Modified TCP Protocol for Wireless Sensor Networks

Hesham Abusaimeh[1], Mohammad Shkoukani[2]

[1] **Computer Networks Systems,**
Faculty of Information Technology,
Applied Science University
Amman, 11931, Jordan

[2] **Computer Information Systems,**
Faculty of Information Technology,
Applied Science University
Amman, 11931, Jordan

Abstract

The transport layer's protocols have been used in the Wireless Sensor Networks (WSNs) in order to achieve data reliability. However, the limitations of the WSNs did not usually considered when implementing these protocols in the stack of the sensor nodes. In this paper, the TCP protocol has been modified in order to use it with the unique characteristics of the WSNs. The TCP is connection-oriented protocol, which may cause extra overhead, but it will reduce the number of redundant packets. This will also cause increasing the lifetime. In addition, that applying the TCP protocol in the WSN will make it reachable from the traditional wire or wireless network without gateway or connecter. Our proposed stack model has achieved better simulation result that the traditional sensor node stacks in term of throughput, packets loss ratio, and network lifetime.

Keywords: *TCP, Wireless Sensor Networks, Transport Layer, ZigBee Stack, Network Lifetime, Network Throughput.*

1. Introduction

Wireless Sensor Networks (WSNs) are composed of low cost, low power, multifunctional sensor nodes that are small in size and communicate wirelessly over short distances. The sensor nodes collaborate to sense and collate information about their environment, through a set of transducers and a radio receiver, and to forward information towards a central sink node [1]. Moreover, information from the sensor node is accessed through a gateway sensor node as depicted in Figure 1, which provides a point of ingress between the WSN and either a direct link to a relatively resource rich computer or indirectly through a LAN to a resource rich computer. Therefore, WSN cannot be separated from the traditional networks. WSNS have been developed to serve many applications used in the human life. These applications are implemented to simplify the human life such as military, environment, health, home and other commercial applications [2].

As shown in figure 1, the WSN is usually connected to gateway or connector in order to send the required data to the external wire or wireless network. The external network control the queries sent to the sensor nodes or process the data received from the sensor nodes. The main reason for that gateway is the difference in the protocol stack that is used in the WSN.

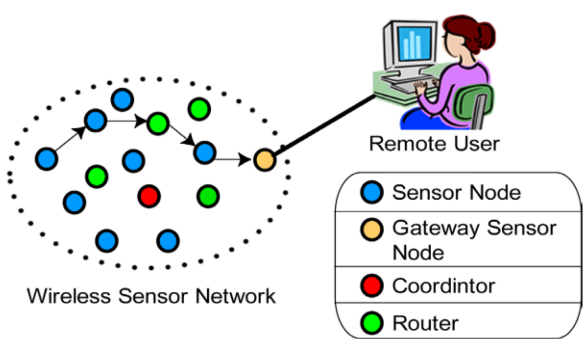

Fig. 1 Simple WSN linked to Base station through a Gateway.

Most of the manufactured sensor nodes is based on the IEEE 802.15.4, and the ZigBee stack, The protocol stack that is used in these two standards is mainly including the four main layers the physical (PHY), Medium Access Control (MAC), Network, and the Applications Layers as shown in Figure 1. Obviously, the Transport Layer is not included in this stack. The overheads of the transport layer addressing, large headers and energy consumption are the main reason of excluding this layer from the stack of these untraditional networks [1].

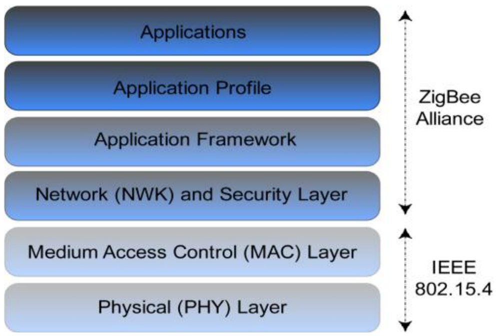

Fig. 2: IEEE 802.15.4 and ZigBee stack used in the WSN

Many researches have been accomplished in order to implement the Transmission Control Protocol in the transport layer of the WSN. Few of these researches suggested modifying the TCP protocol to fit the constraints of the WSN. In this paper, we are going to suggest a modification over the TCP protocol in order to suite the WSN characteristics.

