MAConf: Passive Layer-2 Detection Mechanism for Mobile Wireless Sensor Networks

Faisal Hamady Nadia Bisher Ayman Kayssi Cesar Ghali
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
{fyh07, nbb04, ayman, csg04}@aub.edu.lb

Abstract—**Wireless sensor networks (WSNs) are being widely used in sensing, collecting, and disseminating data for various applications in environmental, industrial, and military applications. In this paper we present MAConf, a layer-2 detection mechanism for mobile WSNs. We propose the use of radio channel analysis to predict the underlying MAC protocol and addressing scheme, facilitating interconnections among WSN while ensuring an energy-efficient detection strategy. By implementing the intelligence using a neural network, we show that a 99% hit rate can be achieved in a variety of scenarios.**

Keywords: Wireless Sensor Network, Auto-Configuration, Energy Efficiency, Radio Analysis, Neural Networks, MAConf.

I. INTRODUCTION

Wireless Sensor Networks (WSN) have recently come into prominence because they hold the potential to revolutionize many aspects of our economy and life, providing a vehicle for enhancing various forms of productivity [1]. High dependency on such platforms, such as smart environments, requires extreme reliability and availability of service, as well as ease of usage and minimal user interaction.

Auto-configuration of nodes, at various layers of the OSI model [2], in a WSN environment is a desired solution due to the nature of deployment, type of operation, and autonomy of the sensor networks.

Desiring no modification to the existing protocols, and minimal radio communication we would like to detect and follow a certain layer-2 protocol upon mobility of the nodes. The convergence of the WSN requires fast and efficient layer-2 auto-configuration and adaptation techniques. One application is to adapt to new environment settings due to change in position [3], such as migrating a set of nodes from one geographic location to another as in the case of various military applications or even wearable sensors. Nevertheless, jamming attacks may target a sensor's ability to transmit or receive data, thus introducing severe effects on critical decisions and instantaneous responses. To prevent such kinds of denial-of-service attacks, one technique proposes to periodically or randomly change the used layer-2 protocol [4]. This introduces the need to redistribute the new utilized protocol to all the other nodes in the network. The detection of the MAC protocol is the first step towards building an interconnected network in a mobile heterogeneous environment.

In this paper, we present MAConf, a layer-2 detection mechanism for mobile WSNs. We propose the use of radio channel analysis to predict the underlying Medium-Access Control (MAC) protocol facilitating the energy-efficient detection strategy. We examine the ability of a node to predict the channel status and behavior in a way that increases the effectiveness of the communication link. We utilize a neural network's intelligence [18] to perform radio channel analysis, whereby the traffic samples sensed in the current environment are matched to pre-known patterns of various MAC protocols. All nodes then, can adjust their behavior as needed, keeping energy (the most critical resource in WSNs) consumption at a minimum. An important aspect in going behind a passive scheme is minimizing the energy consumption of the communication subsystem which is usually much more than that of computation subsystem. It is shown that the transmission of a bit needs an amount of energy that would be good for running a few thousands of instructions [24].

Although neural networks have been extensively used in WSN applications such as clustering and node selection [25-27], or data aggregation and prediction [28-31], this is the first research to introduce artificial intelligence in the context of auto-configuration at layer-2.

By simulating multiple scenarios where the parameters are changed accordingly, we establish a reference model that can be used to link the traffic analyzed under real WSN settings with the pre-obtained patterns. These simulation parameters include the MAC protocol and the traffic load as well as the node density within a certain region.

The rest of this paper is organized as follows: Section 2 presents a literature review of relevant research. In Section 3, a description of the system architecture and the protocol design are provided.

Section 4 analyzes and optimizes the obtained results. Finally, conclusions are given in Section 5.

II. RELATED WORK

A. MAC Protocols for WSNs

With an aim of providing an efficient means of predicting the layer-2 protocol used in a WSN, it is crucial to understand the underlying MAC protocol in a given setup in order to decide on the optimum communication link parameters and prevent energy losses due to needless transmissions.

The MAC protocol regulates access to the shared medium attempting to fulfill the communication performance requirements. The protocol defines what actions are performed during communication and what format this communication will follow. In general, MAC protocols can be categorized into three classes [5] depending on how time on the medium is partitioned:

- *Random*: A random MAC does not require synchronization or mutual time keeping since wake and sleep periods occur at random intervals, as illustrated in Figure 1. The disadvantages of such protocols are the energy waste due to node activation when no transmission is occurring, and collisions between transmissions when multiple active nodes are transmitting simultaneously. The random access protocols differ in how they signal when data is ready to be sent. Well-known examples of such protocols are BMAC [6] and XMAC [7].

