Sensor Wars: Detecting and Defending Against Spam Attacks in Wireless Sensor Networks

Serdar Sancak[1], Erdal Cayirci[2], Vedat Coskun[3]
Naval Science and Engineering Institute
Turkish Naval Academy
Istanbul, Turkey
[1]ssancak@dzkk.tsk.mil.tr, [2]erdal@ece.gatech.edu,
[3]vedatcoskun@dho.edu.tr

Albert Levi
Faculty of Engineering and Natural Sciences
Sabanci University
Istanbul, Turkey
levi@sabanciuniv.edu

***Abstract*—Anti-nodes deployed inside a wireless sensor network can frequently generate dummy data packets that make the nodes relaying them deplete their energy. Especially the nodes closer to the sink fail sooner, because they convey more data packets. This causes the sink to be disconnected from the sensor network. The counter-measures for this type of attacks, namely spam attacks, should consider that the sensor nodes have limited energy, computational power and memory. In this paper, we propose detect and defend against spams (DADS) scheme. In DADS the vicinity of the detected malicious node is notified about the quarantine region, and nodes do not relay unauthenticated messages coming from a node in the quarantine region. Our experiments show that our scheme fits the requirements of the sensor network.**

Keywords-Wireless Sensor Networks, Security, Quarantine Region, Spam Attacks.

I. INTRODUCTION

A sensor network is a collection of sheer number of sensor nodes that collaboratively work in a multi-hop wireless communications architecture. Important features of sensor nodes are outlined below:

- Their lifetime is generally limited with the lifetime of a tiny battery.
- They have limited computational power and memory.
- They are prone to failures.
- They are densely deployed; the distance between two nodes is often less then a few meters.
- Although their location is fixed in many applications, network topology frequently changes due to node failures and objects passing through the sensor field.
- Nodes are supposed to be location aware in many sensor network applications.

Sensor networks have a large set of applications especially in a battlefield because of their flexible, low cost, and self-organizing features. Security is one of the key issues especially in tactical wireless sensor networks. When a sensor network is reachable, sensor nodes can be collected or destroyed by the enemy. The wireless sensor networks that we focus in this paper are the ones deployed in regions which are not accessible for the opposing side. In such networks, fighting against sensor networks by using sensor nodes may be a viable option. Hostile nodes, i.e., anti-nodes, can be deployed inside such sensor networks. Anti-nodes can generate frequent dummy messages, i.e., unsolicited or spam messages. These unsolicited messages may cause nodes, especially the ones close to the sink to fail sooner due to energy depletion. We call this as sensor wars because anti-nodes are used to make the nodes in a sensor field fail.

Various security schemes are introduced for ad hoc networks in [2], [3], [4], and [5]. These solutions cannot be applied directly to the sensor networks, because of the differences between ad hoc and sensor networks. These differences are explained in [1].

SPINS [6] is one of the security schemes proposed for sensor networks that provides data confidentiality, authentication and data freshness. Vulnerabilities and defense mechanisms to DoS (Denial of Service) attacks in a typical sensor network are discussed in [7]. Security, network bandwidth and power consumption in sensor networks are discussed in [8] where two applications have been implemented: target tracking and light sensing. Routing security in wireless sensor networks is discussed in [9].

In this paper, we propose detect and defend spam (DADS) scheme for defending against spam attacks. DADS is based on message authentication in the transmission range of an anti-node. The nodes in the transmission range of the anti-node are called quarantined nodes. Since the number of the quarantined nodes is limited, our scheme does not incur an excessive overhead for security. In this paper, we propose practical solutions for the following issues:

- How to detect spam.
- How to determine quarantined set of sensor nodes.
- How to authenticate messages.
- How to cancel a quarantine region.

The remainder of this paper is organized as follows: In Section II we introduce sensor wars and spam attacks. We explain our DADS scheme as a counter-measure for spam attacks in Section III. The performance of our scheme is evaluated in Section IV. We conclude our paper in Section V.

