Mobile Information Systems 7 (2011) 147–163
DOI 10.3233/MIS-2011-0115
IOS Press

A web-based application of TELOSB sensor network

Tarek R. Sheltami[a,*], Elhadi M. Shakshuki[b] and Hussein T. Mouftah[c]
[a]*Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia*
[b]*Jodrey School of Computer Science, Acadia University, Wolfville, Nova Scotia, B4P 2R6, Canada*
[c]*School of Information Technology and Engineering, University of Ottawa, ON, Canada*

Abstract. Sensor network can be used in a numerous number of applications. However, implementing wireless sensor networks present new challenges compared with theoretical networks. In addition, implementing a sensor network might provide results different from that derived theoretically. Some routing protocols when implemented might fail to perform. In this paper, we implement three routing protocols, namely: Dynamic MANET on-demand, Collection Tree and Dissemination protocols. To compare the performance of these protocols, they are implemented using a Telosb sensor network. Several performance metrics are carried out to demonstrate the pros and cons of these protocols. A telemedicine application is tested in top of the implemented Telosb sensor network at King Fahd University of Petroleum and Minerals Clinic in Saudi Arabia, utilizing Alive ECG sensors.

Keywords: TELOSB, MANET, wireless sensor networks, alive ECG sensor

1. Introduction

Wireless Sensor Network (WSN) is a network of devices that have one or more types of sensors connected together via wireless communication to monitor cooperatively different environmental conditions such as temperature, humidity, sound, etc. in different locations [4,10]. Lately, WSNs are deployed in a variety of applications such as hazard detection applications, healthcare applications, smart homes applications, and industrial applications traffic control [11,14,15] WSN has unique characteristics such as ability to withstand harsh environmental conditions, ability to cope with nodes failures, communication failures and heterogeneity of nodes. There are other challenges like mobility of nodes, dynamic network topology, deployment and synchronization. A node or a mote, in a WSN, consists of sensors, a microcontroller, a transceiver and batteries. Motes use special operating systems that are designed for wireless embedded sensor networks, such as TinyOS [24,25] and contiki [7]. Vendors use different communication protocols for networking; however, the most commonly used protocol in WSNs is Zigbee [5,30]. Motes may have different interfaces, such as USB and RS-232, and they are mainly used for programming and data transfer. The work presented in this paper tests Telosb motes [22] from Crossbow shown in Fig. 1. These motes are equipped with USB interface, TinyOS operating system and Zigbee communication protocol.

[*] Corresponding author: Tarek R. Sheltami, Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. Tel.: +966 3 860 4687; Fax: +966 3 860 3059; E-mail: tarek@kfupm.edu.sa.

1574-017X/11/$27.50 © 2011 – IOS Press and the authors. All rights reserved

Fig. 1. Telosb mote.

This paper discusses different versions of TinyOS and investigates different multi-hop network monitoring protocols. In addition, a testbed of 20 Telosb motes is used for performance evaluations. Finally, a real world application using Alive ECG sensors in top of the developed Telsob network is tested on Electrocardiography medical environment. Some of the Quality of Service (QoS) issues [1–3] are being adopted in our ECG application.