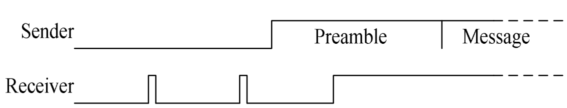

Figure 1. BMAC node status

- *Slots*: In slotted protocols, nodes wake up and sleep simultaneously and contend for the medium for a short period of time. This requires synchronization among all the network nodes. Usually the nodes agree on a common sleep/wake-up pattern and the channel access is based on contention, similar to random access protocols (see Figure 2.) Therefore, the chance for a collision becomes much higher as all traffic is grouped into a much smaller time interval. The prime example of these protocols is the SMAC protocol [8].

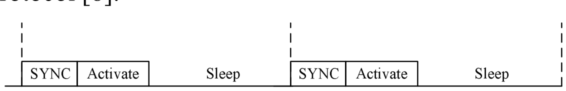

Figure 2. Sleep and wake periods in SMAC

- *Frames*: The frame-based protocols make use of partitioned frames to control the medium access by network nodes. Each frame is split into slots and a scheduling algorithm is employed to organize when nodes are allowed to transmit in order to eliminate possible collisions. This principle is also known as Time Division Multiple Access. Since all the slots are non-overlapping, no collisions occur and overhearing is greatly reduced (see Figure 3.) On the other hand, a synchronization protocol must be implemented to partition the slots and a distributed schedule must be built. Timekeeping is very important in frame-based protocols, hence an accurate clock must be present to maintain synchronization between all nodes. Both LMAC [9] and TRAMA [23] are classified in this category.

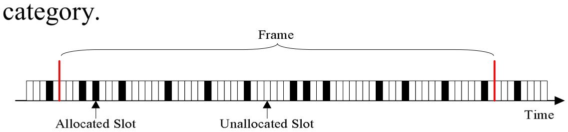

Figure 3. Time slots in TDMA-based protocols

Hybrid protocols that combine features from two or more protocol classes do exist. Few MAC protocols with good power efficiency only handle a small range of traffic loads and applications. This can be a problem when a protocol needs to handle a wide range of traffic loads and characteristics. There are some protocols that can adjust intervals to adapt to various traffic loads. ZMAC [10] and PMAC [11] are examples of such adaptive protocols.

B. Auto-Configuration

Auto-configuration and adaptation are challenging issues in WSNs in very dynamic and changing environments. Reducing energy consumption, increasing the lifetime of the sensors in the network, and minimizing the use of the communication medium are vital design objectives of auto-configuration techniques. Several papers in the literature address auto-configuration of protocols at different layers of the OSI model. Cha et al. [12] propose a new protocol for auto-configuring the application layer of WSN nodes for the 4M Group Management system.

Weniger [13] proposes an approach to auto-configure the network layer address in Mobile Ad hoc Networks (MANETs) by exploiting the received routing information packets to define the address space of the current network. The solution was designed to be passive in order to reduce the consumed energy.

Ros et al. designed EMAP, an extensible MANET auto-configuration protocol [14]. The protocol is used to auto-configure a MANET node with a layer 3 Internet Protocol address by selecting an address and consulting the rest of the network for its uniqueness. This is an active energy consuming approach and might introduce latency in the auto-configuration process since a reply from the network nodes needs to be received in order to assess the uniqueness of the assigned address.

ACAS is a layer-3 auto-configuration protocol proposed by Du et al. in [15]. The protocol is based on clustering the network area and assigning an address to each cluster. Each node then auto-configures its layer-3 address based on its position. This approach requires a positioning algorithm to be involved which may impose the existence of reference nodes with which communications might not be available all the time.

All the previous protocols propose solutions for the auto-configuration problem at the network or application layers. They assume that the layer-2 protocol and its associated parameters are known and properly configured.

On the other hand, Krishnamurthy et al. propose in [16,17] a two-phase technique to auto-configure the communication and layer-2 parameters set in Cognitive Radio (CR) networks. The first phase consists of communicating with neighbor nodes in order to determine all possible common channels, while the second phase computes the global common channel parameters. However, transmission of packets among the nodes in the network turns the solution to be active and energy-consuming. Moreover, several assumptions are made during the auto-configuration process such as low mobility of all nodes and clock synchronization using a positioning system such as GPS.