II. Sensor Wars And Spam Attacks

In sensor wars, the targeted wireless sensor network is neutralized by using anti-nodes scattered randomly inside or close to the network. Anti-nodes, which are much smaller in number comparing to the number of sensors in the network, set spam attacks by generating frequent unsolicited dummy messages and broadcasting them to the neighboring nodes. Hence they increase the data traffic conveyed in the network. All the data generated by the nodes are forwarded to the sink that collects the sensed data from sensor nodes and then relay them to the users or external networks [1]. Number of the nodes close to the sink is limited and they relay much more messages than the other nodes. Therefore, sensors close to the sink are expected to fail earlier than the rest of the network due to energy depletion. This causes the sink to be disconnected from the sensor network. If the attack continues, other sensor nodes exhaust their batteries as well.

Anti-nodes may be fixed or mobile. They can use fixed local identification values or change their identifications as frequent as they need. A mobile anti-node that frequently changes its identification value is harder to detect as compared to fixed cases. However, this is the common case in practice. Therefore in this paper, we focus on the counter-measures against mobile anti-nodes that change their local identifications continuously, which is a more challenging problem.

III. Detect and Defend Spam Protocol

In this section, we explain how to defend against spam attacks. Our scheme mainly consists of the following processes: detecting the spam messages, surrounding the anti-node by putting it in a quarantine region, authenticating the messages in the quarantine region and canceling the quarantine region after anti-node stops sending spam messages.

A. Detecting Spam Attacks

A sink can detect unsolicited messages generated by anti-nodes in two ways. The first method is to filter incoming messages according to their contents and detect the nodes that send faulty messages frequently. Faulty messages can be detected by checking the contradiction between the messages sent by neighboring nodes. The second method uses the frequencies of messages sent by the sensors in the same region. If there is an anomaly in the amount of messages generated by nearby nodes, then this gives a clue about the existence of anti-nodes. In our scheme, messages are detected and filtered by the sink, not by the sensor nodes because of their energy limitations.

The sink can detect unsolicited messages by using the first method explained above, if the anti-nodes do not change their identification values. For the other cases where the anti-nodes change their identification values, the second method would be a better approach. It also better fits the nature of the spam attacks, which can be realized by frequent unsolicited messages. In this method, detection mechanism is based on checking the frequencies of the packets generated by the sensor nodes. A sensor node that generates d times more packets, where d is a system parameter, than the other nodes in the same region is considered as an anti-node.

When the anti-node is mobile, it may not be possible to detect it by comparing its report generation frequency with the report generation frequency of the neighboring nodes. In this case, the packet generation rate of the overall sensor network can be used to detect the spam attacks. If the number of data packets arriving to the sink is in excess of an acceptable level, this may indicate a possible spam attack and the sink can start a network wide alarm.

B. Determining the Set of Quarantined Nodes

In our scheme, authentication is not required in a typical message. The fields of a typical message are shown in Figure 1. *Source id* is the local identification of the source node that generates the *sensed data*. *Source location* is the location of the source node according to a coordinate system, such as polar coordinates, grid coordinates, etc. Location awareness of sensor nodes is generally a requirement for tactical sensor network applications. For example, a target detection data is almost meaningless without a location is associated with it. There are various GPS based, beacon based and beaconless location estimation schemes [10], [11] applicable to the tactical sensor networks. Therefore, it is reasonable to assume that the sensor nodes know their location. *Last hop node id* is the identification of the last node that relays the message. *Last hop location* is the location of the *last hop node*. Every node that relays a message replaces the latter two fields with its identification and location, respectively. The *last hop location* is the same as the *source location* when the message is initially transmitted by the source node. *Sensed data* is the payload of the message.

<table>
<tr><td colspan="2">source id</td><td colspan="2">source location</td><td colspan="2">last hop node id</td></tr>
<tr><td colspan="3">last hop location</td><td colspan="3">sensed data</td></tr>
</table>

Figure 1. The format of a typical message

The sensor node first compares its location with the *last hop location* in an incoming message, and does not repeat the message unless the *last hop location* is closer than the maximum relay distance, d_{max}, which is a system parameter. d_{max} is longer than the transmission range, r, of the sensor nodes, and can be found by

$$d_{max} = \theta \times r. \qquad (1)$$

where q is a multiplication factor which is normally greater than 1.