The reminder of this paper is going as the following: In section 2, an overview about the TCP protocol is explained. Section 3, the proposed modification TCP protocol is discussed to adequate the WSN and the proposed model of the new sensor stack in section 4. The simulation result and discussion is studied in section 5. Finally, the conclusion of the paper is written in section 6.

2. TCP Overview

The TCP protocol is a transport layer protocol, which responsible about sending and receiving data based on the destination and source addresses provided from the network layer protocol, which is usually the Internet Protocol (IP) in the traditional network. The TCP protocol is also considered as connection-oriented protocol, which means that it delivers the data in three phases: establishing the connection between the source and the destination, send the data and receiving the delivery acknowledgment, and closing the connection after that [3].
Using the TCP protocol in the sensor nodes may link the WSN to any other network without a gateway, however implementing the original TCP protocol have many drawbacks as discussed in the following sub section.

2.1 TCP problems in WSN

1. Establishing the connection in the TCP is based on three-way (Request to Send RTS- Clear to Send CTS) handshake process that is described in Figure 3. This three-way handshake process causes extra overhead in the WSN especially for the small amount of data transferred in the WSN.

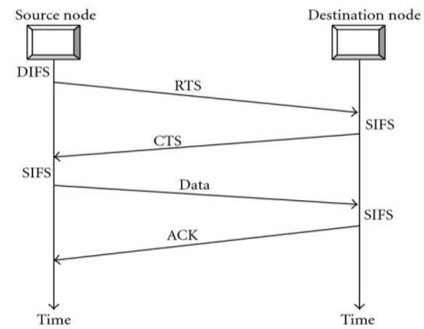

Fig. 3: three-way handshake connection-oriented process

2. WSN is used in real time application such as fire alarm system or monitoring factories, while establishing TCP connection may finish the data usability.
3. As mentioned earlier the TCP protocol is based on the IP address. In WSN, it is not possible to do manual addressing for the large scale of the sensor nodes. In addition, it will cost additional overhead to distribute the addresses automatically as in the DHCP.
4. The TCP also adds large header size in comparison with the small size of data transmitted from the source to the destination nodes.
5. The implementation of the TCP protocol is too heavy on the small memory size of the sensor nodes [4].
6. The continuous byte streaming of the TCP protocol is considered as the main cause of consuming energy in the wireless networks.
7. TCP is based on the acknowledgment (ACK) report from the destination node; otherwise, the packet will be retransmitted again to the destination node. Retransmission means more wasted energy will be consumed [5].

3. Proposed TCP Modifications

Our proposed modification over the TCP protocol is to make it suitable for the WSNs. The benefit of using the TCP in the WSN not only link it directly to the other networks without gateways but also reduce the redundant packet that can be sent from a source node to destination node. As obvious in Figure 4, the modified TCP protocol should be implemented between the application layer and the network layer of the ZigBee stack.

The modifications that are suggested over the TCP protocol as the following:

1. The congestion control in the TCP protocol is not required in the WSN, because the WSN routing protocol such as (Ad hoc On-demand Distance Vector) AODV can specify the bandwidth delay in advance while establishing the route between the source and the destination nodes. This routing protocol is already implemented in the network layer of the ZigBee stack.

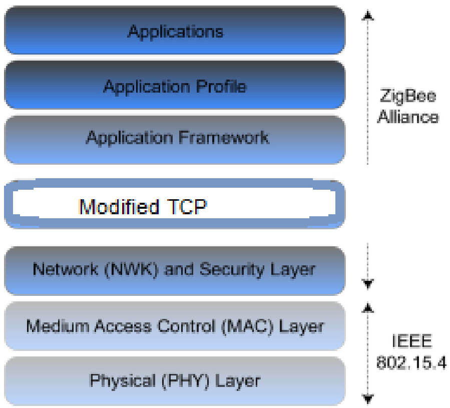

Fig. 4: IEEE 802.15.4 and ZigBee stack used in the WSN

2. Making the ACK packet wait until receiving wide range of data packet then send it back to the source node. This will make the TCP protocol less overhead in the WSN. However, it is important to make the ACK only wait other kind of packets, because receiving the same packet frequently without sending the ACK packet will increase the overhead of the TCP protocol.
   In order to let the destination knowing that there are still data from the source node not arrived before sending the ACK. We have developed a technique to increase the Window field in the last packet with small number SNO in the header of the modified TCP as in the Equation 1. Therefore, the destination node will know that this is the last packet from this source node. Then the destination should send the ACK packet with this SNo in the header of it in order to let the source accept this packet and does not retransmit the data again.

$$W(N) = InitValue + SNO \qquad (1)$$

   Where, N is the number of hopes between the source and the destination. InitValue, is the initial value of the window field, regularly it should be the bandwidth delay between the source node and the destination node. In addition, to the small number SNO which can be defined at the source node or it can be generated randomly.

