01 Mar 2009

Generalized Neuron Based Secure Media Access Control Protocol for Wireless Sensor Networks

Raghavendra V. Kulkarni

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Abhishek V. Thakur

Sanjay Kumar Madria
Missouri University of Science and Technology, madrias@mst.edu

Recommended Citation

R. V. Kulkarni et al., "Generalized Neuron Based Secure Media Access Control Protocol for Wireless Sensor Networks," *Proceedings of the IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making, 2009. MCDM'09*, Institute of Electrical and Electronics Engineers (IEEE), Mar 2009.
The definitive version is available at https://doi.org/10.1109/MCDM.2009.4938823

Generalized Neuron Based Secure Media Access Control Protocol for Wireless Sensor Networks

Raghavendra V. Kulkarni, *Senior Member, IEEE,* Ganesh K. Venayagamoorthy, *Senior Member, IEEE,*
Abhishek V. Thakur and Sanjay K. Madria, *Senior Member, IEEE*

Abstract—**Security plays a pivotal role in most applications of wireless sensor networks. It is common to find inadequately secure networks confined only to controlled environments. The issue of security in wireless sensor networks is a hot research topic for over a decade. This paper presents a compact generalized neuron (GN) based medium access protocol that renders a CSMA/CD network secure against denial-of-service attacks launched by adversaries. The GN enhances the security by constantly monitoring multiple parameters that reflect the possibility that an attack is launched by an adversary. Particle swarm optimization, a popular bio-inspired evolutionary-like optimization algorithm is used for training the GN. The wireless sensor network is simulated using Vanderbilt Prowler, a probabilistic wireless network simulator. Simulation results show that the choice of threshold suspicion parameter impacts on the tradeoff between network effectiveness and lifetime.**

I. Introduction

A wireless sensor network (WSN) is a network of distributed autonomous devices that monitor physical or environmental conditions cooperatively [1], [2]. WSNs are used in varieties of applications such as environmental monitoring, habitat monitoring, prediction and detection of natural calamities, medical monitoring and structural health monitoring. WSNs consist of a large number of small, inexpensive, disposable and autonomous sensor nodes (or motes) that are deployed in an ad hoc manner for remote operations. Sensor nodes are severely constrained in terms of data storage resources, computational capabilities, communication bandwidth and power supply [3]. MICA2 is a popular commercially available sensor mote [4].

Communication tasks consume maximum power available to sensor nodes, and in order to ensure a sustained long-term sensing operation, communication tasks need to be exercised frugally. Nodes may cease to function due to physical damage or power failure; and surviving nodes may go in or out of transmission radii of other nodes due to dynamic environment. Resource constraints and dynamic topology pose technical challenges in network discovery, network control and routing, collaborative information processing, querying, and tasking [3]. Major research areas in WSNs include data aggregation, energy aware routing, dynamic node scheduling, optimal node deployment, self organization, security, node localization and quality-of-service assurance.

This paper presents a secure medium access control (MAC) protocol to enhance the security of a network of MICA2 sensor nodes by detecting (and counteracting to) the denial-of-service (DoS) attacks launched by adversaries. Simulation is carried out in Vanderbilt Prowler, a probabilistic wireless network simulator. The key parameters that reflect the security level are monitored by a compact generalized neuron (GN) [5] on each node, which stops the MAC layer activities if it detects a security breach. The GN is trained using particle swarm optimization (PSO), a popular bio-inspired multidimensional optimization algorithm [6]. A small number of trainable weights and a low computational complexity of the GN makes it suitable to be loaded on a sensor node. Besides, PSO-based training assures accurate and quick convergence of the weights to their final values.

The rest of this paper is organized as follows: A brief overview of the recent research in security issues in ad hoc and sensor networks is presented in section II. DoS attacks and the countermeasures proposed by various researchers are discussed in section III. The MICA2 mote and the Prowler simulation environment are discussed in section IV. The structure of a GN is explained in section V. The network scenario implemented in order to demonstrate the secure MAC is explained in section VI. Training of the GN through PSO are discussed in section VII. The results obtained are discussed in section VIII. And finally, concluding remarks are made in section IX.

II. Related work

Wireless links are susceptible to eavesdropping, impersonating and message distorting. Poorly protected nodes that move into hostile environments can be easily compromised. Authorization of administration becomes difficult due to dynamic topology. The scale of deployment of wireless sensor network requires careful decision about trade-offs among various security measures. These issues are discussed and mechanisms to achieve secure communication in sensor networks are presented in [7]. Various security challenges in wireless sensor networks are analyzed and key issues that need to be addresses for achieving security are summarized in [8].

Secure routing is one of the main research areas in the recent times. Types of routing attacks and their countermeasures are presented in [9]. Secure routing in an ad hoc network is a

Authors acknowledge the support received for this work by the National Science Foundation under the grant *SENSORS: Approximate Dynamic Programming for Dynamic Scheduling and Control in Sensor Networks*, ECCS # 0625737.