2. TinyOS 1.x vs. TinyOS 2.x

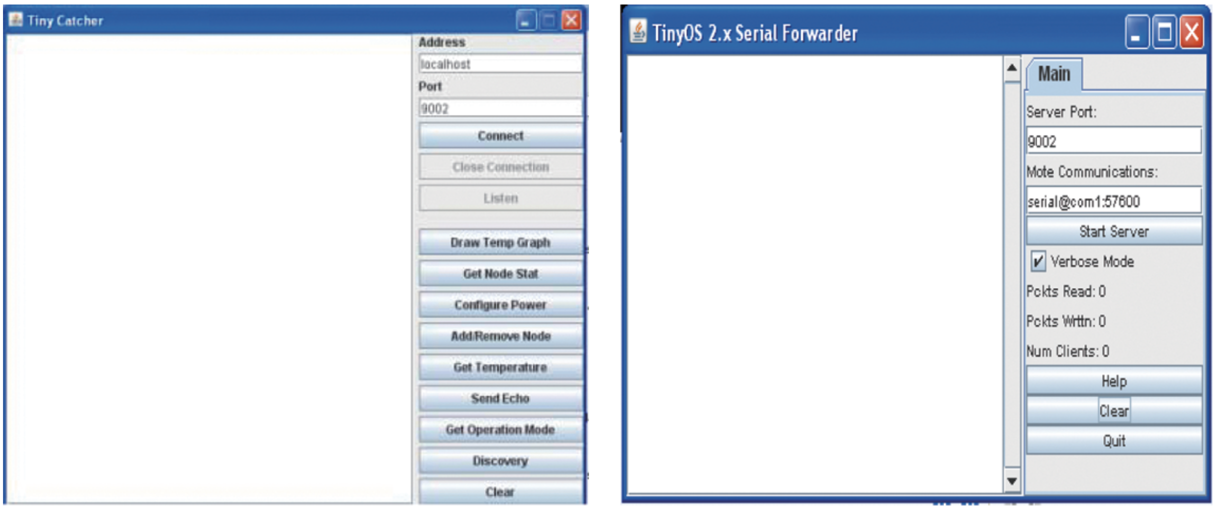
There are several differences exist between TinyOS 1.x [26] and TinyOS 2.x [27] operating systems. For example, TinyOS 1.x has very limited interfaces and structures. These limitations have motivated and encouraged several embedded systems operating systems designers to produce TinyOS 2.x to overcome the problems of the previous version (i.e., TinyOS 1.x) that led to easy programming practices of sensor networks, especially for new programmers. TinyOS 2.x differs from Tinyos 1.x in several aspects. Some of these aspects are: booting and initialization, task scheduling, timers, arbitration, debugging and network protocols. In booting and initialization, TinyOS 2.x has a different boot sequence than TinyOS 1.x. TinyOS 2.x boot sequence do only initialization of the components that are connected to it in the beginning. After it finishes the initialization, the boot sequence signals boot event. Conversely, in TinyOS 1.x any component connected to the boot sequence will be powered up once the boot sequence starts even if the initialization is not complete.

For task scheduling, both versions of TinyOS follow a non-preemptive FIFO algorithm, but TinyOS 2.x offers more flexibility, simplicity and organized task scheduling than Tinyos 1.x. In TinyOS 2.x, every task has its own place in the task queue and can only be posted once by the component. If the task is needed to be posted more than once, the task can repost itself once it finishes. The post operation fails only if the task is already posted. In TinyOS 1.x, the task queue is shared among all tasks, and the tasks can be posted multiple times. This sometimes may cause the task queue to be full, which makes any further post operation a failure. In addition to this disadvantage of TinyOS 1.x, programmers do not have the freedom to use their own scheduling algorithms as in TinyOS 2.x.

For timers, TinyOS 2.x offers more timer interfaces than TinyOS 1.x. It also makes it possible for component to check how much time left before a timer is fired and to make timers to be fired in the future. Moreover, it can make different timers to start asynchronously.

For arbitration, or access to shared resources, TinyOS 1.x lacks in providing a mechanism for managing access to these resources. This problem is eliminated in TinyOS 2.x by the Resource interface. Resource interface provides a policy for managing shared resource accessing.

For debugging, TinyOS 1.x gives only two types of error codes: SUCCESS and FAIL. These two types restrict programmers from being able to understand the cause of the errors. This can be problematic for programmers who want to know the cause of the error. On the other hand, TinyOS 2.x provides four types of error codes. These error codes are SUCCESS, FAIL, EBUSY and ECANCEL.



Address: localhost
Port: 9002

Server Port: 9002
Mote Communications: serial@com1:57600
☑ Verbose Mode
Pckts Read: 0
Pckts Wrttn: 0
Num Clients: 0

(a) (b)

Fig. 2. (a) TinyCatcher interface, (b) SerialForwarder interface.

For network protocols, Tinyos 2.x has updated versions of the two basic network protocols: Dissemination and Collection. These two protocols provide support for more applications with improved results than TinyOS 1.x can provide.

The aforementioned aspects describe a subset of advantages that makes TinyOS 2.x surpasses TinyOS 1.x. TinyOS 2.x have provided great improvements in power management, sensors and communication in a wide range of platform support. TinyOS 2.x advantages made it natural for many researchers in wireless sensor networks to be adapted for their application environments.

3. Graphical user interfaces

To facilitate the interaction between the user and the application environment, a web-based application is built utilizing two Java application interfaces, namely: TinyCatcherand and SerialForwarder [25]. These two interfaces are shown in Figs 2-a and 2-b respectively.

TinyCatcher is a tool that allows the user to control the nodes in the sensor network by reading the messages received from different nodes. It records these messages in a database for analysis purposes. It consists of the following:

Address: address of the base-station PC that has SerialForwarder running and the localhost if TinyCatcher is on the same PC as SerialForwarder.