3. The TCP fragmentation of the packets and giving each fragment a sequence number in order to let the data arrives in sequence order is not feasible and can be discarded in the WSN because of the small amount of data.
4. Rather than the IP address we will use the node number that is provided by the coordinator when establishing the network to act as the address of the node. This address will be used by the modified TCP protocol in order to determine the source node and the destination node in the header of the protocol. In addition, the AODV routing protocol will used this address to find the feasible route to the destination node when needed.
5. Reducing the header size by one of the [6, 7] compression techniques. This will release the scary consuming more energy by transmitting header larger than the sensed data packet.
6. We have used the [4] TCP implementation to run the TCP over small flash memory of the sensor nodes.

4. The Proposed Model of the Sensor Nodes Stack

After modifying the TCP Protocol, now, we need to setup the proposed stack that will be used in our simulation environment.

Firstly, the application layer is on the top of the stack, and primarily deals with the processing of sensed information, encryption, formatting and storage of data. Moreover, the application layer examines the underlying layers to detect if sufficient network resources and services are available to meet the user’s network requests.

Secondly, the Modified TCP Layer that acts as the transport layer of the stack, which is responsible about connecting the devices and sending or retransmitting the sensed data.

Thirdly, the Network Layer in the WSN ZigBee stack includes the Ad hoc On-demand Distance Vector (AODV) and the Cluster-Tree routing protocols. The cluster-tree is used to establishing the network and combining the sensor nodes in clusters while the AODV protocol is used to discover the routes between the nodes when required and then send the data received from the modified TCP layer.

Fourthly, the ZigBee stack is based on the IEEE 802.15.4 MAC layer to define the network structure of the WSNs. In addition, the MAC layer supports super-frame and non super-frame communications. In super-frame communications, the network coordinator emits a synchronizing beacon to indicate the start of a super-frame and a beacon to indicate the end of the super-frame. The super-frame is divided into active times and inactive times where network nodes sleep to conserve energy. The active period is further divided into contention access period (CAP) and contention free period (CFP). During the CAP, nodes compete for access to the communication channel using the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) protocol. During the CFP the network coordinator splits the available CFP into guaranteed time slots (GTS) and assigns each slot to a particular device to communicate without contention from any other device on the network. Alternatively, in the non super-frame based approach all devices on the WSN compete for access to the channel using CSMA-CA. In this case, all devices have an equal chance of gaining access to the channel or not gaining access [8].

Finally, The PHY layer is responsible for characterizing the physical attributes and behaviors of the sensor nodes. This includes turning the transceiver on and off, selecting the appropriate channel, estimating the link quality (LQI), receiver energy detection, and clear channel assessment (CCA) [1]. Moreover, the PHY layer supports three licence free ISM (Industrial, Scientific, and Medical) frequency bands including 2.4 GHz with 16 channels and a 250 kbps data rate, 902 to 928 MHz with 10 channels and a 40 kbps data rate and, 868 to 870 MHz with 1 channel and a 20 kbps data rate [8, 9, 10].