R. V. Kulkarni and G. K. Venayagamoorthy are with the Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science and Technology, Rolla, MO, USA. (e-mail: {arvie, gkumar}@ieee.org).

Abhishek V. Thakur is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA. (e-mail: avtf3d@mst.edu).

Sanjay K. Madria is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA. (e-mail: madrias@mst.edu).

978-1-4244-2764-2/09/$25.00 ©2009 IEEE

daunting task because of some contradictions between the nature of the network and the associated applications. In [10], various routing protocols have been presented with a focus on finding security vulnerabilities. In article [11], a survey of secure ad hoc routing protocols for mobile wireless networks is presented.

In spite of a large number of secure protocols in literature, it is unclear what properties should the protocols achieve, as a formal analysis of these protocols is mostly lacking. Article [12] is concerned with the problem of specifying and proving correctness of a secure routing protocol. Varieties of secure routing protocols have been presented in [13] and [14]. A new secure routing protocol for mobile ad hoc networks based on advanced on-demand distance vector (AODV) [15] called AODV-SEC is presented in [16]. A secure routing protocol, which enhances the security aspects of AODV with a very negligible byte overhead is presented in [17]. A secure routing mechanism based on trust is presented in [18].

Security is not confined to the realm of routing algorithms alone. Researchers have proposed several methods of securing the MAC layer against the attacks by adversaries. DoS attacks and their countermeasures at the CSMA/CA MAC layer are discussed in [19] and [20]. In both these articles, the current security level in the network is assessed by monitoring three critical parameters: collision rate, rate of arrival of Request-to-Send (RTS) packets, and the average waiting time of a packet in the MAC buffer. An abnormal rise in one or more of these parameters is construed as an attack. While the work in [19] uses the parameters in a deterministic way, the fuzzy logic approach is used in [20] to estimate the security and to shut down the MAC in case of a strong suspicion of an attack.

III. Security Against DoS Attacks

The work presented in [19] and [20] aim at detecting and counteracting the DoS attacks launched by adversaries. The DoS attacks are classified into three groups, namely, collision, unfairness and exhaustion attacks. In collision attacks, the attackers transmit packets regardless of the status of the broadcast medium. The packets collide with data or control packets from legitimate sensor nodes. In unfairness attacks, adversaries transmit an unusually large number of packets if the medium is free. This prevents the legitimate sensors from transmitting their packets. In exhaustion attack, adversaries transmit abnormally large number of RTS packets to normal sensor nodes and exhaust them prematurely. A more detailed description of the attacks is available in [20].

Articles [19] and [20] show that DoS attacks can be detected if abnormally large variations occur in sensitive parameters such as collision rate R_c (number of collisions observed by a node per second), average waiting time T_w (waiting time of a packet in MAC buffer before transmission), and RTS arrival rate (R_{RTS}) (number of RTS packets received successfully by a node every second).

In [19], the probability that an attack has been perpetrated is estimated using a decision function that involves two parameters, which are determined using the steepest gradient descent algorithm. On the other hand, in [20], fuzzy logic is creatively used wherein the parameters R_c, R_{RTS} and T_w are fuzzified, and evaluated by a fuzzy inference engine, which produces a binary output to trigger a mechanism to counteract the attack. Both the above methods are reported to extend the security and thus, lifetime of the sensor networks.

This study extends the idea of the secure-MAC discussed in [20] to the MICA2 nodes that employ CSMA/CD protocol in their MAC layer. The most important modification is the use of a GN structure to monitor for a security breach. The best values for the trainable parameters in the GN are determined using the PSO algorithm. The GN has a compact structure, and it uses less computational and storage resources available to the sensor nodes. At the same time, PSO-based training converges quickly and accurately to the best training parameters. Therefore, the approach proposed here has a twofold advantage. The approach is illustrated in the block diagram shown in Figure 1.

Each sensor node in the WSN proposed here has a pre-trained GN running on its MAC layer. Critical parameters collision rate (R_c), packet request rate (R_r) and average packet waiting time (T_w) extracted from the simulator environment are the inputs to the GN. The GN computes the level of suspicion that denotes the probability that there is a security breach. If this output is greater than a predefined threshold level, then the node shuts itself down and saves energy, which is its most crucial commodity. Security is distributed throughout the network in the sense that each node has a GN, and only the node which suspects a security breach shuts itself down. It is quite likely that an attack affects only one geographical area of the network, and therefore, only the nodes in that area need to be shut down. The network of nodes that use the proposed MAC layer has this ability.

