, the base station broadcasts Hello messages that reach every node in the network. A spanning tree overlay is build by learning the parent of each node from the first Hello message received in a cycle.

In the Learning phase, the nodes group the messages received in a predefined period of time by location and determine the erroneous data based on the assumption that two nodes that are close to each other should have similar sensor readings. Based on the errors detected, the local penalty values are modified.

In the Exchange phase the local penalty values are exchanged with their neighbors and the reputation and trust values are recomputed using the local penalty values and the received penalty associations.

The trust values can be used to filter erroneous data packets before reaching the base station, in order to minimize

 ISBN: 978-1-61208-105-2

the energy consumption, and to obtain correct data during the aggregation process.

The protocol has been implemented in TinyOS and its functionality has been evaluated in a test scenario using TOSSIM.

BIBLIOGRAPHY

[1] L. Gomez and C. Ulmer, "Secure Sensor Networks for Critical Infrastructure Protection", 2010 Fourth International Conference on Sensor Technologies and Applications, 2010, pp. 144-150.

[2] K. Kabri and D. Seret, "An evaluation of the cost and energy consumption of security protocols in WSNs", 2009 Third International Conference on Sensor Technologies and Applications, 2009, pp. 49-54.

[3] F. C. Delicato, P. F. Pires, L. Pirmez, and T.V. Batista, "Wireless Sensor Networks as a Service", 2010 17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2010, pp. 410-417.

[4] E. Avilés-López and J.A.García-Macías, "TinySOA: a service-oriented architecture for wireless sensor networks", Service Oriented Computing and Applications, Vol. 3, No. 2, 2009, pp. 99-108.

[5] D.I. Curiac, M. Plastoi, O. Banias, C. Volosencu, R. Tudoroiu and D. Pescaru, "Software Development for Malicious Nodes Discovery in Wireless Sensor Network Security", 2010 Fourth International Conference on Sensor Technologies and Applications, 2010, pp. 402-407.

[6] L. Gheorghe, R. Rughiniş, R. Deaconescu, and N. Ţăpuş, "Authentication and Anti-replay Security Protocol for Wireless Sensor Networks", The Fifth International Conference on Systems and Networks Communications, August 22-27, 2010, pp. 7-13.

[7] J. Zheng and A. Jamalipour, "Wireless Sensor Networks A Networking Perspective", John Wiley & Sons, 2009.

[8] F. Ye, H. Luo, S. Lu, and L Zhang, "Statistical en-route filtering of injected false data in sensor networks", 23th Annual IEEE Joint Conference of the IEEE Computer and Communication Societies (INFOCOM'04), vol. 23, no. 4, March 2004, pp. 839-850.

[9] G.R. Abuaitah, "Trusted Querying over Wireless Sensor Networks and Network Security Vizualization", Master of Science Thesis, 2006.

[10] M. Blaze, J. Feigenbaum, and J. Lacy. "Decentralized Trust Management". In IEEE Symposium on Security and Privacy, 1996.

[11] Dong Hui-hui, Guo Ya-jun, Yu Zhong-qiang, Chen Hao, "A Wireless Sensor Networks Based on Multi-angle Trust of Node," ifita, vol. 1, pp.28-31, 2009 International Forum on Information Technology and Applications, 2009.

[12] S. Ganeriwal and M.B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '04), pp. 66-67, Oct. 2004.

[13] A. Boukerche, X. Li, and K. EL-Khatib, "Trust-Based Security for Wireless Ad Hoc and Sensor Networks," Computer Comm., vol. 30, pp. 2413-2427, Sept. 2007.

[14] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and Localized Trust Management Scheme for Sensor Networks Security," Proc. Third IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '06), pp. 437-446, Oct. 2006.

[15] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," J. Parallel and Distributed Computing, vol. 67, no. 2, pp. 215-228, 2007.

[16] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y. Song, "Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, pp. 1698 - 1712, 2009.

[17] V. Oleshchuk and V.Zadorozhny, "Trust-Aware Query Processing in Data Intensive Sensor Networks", 2007 International Conference on Sensor Technologies and Applications, 2007, pp. 176-180.

[18] P Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler, "TinyOS: An Operating System for Sensor Networks", Ambient Intelligence, 2005, pp. 115-148.

[19] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", In SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, 2003, pp. 126-137.

[20] L. Gheorghe, R. Rughiniş, and N. Ţăpuş, "Fault-Tolerant Flooding Time Synchronization Protocol for Wireless Sensor Networks", The Sixth International Conference on Networking and Services, ICNS 2010, March 7-13 – Cancun, Mexico, 2010, pp. 143-149.

[21] R. Rughiniş and L. Gheorghe, "Storm Control Mechanism for Wireless Sensor Networks", 9th RoEduNet IEEE International Conference, June 24-26, 2010, pp. 430-435.

 ISBN: 978-1-61208-105-2