III. PROTOCOL DESIGN AND SIMULATION

The radio activity of the nodes, how they send and receive packets, is highly dependent on the techniques used to allocate the channel. In scenarios where the configuration of the wireless channel is continuously changing, automatically predicting the utilized layer-2 protocol can reduce the energy consumption of nodes that are trying to guess the used MAC by brute-forcing all possible options.

The sleep/wake up periods are distinctive parameters in a WSN. They may provide a unique signature of the underlying MAC protocol. An intelligent system, based on a neural network approach, can be trained in an *offline manner* with a large set of pre-obtained traffic patterns categorized by the layer-2 protocol to develop an association between the observed channel status and the MAC protocol. These patterns must consider variable traffic loads as well as multiple layouts of sensors. The intelligent system would then be utilized by a node as a pattern recognition tool, where the input is a *Boolean vector* representing the channel status over time, and the output is an intelligent guess of the MAC protocol used in the WSN.

The implementation of the proposed solution consists of three phases. First, a large set of traffic patterns (each pattern in our case is a vector of 1024 bits representing the channel samples) is collected under different network scenarios. The set is then used to train a neural network in order to obtain an intelligent system that would identify the MAC protocol. Finally, random vectors are fed to the neural network for the purpose of validation.

It is worth mentioning that using a single hidden layer along with few neurons (here we used 19 neurons) is an important factor to consider upon designing NNs targeting memory constrained devices such as sensors. Also, training the NN in an offline manner and then porting the prediction model to the target device is a key element in mitigating the processing overhead of the training from the sensor node. The need to predict a MAC protocol in an online setting can then be feasible, given that the whole operation would involve a normalizing function following few multiply and add operations that can be easily executed by today's sensors.

A. Simulation Environment

For the sake of gathering a large set of channel vectors under various MAC protocols, we chose Objective Modular Network Test-bed in C++ (OMNET++) [19] as a basic framework that reflects accurate scenarios. Castalia [20], which supports both SMAC and BMAC, and MACSimulator [21], which supports LMAC, were introduced on top of OMNET++ to provide a full simulation environment for the WSNs. To obtain realistic data sets, the number of nodes and the traffic loads were varied in different runs for each MAC protocol used. These parameters are listed in Table 1.

Table 1. WSN Simulation Parameters

Parameter	**Variation**
Nodes within cluster	7, 9, 11,13 and 15
MAC Protocol	SMAC, LMAC, and BMAC
Traffic Load	Idle, low, and moderate

Upon varying these parameters and running the different scenarios of the simulation with random seeds, a total of 30,000 vectors, 10,000 per each of the MAC protocols SMAC, LMAC, and BMAC, were obtained. These vectors represent the channel status over time.

A sample of the channel status in idle networks operating under SMAC and LMAC are shown in Figure 5 and Figure 6, respectively. In BMAC, where no control packets are sent in idle mode, the channel is always free when the network is idle.

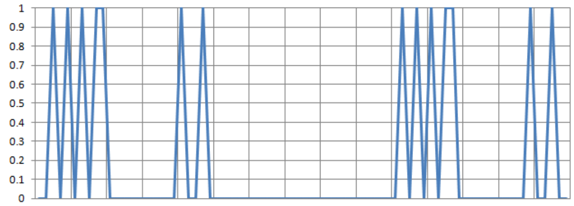

Figure 5. Channel status over time under SMAC (idle network)

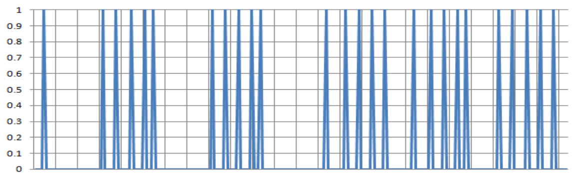

Figure 6. Channel status over time under LMAC (idle network)

B. Neural Network Design

A neural network (NN) is a collection of extremely interrelated processing elements that classifies a set of inputs into a set of targets. The processing elements can be customized by varying their weights and biases as well as their connections to match the input data characteristics in a process known as training. It is crucial to understand that the purpose of the training algorithm is to build a model that can achieve high hit rates coupled with good generalization capabilities. If excessively trained in terms of time and input data, the system would "memorize" the patterns rather than extracting regularities, turning it unfit for the intended use.