IV. MICA2 Mote and the Prowler Simulation Environment

MICA, A low-cost prototype field-node family was developed at University of California, Berkeley. MICA2, an enhanced version of this prototype is commercially manufactured by Crossbow Technology, Inc. MICA2 includes an 8-bit, 4 MHz Atmel ATMEGA103 microcontroller, 128kB program memory, 4KB RAM, and an RFM TR1000 radio chip capable

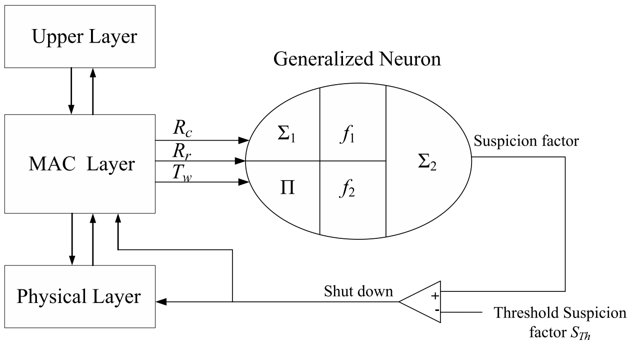


Fig. 1. GN based secure MAC protocol

of providing 50 kbit/s transmission rate at 916.5 MHz. The motes can also accommodate a set of interchangeable sensors to measure varieties of physical parameters like temperature, light and sound. MICA2 runs an embedded operating system called TinyOS, designed to provide the necessary services in spite of limited hardware resources. It contains a complete network stack with bit-level error correction, medium access layer, network messaging layer, and timing.

The MAC layer of MICA2 uses a simple CSMA/CD protocol: it waits for a random duration before trying to transmit a packet and then waits for a random back-off interval if the channel is found busy. It keeps trying until the transmission can be performed. This simple approach is not as effective as the more sophisticated protocols like IEEE 802.11 in terms of collision avoidance, but it consumes less energy and the communication overhead is much smaller.

Probabilistic wireless network simulator Prowler captures the event-driven nature of TinyOS. The simulator can be set to operate either in deterministic mode or in probabilistic mode. The latter simulates the nondeterministic nature of the communication channel and the low-level communication protocol of the MICA2 motes [21]. Prowler can incorporate an arbitrary number of stationary or mobile motes and it can be easily embedded into optimization algorithms. The simulator core runs on MATLAB, which provides a fast and easy way to build applications. It has attractive visualization effects for easy interpretation of debugging.

A. Radio propagation models

Prowler's radio propagation model determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. It models the decay of signal strength with distance. It also models the fading effect, the time-varying nature of the signal strength, and other miscellaneous transmission errors. Strength of the signal received at a node at a distance d from a transmitter under ideal conditions is given by (1).

$$P_{rec,ideal}(d) = P_{transmit}\frac{1}{1+d^{\gamma}} \tag{1}$$

where $P_{transmit}$ is the power transmitted and γ is the decay parameter, $2 \leq \gamma \leq 4$. However, the power of signal the node j receives from a node i in real environments is given differs from the ideal value. Prowler uses two fading models. In model 1 the signal is received if the signal strength is greater than a user definable reception limit parameter. In model 2, Raleigh fading model is used. Model 1 is simple and fast, while Model 2 is more accurate. Prowler enables the network designers to choose a radio model from the two available models. Moreover, various environment parameters can be appropriately chosen.

B. Simulation application

Prowler applications are event based. The interactions between MAC layer and the application take place through the following aptly named events: *Init_Application*, *Packet_sent*, *Packet_received*, *Collided_Packet_Recieved*, *Clock_Tick*, *Application_Finished*, and *Application_Stopped*. The application can trigger the actions *Set_Clock* and *Send_Packet*. In addition, debug and visualization actions can be invoked. These actions include *Print_Message*, *LED* and *Draw_Line*.

C. The MAC layer

Prowler's MAC layer is modeled as a sequence of events. When the application invokes the *Send_Packet* event, the MAC layer goes idle for a random *Waiting_Time* after which it senses if the broadcast channel is free. This is the basic essence of CSMA. If channel is not free, the MAC layer frequently checks if it is free. Before each check, it waits for a *Backoff_Time*. Both *Waiting_Time* and *Backoff_Time* can be random, user defined a combination of the two. When the channel is found free, the MAC layer transmits the constant sized packet (960 bytes), for the *Transmission_Time*, and notifies the application through a *Packet_Sent* report. When a packet is received, the MAC layer reports it to the application through a *Packet_Received* or a *Collided_Packet_Received* message as the case may be.

V. The Generalized Neuron

It is shown in the literature that multilayer perceptrons (MLPs) are universal approximators of continuous functions for the given input-output patterns [22]. The general structure of a typical MLP contains an aggregation function and an activation function. A typical neuron uses summation or multiplication as aggregation function and a hard-limiter, log sigmoidal, radial basis, or linear activation function [23]. In applications in WSNs, memory constrains in the sensor nodes call for neural networks that use a small number of trainable parameters. GN is a neural network model that is more compact and flexible than MLPs [24], [5]. The GN used here uses both summation and multiplication as aggregation functions, and both sigmoid and Gaussian activation functions. Therefore, the GN has flexibility and resilience to the nonlinearities of real world problems. The compact structure of a GN is shown in Figure 2.