- Port: port number used by SerialForwarder.
- Connect/Close Connection: for connecting and disconnecting to/from the SerialForwarder. A certain connection protocol to SerialForwarder must be followed.
- Listen: start listening on TCP/IP port for messages after establishing the connection.
- Draw Temp Graph: creates a graph for temperature reading.

In our application, we are only interested in temperature mission. TinyCatcher is designed to interpret the temperature readings; however, it could be designed to interpret other phenomenon.

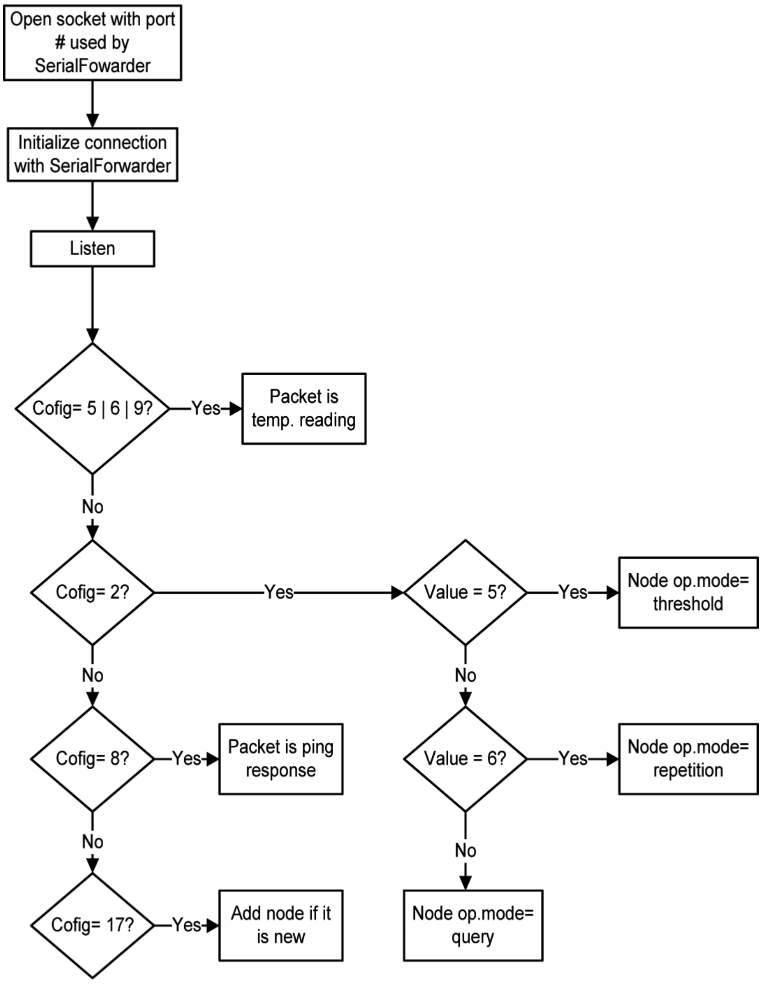


Fig. 3. TinyCatcher operation.

TinyCatcher depends on the SerialForwarder, which is a Java tool built in TinyOS. This tool is used for reading messages that arrive through the serial port from the gateway node (base-station). This allows other applications to share the connection over TCP/IP, which provides an advantage of remote monitoring. In other words, SerialForwarder acts as a proxy for reading and writing packets from and to the sensor network. Instead of reading data directly from the serial port, TinyCatcher can connect to SerialForwarder. The operation flow of TinyCatcher is demonstrated in Fig. 3.

Socket programming is used in the operation of the TinyCatcher. At the beginning, a socket is opened and a connection is initialized with the Serial forwarder. We set the ports 5, 6 and 9 for temperature readings, port 2 for control packets such as threshold, repetition and query, port 8 for ping response and port 17 to add new nodes.

The SerialForwarder has a very simple interface which is shown in Fig. 2-b. It consists of the following:

– Server Port: port of the socket the SerialForwarder opens.
– Mote Communication: specify serial port number and baud rate of the mote. It should follow this syntax [serial@(serial port):(baud rate)]. Serial port depends on the platform type, e.g. Win-

dows/Cygwin or Linux. Baud rate can be a numerical value or mote type, e.g. Telosb.

- Pckts Read: number of packets read from the serial port.
- Pckts Wrttn: number of packets sent through the serial port.
- Num Clients: number of clients connected to the SerialForwarder.

4. Sensor node configuration

In Telosb motes, the transmission power is divided into 31 power levels, where level 1 is the minimum and level 31 is the maximum. Many algorithms are introduced to solve both the transmission and the reception problem [6], some of which are described below.