When the neural network is trained with enough data, it can provide a probabilistic estimate of the matching between the input and the patterns it was trained to recognize. The configuration of the system is refined until an adequate level of response is reached.

The training of the neural network implemented in the proposed MAConf was conducted using the neural network modules of MATLAB [22]. A batch, back-propagation algorithm was applied for 15,000 samples of channel vectors. From all the records which were preprocessed for use in the prototype, about 1500 were randomly selected for testing and the remaining were used to train and validate the system. In an effort to produce the most accurate results, the following steps were repeatedly performed until a satisfactory network response is observed.

- Increase the number of hidden neurons in the NN
- Increase the number of training vectors
- Reset the initial network weights and biases then re-train with the channel vectors.

After the completion of training and testing of the neural network, the various connection weights were frozen and the network was interrogated.

IV. Results and Analysis

At the end of the training phase of the NN, the performance metrics are calculated. They are summarized in Figure 7. A mean square error (MSE) of 0.1223 may not be desired in this application, so we will propose later in this section an enhancement that will effectively enhance the detection methodology.

Results

	Samples	MSE	%E
Training:	12000	1.16282e-1	22.03333e-0
Validation:	1500	1.26737e-1	25.60000e-0
Testing:	1500	1.22351e-1	23.73333e-0

Figure 7. Error Percentages of the NN

A plot of the training errors, validation errors, and test errors is shown in Figure 8. The best validation performance occurred at iteration 224, and the network at this iteration is returned. The MSE of the network starts at a large value and decreases to a smaller value demonstrating the learning process of new characteristics form the training set.

The plot has three lines, because the 15,000 input and targets vectors are randomly divided into three sets. Of the 15,000 vectors, 80% of the vectors were used to train the network and 10% of the vectors were used to validate how well the network generalized. Training continues as long as the training reduces the network's error on the validation vectors. After the network memorizes the training set, at the expense of generalizing more poorly, training is stopped. This technique automatically avoids the problem of over-fitting, which plagues many optimization and learning algorithms. Finally, the remaining 10% of the vectors provide an independent test of network generalization to data that the network has never experienced.

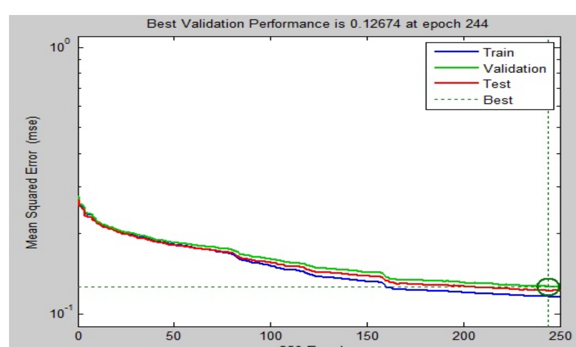

Figure 8. Training, Validation, and Test Errors

Figure 9 shows the confusion matrices for training, testing, and validation, and the three kinds of data combined (at the bottom right matrix.) SMAC, LMAC and BMAC are marked as classes 1, 2, and 3, respectively. The diagonal cells in each table show the number of cases that were correctly classified, and the off-diagonal cells show the misclassified cases. The blue cell in the bottom right corner of each matrix shows in green text the total percent of correctly classified cases and in red the total percent of misclassified cases. The results for all three data sets, training, validation, and testing, shows a hit rate of around 78%. We notice that the misclassified cases result from ambiguity between SMAC and BMAC.

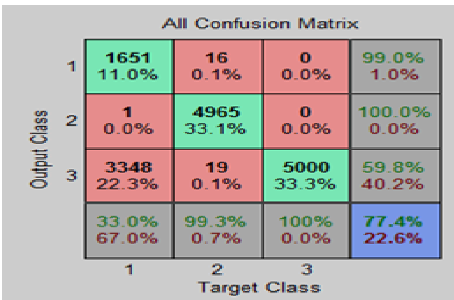

Figure 9. Confusion matrix for 3 Classes

The receiver operating characteristic (ROC) curve graphically describes the performance of the classifier. ROC curves shown in Figure 10 are plots of the true positive rate, sensitivity, versus the false positive rate, 1 – specificity, as the threshold is varied. The colored lines in each axis represent the ROC curves for all categories of this test problem: training, validation and testing. For this detection problem, the network performance is unsatisfactory since we would like the curves to be in the upper-left corner indicating a high detection rate with a low false positive rate.