A GN uses both Σ (sum) and Π (product) aggregation functions. The weighted vector of inputs $\vec{X}$ is summed up by aggregation function Σ_1. Output of this unit is processed by an activation function f_1. Similarly, weighted inputs are multiplied by aggregation function Π. Output of this unit is processed by a different activation function f_2. Weighted

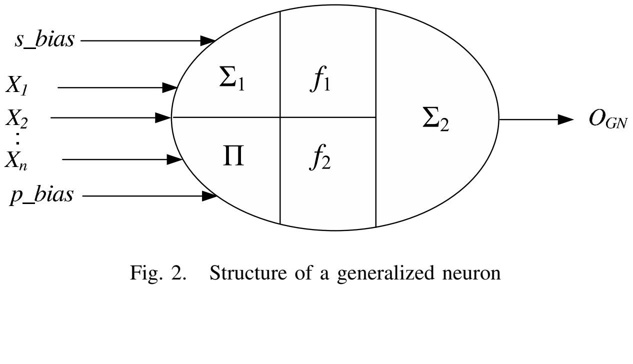


Fig. 2. Structure of a generalized neuron

outputs of these two units are summed up. Two different sets of activation and aggregation functions endow the GN with the flexibility that is not possible in the MLPs having the same number of trainable parameters.

The Σ section of the GN is associated with summation of weighted inputs, and it uses the sigmoidal activation function. Its output is given by (2),

$$O_{\Sigma} = f_1(s\_net) = \frac{1}{1 + \exp(-\lambda_s \times s\_net)} \tag{2}$$

where

$$s\_net = \sum W_{\Sigma i} X_i + X_{o\Sigma} \tag{3}$$

Here, W_{Σ} are input weights, $X_{o\Sigma}$ is bias weight of Σ section and λ_s is the gain factor of Σ section. The Π section of the GN is associated with multiplication of weighted inputs. It uses the Gaussian activation function given by (4).

$$O_{\Pi} = f_2(pi\_net) = \exp(-\lambda_p \times pi\_net^2) \tag{4}$$

where

$$pi\_net = \prod W_{\Pi i} X_i \times X_{o\Pi} \tag{5}$$

Here, W_{Π} are input weights, $X_{o\Pi}$ is bias weight of Π section and λ_p is the gain factor of Π section. The output is obtained as in (6)

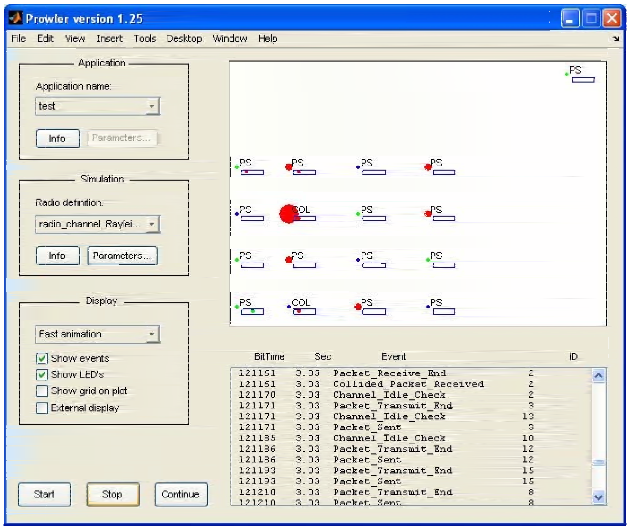
$$O_{GN} = O_{\Pi} \times (1 - W) + (O_{\Sigma} \times W) \tag{6}$$

A GN has multiple inputs but only one output. If multiple outputs are desired, then as many GNs will have to be used. A GN having n inputs has $(2n + 1)$ weights and two biases, a total of $(2n + 3)$ trainable parameters. Either or both gain factors λ_s and λ_p can be taken as trainable parameters as well, in which case, their total number increases accordingly. Other activation functions like sine, cosine, or hyperbolic tangent can also be used. Because weighted outputs of Σ and Π sections of the proposed GN are added, this type of GN is called the summation type GN. Weighted outputs of Σ and Π sections can be multiplied to construct a multiplication type GN.

VI. The Scenario for the GN-Based Secure MAC Protocol

In order to demonstrate the secure MAC layer protocol proposed in this work, a test WSN scenario is implemented as shown in the screen shot in Figure 3.

The mission space is a two-dimensional plane having its origin at the lower left corner. The deployment scenario involves 17 sensor nodes having unique IDs from 1 through 17. Nodes 1 through 16 are placed in a 4×4 square grid. Distance between two consecutive nodes in a column, or a row is three units. Node 17 is placed away from the rest, at coordinates (20,20). This node can be moved to any place in the scenario. Each of the nodes one through 16 attempts to transmit a packet at every 0.25 seconds with a probability P. However, because random *Waiting_Time* and *Backoff_Time*, all nodes do not transmit simultaneously. If two nodes transmit simultaneously, their neighbors get collided packets. The number of request packets a node receives in a minute is measured as the request rate R_r. Each node that needs a data packet from another node sends a request packet with a probability of $P = 0.1$. The recipient node responds to the request by sending a data packet. Node 17 never receives a packet because it is too far away from others to receive a sufficiently strong signal. Average time for which a packet waits in the node buffer before it is transmitted is measured as the average waiting time T_w. Average number of collisions in a minute is measured as the collision rate R_c. These parameters can be determined from the event responses obtained from the Prowler environment.