The quarantined set of nodes and quarantine region are determined dynamically by using a distributed approach. When the sink finds out that there is a spam attack, it broadcasts a defend against spam (DAS) message. When a sensor node receives a DAS message, it does not relay unauthenticated messages during a time period t_q. If it receives an unauthenticated message during t_q, it first requests authentication from the last hop node of the message. If the last hop node fails in authentication, the node assumes that it is in quarantine region and do not relay any data messages unless it is successfully authenticated and it transmits its messages authenticated. In this way, as shown in Figure 2, the quarantine region becomes the region where the transmissions of the anti-node can be received. The authentication algorithm used in DADS is elaborated in Section C. We explain how to cancel a quarantine region in Section D.

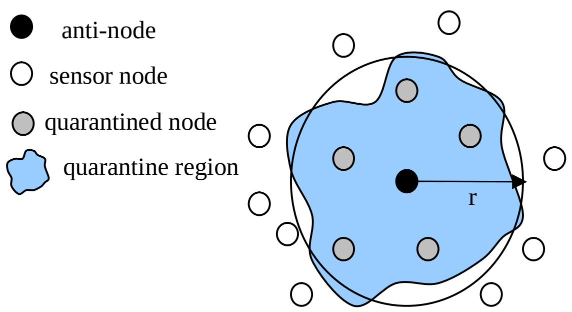

Figure 2. Boundaries of a quarantine region.

C. *Authentication in a Quarantine Region*

When a sensor has a message to send, it first checks if it is in quarantine region. If so, it sends the message authenticated. Moreover, these nodes do not relay any unauthenticated messages. For example, consider Figure 3 that shows a sensor field where a quarantine region is indicated by the gray area. The nodes 3, 4, 7, 8 are in the quarantine region, therefore they have to send and relay only authenticated messages. However, nodes outside the quarantine regions do not need authentication to transmit a message even if the message was an originally authenticated message coming from a quarantine region. For example, node 11 receives authenticated messages from nodes 7 and 8, and transmits them unauthenticated.

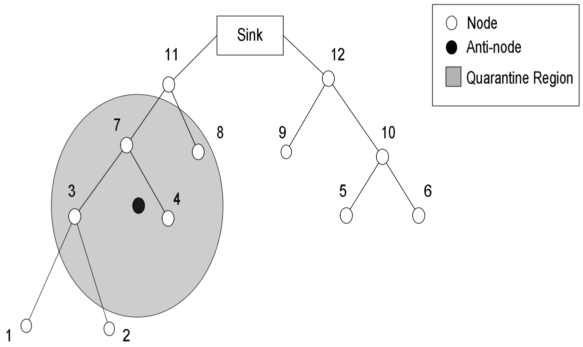

Figure 3. *A sample sensor network and a quarantine region*

Sensor nodes use the proposed DADS protocol for authentication. DADS must be simple enough to fit the stringent constraints of tiny sensors. Therefore it does not use asymmetric and symmetric cryptography, but only cryptographic hash functions. We use the standard HMAC (hash-based message authentication code) mechanism [12], [13] for message authentication.

HMAC uses a cryptographic one-way hash function, such as MD5 [14]. The sender and the receiver share a secret key, *K*. The message authentication code over the message, *M*, is calculated as,

$$\mathrm{HMAC} = \mathrm{H}\,(\mathrm{K} \oplus \mathrm{opad} \,\|\, \mathrm{H}\,(\mathrm{K} \oplus \mathrm{ipad} \,\|\, \mathrm{M})). \tag{2}$$

where ⊕ is the bitwise "exclusive OR" operation, *H* is the underlying one-way hash function, *opad* and *ipad* are two constants defined in [12], [13].

It is necessary to mention about the power consumption of HMAC algorithm. For example, the Berkeley motes [17] consume 1 µJ for transmitting and 0.5 µJ for receiving a single bit, while the CPU can execute 208 cycles (roughly 100 instructions) with 0.8 µJ [18]. We have written the HMAC algorithm in C and assembled it using AVR Studio [19]. We observe that HMAC algorithm consumes approximately 45.6 µJ, if it runs on a Berkeley mote.

In DADS, an authenticated message contains the fields shown in Figure 4. *Source id*, *source location*, *last hop node id*, *last hop location* and *sensed data* fields are the same as the fields in a typical message given in Figure 1. *Sequence number* and *authentication code* fields are added to the message structure in support of authentication. *Sequence number* is the number of outgoing messages. *Authentication code* is the HMAC value.