Targeting a higher hit rate and thus a more accurate identification of the layer-2 protocol requires an improvement of the prediction model used.

As observed previously, the confusion in the NN is mainly between SMAC and BMAC. We will make use of two facts that would help in uniquely differentiating them:

- An idle network under BMAC never utilizes the channel medium
- In SMAC, control packets are always observed within a period of 930 ms (minimum duty cycle = 10%)

Now the identification process consists of two steps. First, the channel is monitored for a relatively long duration (around 10 seconds.) If we observe a free channel spanning the duration of a full period, we can be certain that BMAC is used. If packets are continuously being exchanged, we record a channel vector and feed it into the neural network that will now classify the MAC as SMAC or LMAC.

Figure 11 clearly illustrates the accurate detection (99.5% in this case) upon limiting the input to SMAC and LMAC vectors. This proves that a proper classification scheme can be adopted provided that the sensor network operates in an idle mode for a considerable amount of time.

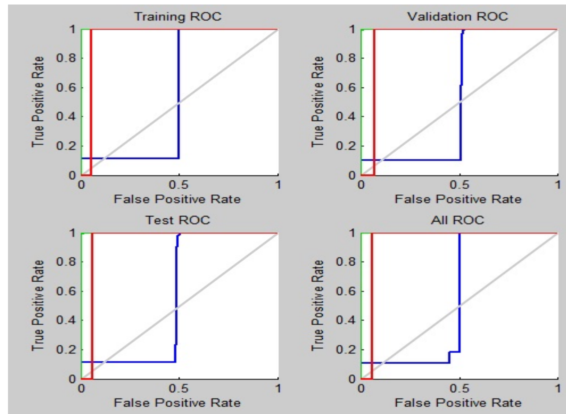

Figure 10. ROC curves

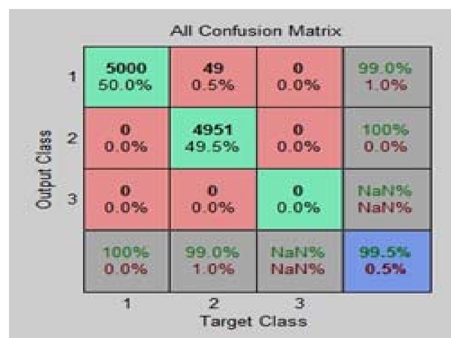

Figure 11. Confusion matrix for two classes

The nature of sensor networks and the activity patterns in the vast majority of their applications enable us to assume that the network would be idle most of the time. This ensures that a proper assessment can be made at the first step: differentiating BMAC from SMAC and LMAC. The modified version of the NN thus shows optimal results and can used as a reliable decision engine.

V. CONCLUSIONS

In this paper, we presented an overview of MAC protocols and auto-configuration schemes used in Wireless Sensor Networks. We proposed a new passive and energy-efficient technique named MAConf, to predict the layer-2 protocol used in the network. The implemented scheme employs an optimized neural network to identify the MAC protocol after sampling the channel for a small period of time. Our results demonstrate that we can achieve a correct prediction of the MAC protocol with minimal false positives.

While this prototype was not designed to be a complete MAC detection system, hit rates higher than 99% clearly demonstrate the potential of a neural network to detect particular protocols given the knowledge of radio channel status.

As future work, more studies are required to define candidate attributes (such as packet inter-arrival periods) that can be fed into the system to better classify heterogeneous MAC protocols, satisfying the real time requirements of certain applications. Multi-layer NN may also be considered to come up with a one-step prediction model, supporting a wider range of protocols. It would be useful then to implement this solution in a real WSN to verify the results obtained by simulators and to assess the impact of such intelligent models on the performance and power consumption of the mobile sensor nodes.

REFERENCES

[1] D. Culler, D. Estrin and M. Srivastava, "Overview of sensor networks," *Computer,* vol. 37, pp. 41-49, 2004.

[2] D. Wetteroth, *OSI Reference Model for Telecommunications,* McGraw-Hill Professional, 2001.

[3] L. C. Zhong, J. Rabaey, C. Guo and R. Shah, "Data link layer design for wireless sensor networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE,* 2002.

[4] W. Xu, W. Trappe, Y. Zhang and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing,* 2005, pp. 46-57.

[5] K. Langendoen, "Medium access control in wireless sensor networks", *Medium Access Control in Wireless Networks,* vol. 2, pp. 535–560, 2008.