Application name: test

Radio definition: radio_channel_Raylei...

Fast animation

☑ Show events

☑ Show LED's

□ Show grid on plot

□ External display

```
BitTime   Sec    Event                        ID
121161    3.03   Packet_Receive_End            2
121161    3.03   Collided_Packet_Received      2
121170    3.03   Channel_Idle_Check            2
121171    3.03   Packet_Transmit_End           3
121171    3.03   Channel_Idle_Check           13
121171    3.03   Packet_Sent                   3
121185    3.03   Channel_Idle_Check           10
121186    3.03   Packet_Transmit_End          12
121186    3.03   Packet_Sent                  12
121193    3.03   Packet_Transmit_End          15
121193    3.03   Packet_Sent                  15
121210    3.03   Packet_Transmit_End           8
121210    3.03   Packet_Sent                   8
```

Fig. 3. The test scenario for GN-MAC secure protocol against DoS attacks

To begin with, all nodes are endowed with batteries that have random power levels between 500 and 1000 units. All the nodes numbered one through 16 transmit with a unit power level. The simulation shows successful packet transmissions and collisions.

Node 17 turns into an adversary and launches a DoS attack when it is placed close to other nodes. The three types of DoS attacks are simulated in this work. In normal case, because it is away from other nodes, node 17's MAC will always find the channel to be free. It transmits packets with probability of unity and power level of eight units. This transmission will collide with the packets on the broadcast medium, causing a substantial rise in the number of collisions. This attack is a collision attack, which results in an unusual rise in the collision rate R_c. In an unfairness attack, node 17 sends repeated request packets to a particular node, node 1 in this simulation. Node 1 repeatedly sends sensor data packets in repones to requests by node 17, and exhausts its energy reservoir. This results in an unusual rise in R_r. In an unfairness attack, node 17 transmits persistently, but at the same power level as the other nodes transmit (power level of 1 unit in this simulation). This forces other nodes in the vicinity to remain silent for unusually long periods, causing an unusual rise in T_w.

The GN senses the rise in critical parameters R_c, R_r and T_w and produces the output that shuts down the node. The values of the critical parameters in presence and absence of different types of attack are recorded in 50 trial runs and their mean and standard deviation is computed. Table I shows the values of the critical parameters in absence and presence of the three types of attacks.

The output of GN is used as a measure of suspicion of an attack. When the suspicion of an attack exceeds a predefined threshold suspicion level S_{Th}, the MAC and physical layer of the node is switched off.

VII. Training of the GN Using PSO

Training of an MLP involves adjusting its parameters iteratively in such a way that the mean-square-error between the desired and actual outputs for all training patterns is equal to or lesser than a preset tolerance. Backporpagation algorithm (BPA) is a popular training algorithm for MLPs [23]. Particle swarm optimization (PSO) has been used as a training algorithm for neural networks, and it is shown to be more accurate and computationally efficient than BPA [25]. This is the motivation for the use of PSO to train the GN in this study.

PSO is a population based parallel search algorithm that models social behavior of birds within a flock [6]. PSO consists of a population (or a swarm) of particles, each of which represents a potential solution. Particles are initially assigned random positions and velocities. The direction of position change is influenced by both particle's experience and the knowledge a particle acquires from the flock. Each particle is evaluated using a fitness function which indicates how close the particle is to the global solution. It is desired to maximize the fitness as the PSO iterations progress. Several versions of PSO have been proposed [26].

Each particle has a memory where it stores the knowledge of position p_{id}, which is defined position at which the particle had maximum fitness. Besides, the best of p_{id} of all particles, called p_{gd}, is stored too. At each iteration k, PSO adds velocity v_{id} to the position x_{id} of each dimension in a particle and steers the particle towards its p_{id} and p_{gd} using (7) and (8). This paper uses the global best (g_{best}) version of PSO.

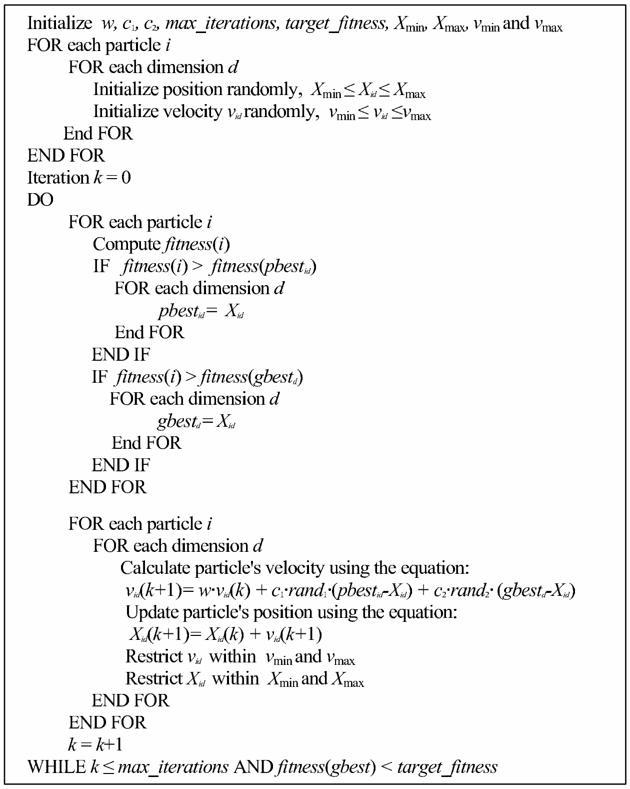
$$\begin{aligned} v_{id}(k+1) &= w \cdot v_{id}(k) + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}) \\ &\quad + c_2 \cdot rand_2 \cdot (p_{gd} - x_{id}) \end{aligned} \tag{7}$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \tag{8}$$