5. Simulation Results and Discussion

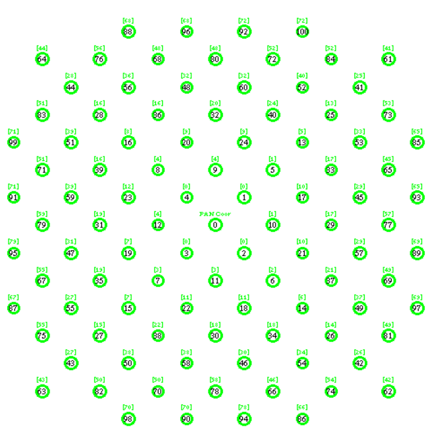

Fig. 5: Static distribution of the sensor nodes in the simulation model

The simulation of the modified TCP protocol and the proposed stack model has been accomplished via NS-2 simulator [11]. The simulator has been used to validate the performance of the WSNs with the modified TCP. The simulation is using static sensor nodes distribution as shown in the following Figure 5.
The simulation model lies on 100 wireless sensor nodes distributed in a grid area of size 100x100 meters with the TwoRayGround radio propagation model. The range of the wireless signal transmission is 30 meters. The coordinator, which is node number zero in the center of the network, will establish the network and distribute the addresses among the nodes.
In order to load the network, A CBR traffic flow is generated between the source node S to coordinator C. the packet size of this traffic will 80 bytes in addition to the modified TCP header size. This packet will be retransmitted from the border nodes Bs to the Coordinator each 20 milliseconds until each node receive its ACK packet. Each simulation runs for 5 minutes as shown in the following Figure 6.

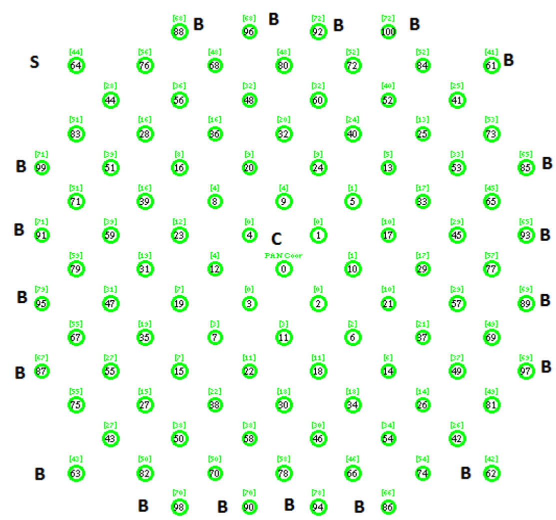

Fig. 6: the coordinator, the source and the border nodes

Furthermore, the simulation based on the default NS-2 energy model in order to judge the efficiency of the modified TCP in term of the nodes power, which is the most concern in the WSNs. The energy model includes three initial values: the initial energy of the nodes (InitEng), the transmission power (txPower), which is used up transmitting each packet, and the reception power (rxPower), which is consumed on receiving each packet. The output of this model is the total remaining energy, packets transmission energy, and the reception energy of each node. This NS-2 energy model decreases the current energy of the node, when the node transmits and receives packets by applying the following equations as mentioned in [11].

$$Node\_Eng = Node\_Eng - (txPower * txTime) \quad (2)$$

$$Node\_Eng = Node\_Eng - (rxPower * rxTime) \quad (3)$$

Where, Node_Eng is the node energy, and initially is equal to the initial energy of the node, txTime is the time required to transmit each packet, and rxTime is the time required to receive each packet. In this simulation, the initial full energy is equal to 4.0 joules. The transmission power and the reception power are equal to 1.0 watt. Most of the simulation parameters are shown in the following Table (1).

Table 1: Simulation Setup Parameters

Parameter Name	***Parameter Value***
InitEng	4.0 Joules
rxPower	1.0 Watt
txPower	1.0 Watt
Packet Size	80 Bytes
Packet Interval	20 Milliseconds
Simulation Time	5 Minutes
Number of Nodes	100
Antenna Type	OmniAntenna
Radio Model	TwoRayGround
Grid Size	100 $meter^2$
Distance between neighbours	10 meter
Routing Protocol	AODV
MAC Protocol	MAC/802_15_4
System Loss (L_)	1.0
Gain Transmitter antenna (Gt)	1.0
Gain Reciever antenna (Gr)	1.0
Transmitter Hieght (Ht)	1.5meter
Reciever Hieght (Hr)	1.5meter

In this simulation, The AODV has been used as the routing protocol of the network layer. The IEEE 802.15.4 PHY and MAC layers have been implemented in this simulation as the specification of the physical and the MAC layer of the sensor node [12].

The first simulation result that we want to discuss first is the size of the modified TCP ACKs' packets. Because mainly this is the main addition to the traditional WSN flow size. In the following Figure 7, the capacity of the ACKs flow in the WSN is small in compared with the data flow. Only around 10 % of the network flow is the ACKs packets flow, and the size of the data flow is the main flow in the network, which is more that 90% of the total network flow. The total network flow in our simulation was around 14000 bytes of packets.