<table>
<tr><td>source id</td><td colspan="2">source location</td><td>last hop node id</td><td>last hop location</td></tr>
<tr><td colspan="2">sequence number</td><td colspan="2">authentication code</td><td>sensed data</td></tr>
</table>

Figure 4. An authenticated message

Sensors are equipped with the same secret key *K* before deployment. When a sender has a message to send, it first generates the *authentication code* using the HMAC algorithm and the key, *K*. The message over which HMAC is to be calculated contains the *source id*, *source location*, *last hop node id*, *last hop location*, *sequence number* and *sensed data* fields. The *sequence number* is incremented for every outgoing message. After the composition, the authenticated message is transmitted. Any node that should relay this message generates the *authentication code* by using the same algorithm, message and key. If the value calculated at the end of this process is not equal to the value in the *authentication code* field of the incoming message, the message is discarded. Otherwise the message is accepted. This mechanism is depicted in Figure 5.

To facilitate the implementation of HMAC in the sensor nodes, (K ⊕ opad) and (K ⊕ ipad) can be precomputed as offered in [12] and [13]. This implementation is more efficient especially when the message is short. As discussed in [6], sensor nodes use small messages (approximately 30 bytes). Thus, this implementation is suitable for DADS.

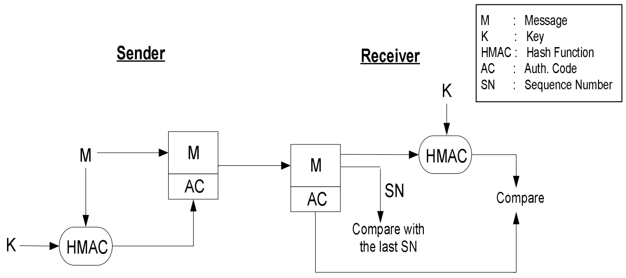

Figure 5. Message authentication in DADS

DADS thwarts possible replay attacks of anti-nodes using the *sequence numbers*. Every sensor node has an internal counter to be used for sequence numbers in outgoing messages. It begins with zero and is incremented by one for each outgoing message. Thus, the *sequence number* of a message created or relayed by a sensor node should be greater than that of every message sent or relayed by the same node before. Each sensor node keeps the last sequence number obtained from each of its neighboring nodes. The freshness of each received message is checked by comparing the *sequence number* in the received message with the last sequence number of the *last hop node id* of that message that is kept locally. If the received message has a higher *sequence number* and *authentication code* is verified, then it is concluded that the message is not a replay and authentic. Such a message is accepted for relaying and the locally kept last sequence number is updated accordingly. Relaying nodes do not accept a message with a *sequence number* which is equal or less than the preceding ones. In such a case, relaying node asks the *authentication code* for the same message but with the expected *sequence number*. If the last node cannot regenerate this *authentication code*, then the message is discarded.

We assume that the sequence number is long enough that it never repeats within the lifetime of the node. Since we do not always use authentication but a node needs authentication only when it is in a quarantine region; we have advantages in making this assumption comparing to the other techniques [6].

One may argue that the *authentication codes* created by a sensor node which is hidden to another node, *n*, can be exploited by anti-nodes. Since those authenticated messages are not received by *n*, the anti-nodes can record and later resend them to *n*. The node *n* accepts those replays as valid and relays them. However anti-nodes can never reach to a significant spam rate by using any of these techniques, because they need to keep pace with the other nodes to use the *authentication codes* generated by them.

One may also try to capture a node and obtain the key by a physical examination. However, DADS aims to prevent spam attacks for the sensor networks deployed in physically inaccessible regions. Moreover, the lifetime of a sensor network in a battlefield is too limited for tampering with a sensor node and obtaining the key out of it.

D. *Canceling the Quarantine Region*

Sensor nodes determine when they will go out of the quarantine region. If a quarantined sensor node does not detect an unsuccessful authentication attempt during the quarantine period t_q, it switches back to not quarantined mode. Sensor nodes start a quarantine period every time they detect an unsuccessful authentication attempt. When a sensor is out of the quarantined set, it sends its messages unauthenticated unless authentication is requested by the relaying node, and it relays also unauthenticated messages. Sensor nodes stay out of the quarantined set until they receive a DAS message from the sink.