In the Local Mean Algorithm (LMA) [8], every node starts with the same transmission power and then periodically broadcasts a life message including the transmitter's identity. After successfully receiving the life messages, the receivers respond by sending a life acknowledgement message that contains the address of life message transmitter. Therefore, the transmitter is able to count the number of reachable neighbors at a particular power. If the number of received acknowledgements is less than a predefined threshold value, the transmitter increases its power by a certain predefined factor to allow more nodes to respond. Moreover, the degree of power increases for a single step does not exceed a specified value. The minimum value allocated to both thresholds is used as the transmission power for the next operation cycle. The power is decreased if the number of acknowledged messages is greater than the threshold value.

The Transmission Power Control and Blacklisting (PCBL) [8] algorithm has two main objectives. The first objective is to investigate the behavior of low-power wireless communication links with respect to varied transmission power under different settings. The second objective is to propose a new scheme on power control that is capable of removing low quality links. At the beginning, each node measures the quality of each link by determining Packet Reception Rate (PRR) at various transmission power levels. In order to set a threshold value for the required link reliability, an optimal unicast transmission power for each link is set. The threshold is set to the minimum value of the observed PRR if the PRR is greater than the threshold. Otherwise, it is set to the maximum transmission power. Another threshold, which is known as the blacklist threshold, is set to consider whether a link can be converted to a good link. If not, then that link is blacklisted and no longer used.

The Transmission Power Control mechanism proposed in [16] uses dynamic transmission power adaptation. This mechanism starts with a node transmitting a beacon message. A receiving node with a good link quality, records the ID of the sender. The list of the IDs is piggybacked on the beacon message. Finally, a node knows the number of its neighbors by determining its ID in the incoming messages. The number of neighbors is then compared to the predefined threshold value and the transmission power control mechanism is applied to achieve the goal. The transmission power is increased if the observed number of neighbors is less than a predefined threshold value. The mechanism is capable of reducing the degree of adjustment as the number of neighbors converges to the targeted value.

The Adaptive Transmission Power Control (ATPC) approach described in [20] is designed based on the concept of changing a pair-wise transmission power level over time. As a result, each node assigns a different minimum transmission power for each link. Two main ideas behind its design is a neighbors table which is maintained by each node and a closed loop for transmission power control which runs between each pair of sensors. The entries of the table include Node ID, proper transmission power levels defined as the minimum power which provides a good link quality, and several parameters used

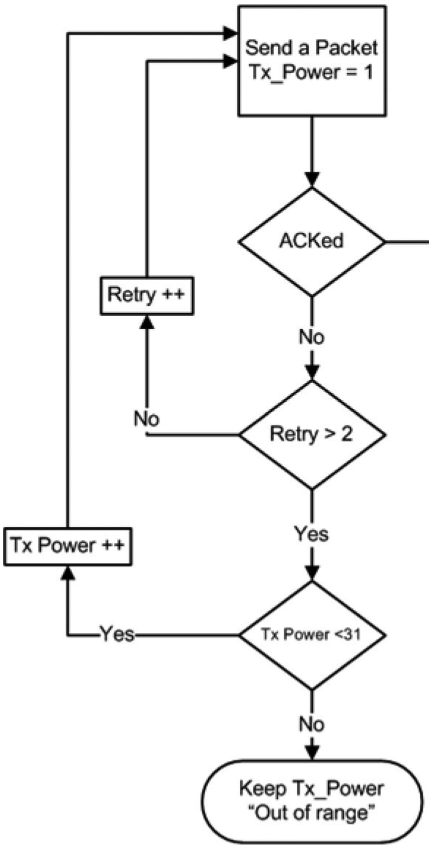


Fig. 4. Proposed power control flow.

for linear predictive models of transmission power control. The closed loop feedback is used to obtain the minimum transmission power by gradually adjusting the power.

According to the reviewed transmission power control schemes, most of them demonstrate two similar procedures. The first procedure can be described as: a transmitting node discovers which or how many active neighbors it has by broadcasting messages. The second procedure can be described as: a feedback or acknowledgement system is used after the neighbors have successfully received the messages.

In our approach, the nodes attempt to discover the minimum required transmission power. A packet is sent with minimum transmission power and then waits for an acknowledgment message. The acknowledgments are generated by the CC2420 radio transceiver as specified in the IEEE 802.15.4 protocol. If a packet is acknowledged then the node keeps its current transmission power. If the transmission fails after three attempts in a row, the node increases its power by one until it gets an acknowledgment. If no acknowledgement is received and reaches level 31, then this means that the other node is out of range. This is described in Fig. 4.