TABLE I
Critical parameters in absence and presence of an attack averaged over 50 trial runs

	R_c	R_r	T_w
Normalcy	3.2 (1.13)	23.5 (11.72)	1223 (988)
Collision Attack	19.3 (2.01)	29.7 (10.3)	10211 (1324)
Exhaustion Attack	4.5 (1.2)	42.3 (27.2)	1868 (1158)
Unfairness Attack	7.9 (3.3)	32.4 (9.3)	1791 (329)

The umbers shown in brackets represent standard deviation.



```
Initialize  $w, c_1, c_2, max\_iterations, target\_fitness, X_{\min}, X_{\max}, v_{\min}$ and $v_{\max}$
FOR each particle $i$
    FOR each dimension $d$
        Initialize position randomly,  $X_{\min} \leq X_{id} \leq X_{\max}$
        Initialize velocity $v_{id}$ randomly,  $v_{\min} \leq v_{id} \leq v_{\max}$
    End FOR
END FOR
Iteration $k = 0$
DO
    FOR each particle $i$
        Compute $fitness(i)$
        IF  $fitness(i) > fitness(pbest_{id})$
            FOR each dimension $d$
                $pbest_{id} = X_{id}$
            End FOR
        END IF
        IF  $fitness(i) > fitness(gbest_d)$
            FOR each dimension $d$
                $gbest_d = X_{id}$
            End FOR
        END IF
    END FOR

    FOR each particle $i$
        FOR each dimension $d$
            Calculate particle's velocity using the equation:
            $v_{id}(k+1) = w \cdot v_{id}(k) + c_1 \cdot rand_1 \cdot (pbest_{id} - X_{id}) + c_2 \cdot rand_2 \cdot (gbest_d - X_{id})$
            Update particle's position using the equation:
            $X_{id}(k+1) = X_{id}(k) + v_{id}(k+1)$
            Restrict $v_{id}$  within  $v_{\min}$ and $v_{\max}$
            Restrict $X_{id}$  within  $X_{\min}$ and $X_{\max}$
        END FOR
    END FOR
    $k = k+1$
WHILE $k \leq max\_iterations$ AND $fitness(gbest) < target\_fitness$
```

Fig. 4. Pseudocode for PSO

Maximum values chosen for position x_{id} and velocity v_{id} of a particle are 100 and 2 respectively. Inertia weight w is reduced linearly in every iteration from 0.9 in the beginning to 0.4 at the end. Cognition and social acceleration constants c_1 and c_2 are chosen as 2 as done generally. Pseudocode for the PSO based training algorithm is given in Figure 4.

The GN used for this task has three inputs X_1, X_2 and X_3 which are the parameters R_c, R_r and T_w and respectively. The GN uses just 9 trainable parameters (the number of inputs $n = 3$). The GN is trained to compute the attack suspicion factor, which refers to how strongly the GN suspects that an attack is launched. Numerous experiments are conducted to generate a training pattern set containing 50 patterns. The patterns range from the normal scenario without an attack (where suspicion is zero) to the most vicious attack (unit suspicion factor). The first training set has inputs equal to the average values of R_c, R_r and T_w under no attack, and the 50th pattern has inputs equal to the average values of the same parameters at the most vicious attack (all attacks launched simultaneously). The value of a parameter X_i, where i=1, 2 or 3, in the j^{th} training pattern is obtained as in (9). The value of the outputs in each of the 50 training patterns is shown in Figure 5. The GN is trained to approximate this input-output relation.

$$X_i^j = X_i^{\min} + (j-1) \times \frac{X_i^{\max} - X_i^{\min}}{50} \tag{9}$$

VIII. Simulation Results and Discussions

PSO algorithm having 30 particles is executed for 2000 iterations in the training phase. The GN achieves good learning of the problem. The training target suspicion factor, and the suspicion factor that the GN came up with after it learnt the problem are depicted in Figure 5.

The training pattern set containing 50 patterns is applied during the PSO based training. Figure 6 depicts the reduction of training error as the training iterations progress. Average training error of 2.176×10^{-4} is achieved in 50 training trials. After the training is complete, the final values of trainable parameters are loaded on the GNs on each of the nodes of the sensor network for real-time use.