IV. THE PERFORMANCE OF OUR SCHEMES

In our simulations, 100 sensor nodes are randomly deployed in the sensor field 200×200 units in size. We use directed diffusion [15] as the data dissemination scheme. We use Matlab 6.0 [16] for our simulations.

In Figure 6, we depict how DADS scheme prevents spam traffic effectively by reducing the number of hops caused by anti-nodes in the network. We simulate the sensor network with 1 to 10 anti-nodes each generating 100 messages. In the average the proposed DADS scheme eliminates 72 % of the traffic caused by anti-nodes.

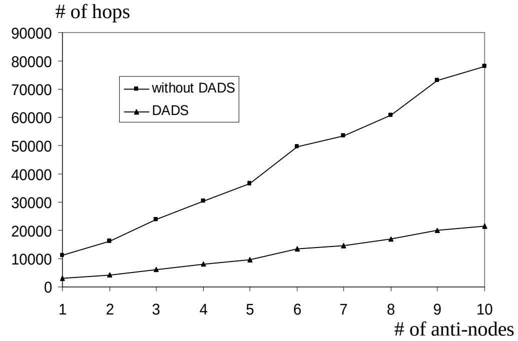

Figure 6. Effect of DADS in a sensor field with 100 sensor nodes and some anti-nodes

Figure 7 depicts the number of authenticated hops versus the number of anti-nodes for different values of d_q, distance between the anti-node and the borderline of the quarantine region. The higher the distance d_q is, the more authenticated hops in the quarantined region, because increase in d_q implies bigger quarantine regions.

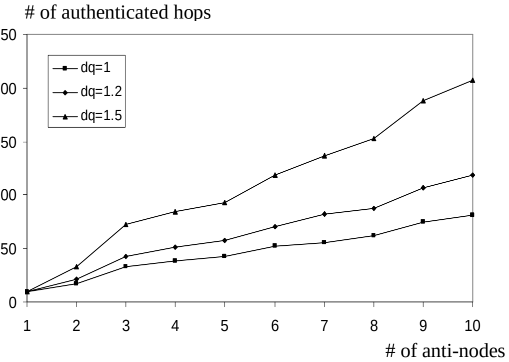

Figure 7. Sensitivity against the changes in d_q

In Figure 8, we depict the effect of spam frequency in DADS with 10 anti-nodes in network. Since it takes some time to detect that there is a spam attack, network performances are very close to each other when the number of spam messages per anti-node is small. Moreover the percentage of traffic eliminated by the DADS scheme increases as the number of messages generated by anti-nodes increases.

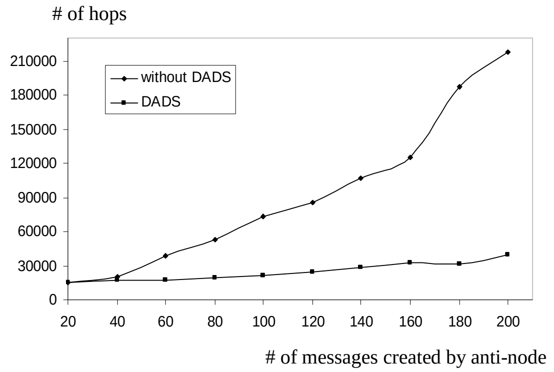

Figure 8. Impact of the spam frequency

We depict the area percentage of the quarantined regions versus the number of anti-nodes in the sensor field in Figure 9. The area of the quarantine regions created by 10 anti-nodes is 48% of the whole sensor field. In other words, in order to neutralize the anti-nodes, only 48% of the whole sensor field should send authenticated messages; the remaining 52% does not have to take the burden of authenticating messages.