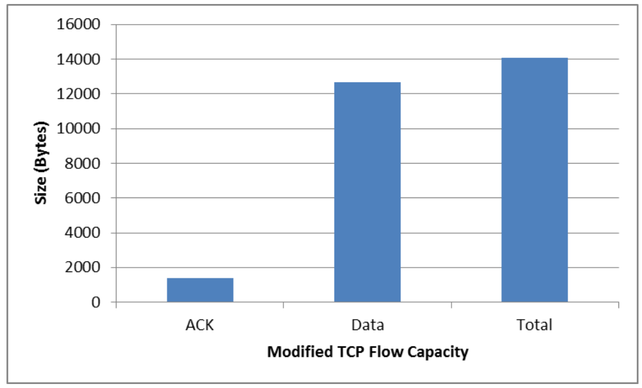

Fig. 7: The Modified TCP flow Capacity in the WSN

The Second simulation result will be discussed are the throughput of the sender nodes in the traditional WSN with the AODV routing protocol compared with the WSN with our Modified TCP over the AODV routing protocol. As shown in Figure 8, the throughput of the WSN after adding the modified TCP protocol above the routing protocol becomes better for all the source sensor nodes compared with the throughput of the same source nodes without the TCP protocol. The throughput for most of the sources sensor nodes in the modified TCP network was around 2000bytes/second.

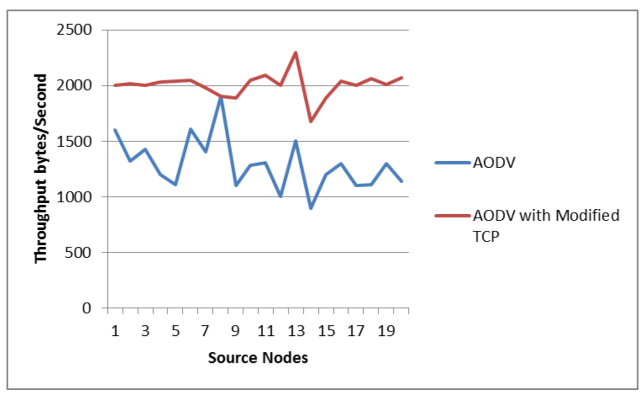

Fig. 8: The Throughput of the WSN before and after the Modified TCP protocol.

The packet loss ratio is the third simulation result, which calculated in our simulation. Figure 9 shows that the packet loss ratio in the WSN during the 5 minutes simulation time is much less in the new model with the modified TCP protocol over the AODV. While the packet loss ratio in the traditional WSN, is much bigger.

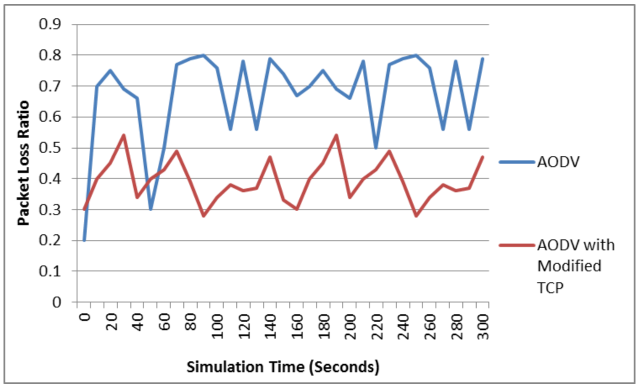

Fig. 9: The Packet Loss Ratio before and after the Modified TCP protocol.

The last simulation results we need to compare is the lifetime of the sensor node in the network during the simulation before and after adding the modified TCP protocol to the sensor stack. We have calculated the lifetime based on our previous energy model in [13]. Obviously the lifetime of the sensor nodes after implementing the modified TCP protocol in the sensors stacks is much better than the old sensor stack and its almost the double. Where the lifetime average in the old stack for the sensor nodes was around 134 seconds, the lifetime average for the sensor nodes in the new stack was around 230 seconds.

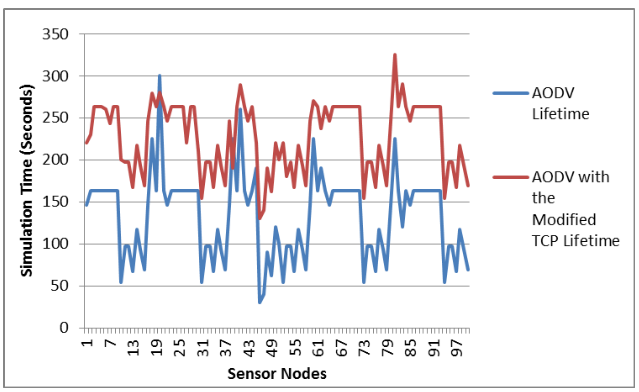

Fig. 10: The lifetime of the sensor nodes before and after the Modified TCP protocol.