The choice of the threshold suspicion factor S_{Th} strongly influences the network behavior. Different nodes can have different values of S_{Th} depending on where in the network the nodes are located. If the GN on a node produces an output (suspicion factor) equal to or greater than the S_{Th} , the node's physical layer is turned off. After a time of 30 seconds, the node's physical layer is turned on again, and the sensor would return to its normal working state under the watchful monitoring of the GN. This saves the energy the node would have spent in collisions or transmissions in response to adversary's malicious requests.

In this simulation study, 50 trial runs are conducted, for each of the four values of the threshold suspicion factor. The average network lifetime before the first node death and the number of packets the node transmits successfully are recorded. In each trial run, all nodes are assigned equal initial energy, represented by a variable. This variable is decremented after each packet is transmitted, regardless of whether or not the packet succeeds in reaching its destination. Simulation is carried out until the variable reaches zero, which represents the node death due to exhaustion of energy. In addition, packets that reach their destination without collision are counted. This gives the number of successful transmissions. A summary of the results is presented in Table II.

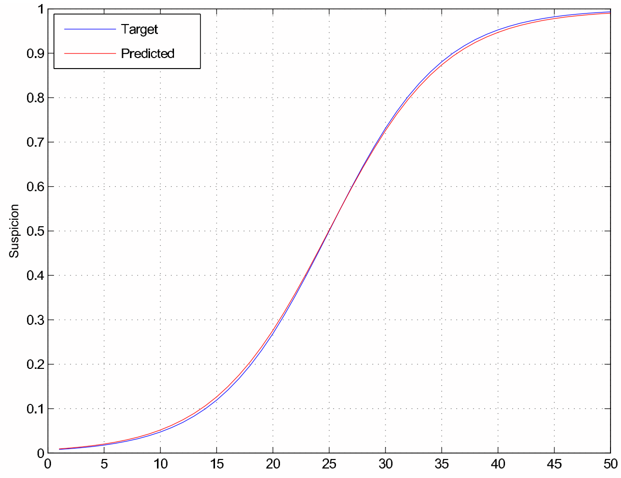


Fig. 5. Results of GN training for secure MAC

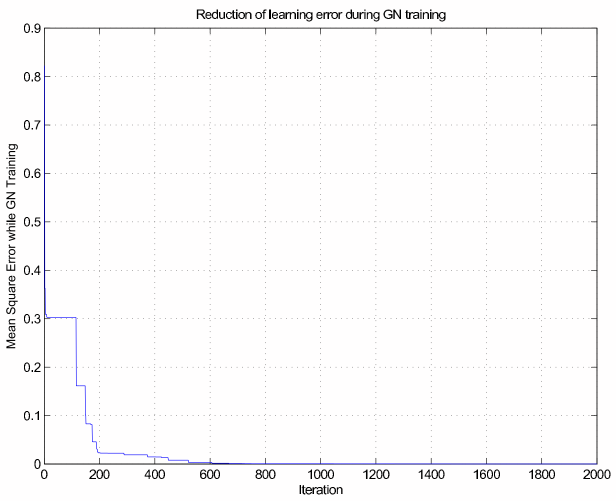


Fig. 6. Reduction of training error during GN training

The case with $S_{Th} > 1$ represents a passive GN (the GN never produces $S_{Th} > 1$). Here, the node is not shut down even in case of an attack. This corresponds to a large number of collisions and retransmissions, which result in a short network life and a small number of successful packets. The results show a strong presence of false triggers the threshold suspicion factor of 0.4. The GN suspects an attack even if there isn't any. Though this gives an impression that the network life is higher at a low value of S_{Th}, it is not. Due to false triggering, the node frequently shuts down itself which can be inferred from a low number of successful transmissions. At this value of S_{Th}, the network is not very effective though it lives reasonably long. On the contrary, at $S_{Th} = 0.8$ number of false triggers is zero. Here, the node shuts down only when it strongly suspects a real attack. Due to this, the number of successful transmissions is higher. This situation represents n a tradeoff between network's effectiveness and its lifetime. The results with $S_{Th} = 0.6$ show a mix of false and true triggers. This is where the multicriteria decision making comes into play. The choice of the best value of S_{Th} depends on how long the network is expected to live and how effective it is expected to be.

IX. Conclusions and Future Work

This paper discusses the use of a GN to watch for DoS attacks in wireless sensor networks that uses CSMA/CD protocol. Prowler, a probabilistic wireless network simulator is used to simulate a GN based secure MAC layer for a network of MICA2 motes. Each node in the network hosts a pre-trained compact GN which watches the crucial parameters collision rate R_c, packet request rate R_r and packet waiting time T_w, and computes a measure of suspicion. In a DoS attack, these parameters vary in such a way that the suspicion factor increases in value. If the suspicion factor exceeds a preset threshold level, the node's physical layer is switched off. This results in saving of power that would have been wasted in retransmission of collided packets. The results show that the