[6] J. Polastre, J. Hill and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, November,* 2004.

[7] M. Buettner, G. V. Yee, E. Anderson and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems,* 2006, pp. 320.

[8] W. Ye and J. Heidemann, "Medium access control in wireless sensor networks," *Wireless Sensor Networks,* pp. 73-91, 2004.

[9] L. Van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks," in *1st Int. Workshop on Networked Sensing Systems (INSS 2004),* 2004.

[10] I. Rhee, A. Warrier, M. Aia, J. Min and M. L. Sichitiu, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking (TON),* vol. 16, pp. 511-524, 2008.

[11] T. Zheng, S. Radhakrishnan and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," in *19th IEEE International Parallel and Distributed Processing Symposium, 2005. Proceedings,* 2005, pp. 8.

[12] S. K. Cha, J. H. Lee, J. Y. Song, D. H. Kim and G. B. Lee, "An auto-configuration of 4M group management using wireless sensor networks for reconfigurable manufacturing system," in *Assembly and Manufacturing, 2009. ISAM 2009.*

[13] Weniger, "PACMAN: Passive autoconfiguration for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on,* vol. 23, pp. 507-519, 2005.

[14] F. Ros et al., "*Extensible MANET Auto-configuration Protocol (EMAP)"*, Internet Draft, draft-ros-autoconf-emap-02, Mar. 2006.

[15] Z. Du, D. Qian, S. Stanczak, R. Heras and Y. Liu, "Auto-configuration of shared network-layer address in cluster-based wireless sensor network," in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on,* 2008, pp. 148-153.

[16] S. Krishnamurthy, R. Chandrasekaran, N. Mittal and S. Venkatesan, "Brief announcement: Synchronous distributed algorithms for node discovery and configuration in multi-channel cognitive radio networks," *Distributed Computing,* pp. 572-574, 2006.

[17] S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mittal, S. Venkatesan and R. Prakash, "Time-efficient distributed layer-2 auto-configuration for cognitive radio networks," *Computer Networks,* vol. 52, pp. 831-849, 2008.

[18] M. T. Hagan, H. B. Demuth and M. H. Beale, *Neural Network Design.* PWS Boston, MA, 1996.

[19] A. Varga and R. Hornig, "An overview of the OMNeT simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops,* 2008, pp. 60.

[20] A. Boulis, "Castalia: Revealing pitfalls in designing distributed algorithms in WSN," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems,* 2007, pp. 408.

[21] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. Haneveld, T. Parker, O. Visser, H. Lichte and S. Valentin, "Simulating wireless and mobile networks in OMNeT the MiXiM vision," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops,* 2008.

[22] H. B. Demuth and M. Beale, *Neural Network Toolbox: For use with MATLAB.* Citeseer, 2000.

[23] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Networks,* vol. 12, no. 1, pp. 63-78, 2006.

[24] Anastasi, G., Conti, M., Francesco, D.M, Passarella. A, "Energy Conservation in Wireless Sensor Networks: a Survey",2009, Ad Hoc Networks, Vol.7, No.3, pp.537—568

[25] Aslam N, Philips W, Robertson W, Siva Kumar SH, "A multi-criterion optimization technique for energy efficient cluster formation in Wireless Sensor networks",2010, Information Fusion, Elsevier

[26] Dehni L, Krief F, Bennani Y., "Power Control and Clustering in Wireless Sensor Networks", Challenges in Ad Hoc Networking, vol 2005, pp.31-40.

[27] Cordina M, Debono C.J. "Increasing Wireless Sensor Network Lifetime through the Application of SOM neural networks." ISCCSP, IEEE, Malta, 2008, pp. 467-471

[28] Oldewurtel, Frank and Mahonen, Petri, "Neural Wireless Sensor Networks", International Conference on Systems and Networks Communications, ICSNC '06

[29] Smith, D., Singh, S. "Approaches to Multisensor Data Fusion in Target Tracking: A Survey", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2006, Vol. 18

[30] M. Winter and G. Favier, "A Neural Network for Data Association", Proc. 1999 IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, vol. 2, pp. 1041-1044

[31] Mei, L., Haihao, L., Shen, Y., Fan, J., Huang, S.H., "Elastic neural network method for multi-target tracking task allocation in wireless sensor network", 2009, Computers and Mathematics with Applications 57(11-12), pp.1822–1828