Therefore, all the simulation results proved adding our proposed TCP protocol with the modification that have been accomplished on it to be suitable for the WSNs are much better in term of performance of throughput, packets loss ratio and energy consumption than the traditional WSN.

6. Conclusions

Both the ZigBee and the IEEE 802.15.4 standards do not have implementation for the transport layer protocols. Many researchers have worked on implementing transport layer protocols in the WSN stack especially the TCP protocol in order to manage the data reliability and confidentiality. Few of these researches consider the limitations of the WSN in term of memory, computing and power supply in implementing the TCP protocol in the stack layers.
The motivation of this paper is the modified TCP protocol proposed in this paper to fit the WSN has better performance in term of data reliability, network throughput, and packet delay in comparison with the traditional WSN without transport layer protocol.
The performance evaluation of our proposed modified TCP protocol was conduction by the NS-2 simulator. All the simulation result in term of network lifetime, network throughput, packets loss ratio, and network flow showed that the new stack with our modified TCP in the transport layer has better performance than the traditional WSN stack.

Acknowledgments

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to cover the publication fee of this research article.

References

[1] P. Barontib, P. Pillaia, V. Chooka, S. Chessab, A. Gottab, and Y.F. Hua,"Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards", Computer Communications, Vol. 30, No. 7, pp. 1655-1695, 2007.
[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, "A Survey on Sensor Networks", IEEE Communications Magazine, Vol. 40, No. 8, pp. 102- 114, 2002.
[3] W. Stalling, “Data and Computer Communications”,9th Edition, Pearson Education, 2011.
[4] A. Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller, “Connecting Wireless Sensornets with TCP/IP networks”, In WWWIC2004, February, 2004.
[5] A. Willing, and H. Karl, “Data Transport reliability in Wireless sensor networks-A Survey of issues and solution”, Praxis der informationsverarbeitung und kommunikation, Vol. 28, April 2005, pp.86-92.
[6] M. Degermark, B. Nordgren, and S. Pink, “IP header compression”. RFC 2507, Internet Engineering Task Force, February 1999.
[7] V. Jacobson. “Compressing TCP/IP headers for low-speed serial links”. RFC 1144, Internet Engineering Task Force, February 1990.
[8] ZigBee Alliance, ZigBee Specification, Zigbee standard organization, http://www.zigbee.org, 27th Jun 2005.

[9] W. Craig, "Zigbee: Wireless Control That Simply Works", http://www.zigbeeresourceguide.com/images/ZigBee_RG_2008.pdf, 2005.

[10] S.C. Ergen, 2004, Zigbee/IEEE802.15.4 Summary, September 2004.

[11] USC Information Sciences Institute, Marina del Rey, CA. Network Simulator - NS2, http://www.isi.edu/nsnam/ns

[12] IEEE Std 802.15.4-2003, 2003, IEEE Standard for Information technology —Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY), Specifications for Low- Rate Wireless Personal Area Networks (LR-WPANs), 2003

[13] H. Abusaimeh, and S. Yang, 2008, “Balancing the Power Consumption Speed in Flat and Hierarchical WSN”, International Journal of Automation and computing, V 5, N 4, pp 366-375, October 2008.

Hesham Abusaimeh received his B.Sc. degree from Applied Science University, Amman, Jordan in 2003, and M.Sc. degree from New York Institute of Technology in 2004, both in computer His Ph.D. degree in computer science in the field of wireless sensor networks communication and routing protocols from Loughborough University, UK in 2009. His research interests include Network and Controls, Routing Protocols, Network Lifetime and Consumption Energy, wireless sensor networks, and web applications security.

Mohammad Shkoukani received his B.Sc. degree from Applied Science University, Amman, Jordan in 2002, M.Sc. and PHD degrees from The Arab Academy for Banking and Financial Sciences, Amman, Jordan, in 2004, and 2009 respectively, all in Computer Information Systems. His research interests include Agent Oriented Software Engineering, System Analysis and Design, and Electronic Commerce Applications.