TABLE II

A SUMMARY OF RESULTS OF 50 TRIAL RUNS EACH AT DIFFERENT SETTINGS OF THE THRESHOLD SUSPICION FACTOR

Threshold suspicion factor S_{Th}	Number of false triggers	Number of true triggers	Network life in minutes	Number of packets transmitted successfully
0.4	31	19	16.46	702
0.6	14	36	12.78	1091
0.8	0	50	8.61	1738
> 1*	0	0	5.09	547

*This represents the case in which the GN does not intervene in the MAC.

choice of the threshold suspicion factor facilitates a tradeoff between the network throughput and the network life-time. The PSO-trained GN is not expensive in terms of storage and computational time. The proposed scheme provides distributed security against the collision attacks because only the node which suspects an attack shuts itself. Simulation results show that the power saving due to shutting down the attacked nodes results in reduction in power wastage, which in turn extends the network life.

This study can be extended in several directions. If there is a surge of activities in normal conditions without any attacks, a GN on a node can trigger a false alarm causing the node to shut down. Such false triggering can prove to reduce the effectiveness of the network. An investigation on constant online training of the GN is one direction in which the work can be extended. Investigation remains to be carried out on the extent to which the network life is extended under different types of attacks. This needs an energy model to be built into the proposed simulation, which is a possible extension of the study. Testing a real MICA2 network rather than a simulated one is the most obvious necessity, which will establish the real time applicability of the proposed method.

REFERENCES

[1] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] C. Y. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[4] The MICA2 datasheet. [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_-pdf/MICA2_Datasheet.pdf

[5] R. Kiran, G. Venayagamoorthy, and M. Palaniswami, "Density estimation using a generalized neuron," in *Proc. 9th International Conference on Information Fusion ICIF '06*, July 2006, pp. 1–7.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, Perth, Australia, Jan. 1995, pp. 1942–1948.

[7] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Commun. Mag.*, vol. 11, no. 6, pp. 38–43, 2004.

[8] F. Hu and N. K. Sharma, "Security considerations in ad hoc sensor networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 69–89, 2005.

[9] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, no. 2–3, pp. 293–315, September 2003.

[10] A. R. M. Kamal, "Adaptive secure routing in ad hoc mobile network," Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, Nov 2004.

[11] P. Argyroudis and D. O'Mahony, "Secure routing for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 3, pp. 2–21, 2005.

[12] P. Papadimitratos, Z. Haas, and J. P. Hubaux, "How to specify and how to prove correctness of secure routing protocols for MANET," in *Proc. 3rd Int. Conf. on Broadband Communications, Networks and Systems BROADNETS 2006*, Z. Haas, Ed., 2006, pp. 1–10.

[13] S. M. Nikjoo, A. S. Tehrani, and P. Kumarawadu, "Secure routing in sensor networks," in *Proc. Canadian Conf. on Electrical and Computer Engineering CCECE 2007*, A. Saber Tehrani, Ed., 2007, pp. 978–981.

[14] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure routing and intrusion detection in ad hoc networks," in *Proc. 3rd IEEE Int. Conf. on Pervasive Computing and Communications PerCom 2005*, J. Parker, Ed., 2005, pp. 191–199.

[15] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications WMCSA '99*, 25–26 Feb. 1999, pp. 90–100.

[16] S. Eichler and C. Roman, "Challenges of secure routing in MANETs: A simulative approach using AODV-SEC," in *Proc. IEEE Int. Conf. on Mobile Adhoc and Sensor Systems (MASS)*, C. Roman, Ed., 2006, pp. 481–484.

[17] J. Yin and S. Madria, "SecRout: A secure routing protocol for sensor networks," in *Proc. 20th Int. Conf. on Advanced Information Networking and Applications AINA 2006*, vol. 1, 18–20 April 2006, p. 6pp.

[18] Z. Liu, S. Lu, and J. Yan, "Secure routing protocol based trust for ad hoc networks," in *Proc. 8th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing SNPD 2007*, vol. 1, July 30 2007–Aug. 1 2007, pp. 279–283.

[19] Q. Ren and Q. Liang, "Secure media access control (MAC) in wireless sensor networks: Intrusion detections and countermeasures," in *Proc. 15th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2004*, vol. 4, 5–8 Sept. 2004, pp. 3025–3029.

[20] ——, "Fuzzy logic-optimized secure media access control (FSMAC) protocol wireless sensor networks," in *Proc. IEEE Int. Conf. on Computational Intelligence for Homeland Security and Personal Safety CIHSPS 2005*, March 31 2005–April 1 2005, pp. 37–43.

[21] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," in *Proc. IEEE Aerospace Conf.*, P. Volgyesi, Ed., vol. 3, 2003, pp. 1339– 1346.

[22] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[23] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[24] D. Chaturvedi, O. Malik, and P. Kalra, "Experimental studies with a generalized neuron based power system stabilizer," *IEEE Trans. Power Syst.*, vol. 19, pp. 1445–1453, 2004.

[25] V. Gudise and G. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proc. IEEE Swarm Intelligence Symposium SIS '03*, 2003, pp. 110–117.

[26] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.