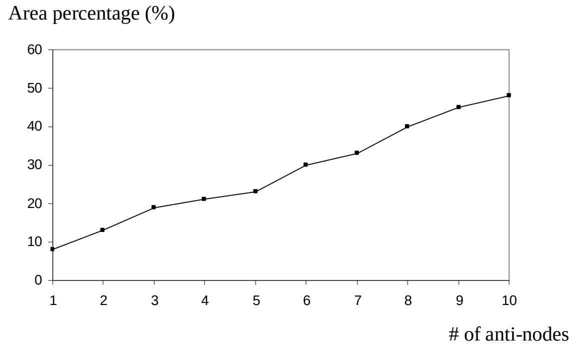

Figure 9. The percentage of the quarantined regions versus the number of anti-nodes

V. CONCLUSION

The defender can deploy some anti-nodes in a tactical sensor network to create spam traffic that causes the sensor nodes to waste their energy by relaying those spam messages towards the sink. We call this attack as spam attack. If a sensor network is not protected against spam attacks properly, even a few anti-nodes can lessen the sensor network lifetime drastically. One solution for spam attack problem is to use authentication mechanism in the sensor network. In this method, normally, all messages in the network have to be sent authenticated to prevent anti-nodes from sending messages. The cost of authenticating all messages through sensor network is costly. In this paper we presented DADS (detect and defend spam) scheme in order to lessen this cost effectively by using local quarantine regions. Authentication mechanism is applied in these local areas. Hence, authentication mechanism is not a burden to all of the sensor nodes in the network. Naturally, authentication in quarantine regions still costs, but this cost is considerably lower than the cost of the case where authentication is applied always in the whole sensor field.

In DADS the sink of the sensor network detects spam messages by checking the frequency of messages sent by the sensor nodes. If an anti-node is spotted, then the sink broadcasts a DAS message. The nodes do not relay unauthenticated messages after receiving a DAS message. If the nodes do not detect an unsuccessful authentication attempt for a system specific time period, they switch back to the normal mode where they can relay also unauthenticated messages.

REFERENCES

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks Journal (Elsevier), Vol. 38, No.4 (2002) 393-422

[2] Stajano, F., Anderson, R.: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. Proc. of the 7th International Workshop on Security Protocols, Berlin (2000) 172-182

[3] Karpijoki, V.: Security in Ad Hoc Networks. Proc. of the Helsinki University of Technology Seminar on Network Security (2000)

[4] Zhou, L., Haas, Z.J.: Securing Ad Hoc Networks. IEEE Network Magazine, vol. 13, no.6, pp. 24-30 (1999)

[5] Hubaux, J.P., Buttyan, L., Capkun, S.: The Quest For Security in Mobile Ad Hoc Networks. Proc. of ACM Symposium on Mobile AdHoc Networking and Computing (MobiHOC 2001) (2001)

[6] Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: SPINS: Security Protocols for Sensor Networks. Proc. of ACM MobiCom'01, Rome, Italy (2001) 189-199

[7] Wood, A.D., Stankovic, J.A.: Denial of Service in Sensor Networks. IEEE Computer Magazine (2002) 54-62

[8] Chen, M., Cui, W., Wen, V., Woo, A.: Security and Deployment Issues in a Sensor Network. http://www.cs.berkeley.edu/ wdc/classes/cs294-1-report.pdf. (2000)

[9] Karlof, C., Wagner, D.: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. 1st IEEE International Workshop on Sensor Network Protocols & Applications (SNPA 2003). (2003)

[10] Nasipuri, A., Li, K.: A Directionality Based Location Discovery Scheme for Wireless Sensor Networks. Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications, Atlanta (2002) 105-111

[11] Savvides, A., Park, H., Srivastava, M.B.: The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems. Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications (2002) 112-121

[12] Stallings, W.: Cryptography and Network Security. Prentice Hall, Third edition, 2003.

[13] Krawczyk, H., Bellare, M., Canetti, R.: RFC 2104 - HMAC: Keyed-Hashing for Message Authentication. (1997)

[14] Rivest, R. L.: RFC 1321-The MD5 message-digest algorithm. (1992)

[15] Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. Proc. of the 6th Annual International Conference on Mobile Computing and Networks (MobiCOM'00) (2000)

[16] http://www.mathworks.com/products/matlab

[17] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System Architecture Directions for Networked Sensors. ASPLOS 2000.

[18] Bhattacharya, S., Kim, H., Prabh, S., Abdelzaher, T.F.:Energy-Conserving Data Placement and Asynchronous Multicast in Wireless Sensor Networks. MobiSys 2003.

[19] http://www.atmel.com/products/AVR