The Telosb sensor node can be configured for different power levels from 1 to 31 (1 is the lowest). Figure 5 depicts Java graphical user interface that allows users to set the power level for any given node using NodeID.

Figure 6 shows an interface that allows a user to add or remove nodes at the base-station.

Through the interfaces shown in Fig. 7, the user is allowed to set the desired period, threshold value and query. These operation modes can be described as follows:

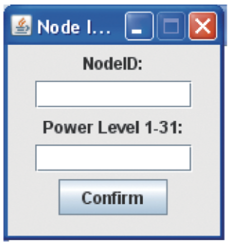


Fig. 5. Power configuration interface.

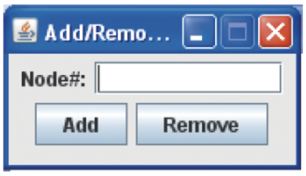


Fig. 6. Node add/remove interface.

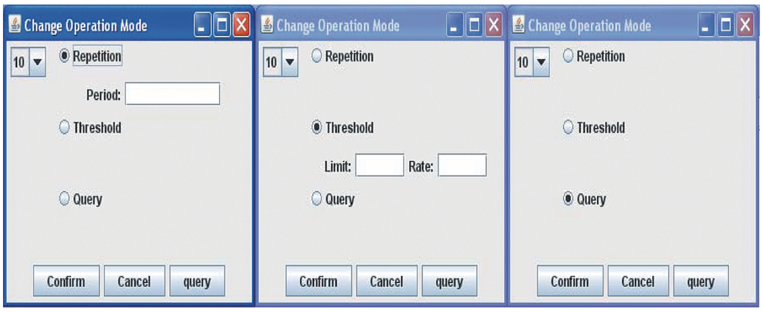


Fig. 7. Operation modes interface.

- Periodic: messages are sent periodically by the selected node.
- Threshold: specify the desired minimum and maximum temperature thresholds for the node to start sending messages when temperature reaches the threshold. Rate of the incoming messages is specified.
- Query: for sending a query to the specified node.

5. Implementation

Three routing protocols: Dynamic MANET On-demand (DYMO) Routing Protocol [12,13], Collection Tree Protocol (CTP) [9,10,19,23], and Dissemination Protocol [9,21,28,29] are implemented using Telsob motes with TinyOS2 operating system. We test and evaluate each of these protocols and compare their advantages and disadvantages. Typically, the base-station (gateway) acts as a bridge between the computer application side and motes. Thus, the data about temperature is collected at the base-station. At the base-station, the payload packet is extracted and forwarded it via the serial port to the computer.

5.1. DYMO routing protocol

After testing the implemented DYMO protocol, we notice that the protocol is unstable. DYMO protocol shows some unpredictable behaviors. In many cases, the nodes stop responding or refuse to discover new routes when old route(s) is removed. Although we could have improved the results with troubleshooting and using different methods, our decision is geared towards investigating other protocols due to time limitation. It should be noted that the tested operation is bi-directional, i.e. the base-station and sensor nodes are using the same routing protocol.

5.2. CTP protocol

A tree around a root is formed for CTP protocol to function. Nodes chose their parents based on the Link Quality Indicator (LQI), which could be calculated using Received Signal Strength Indicator (RSSI) and packet drop rate. Some nodes are equipped with Chipcon CC2420 radio chip that provides such statistics. This resulted in a new CTP called CTP-LQI. The only difference between CTP and CTP-LQI is in how the link quality calculated. Since Telosb motes use the CC2420 chip, it is more appropriate to use the CTP-LQI implementation. We observe that CTP routing protocol works much better than DYMO protocol. Unlike DYMO protocol, CTP data is limited to the size of the tree. This means that the data can only be sent from the edges towards the base-station.

5.3. Dissemination protocol

The Dissemination protocol ensures that every node in the network receives a copy of the data. This protocol is used by the base-station. Although it can be used for both ways communication, the fact that every node has to receive the message creates unneeded traffic in the network. This in turn has an effect on the battery performance. This protocol works by sharing a global variable over the whole network. However, this variable needs to be of a limited size that can be sent in one message. The dissemination protocol ensures, over time, the whole network shares the same value of the shared variable. Such value code is any data type of the default TinyOS data types or a user defined C structure that the user can select [24].

6. Performance evaluation

This section discusses the measurement of the RSSI of Telosb motes, the power field distribution, the throughput, and delay of transmitting packets.

6.1. RSSI measurement

The RSSI of Telosb motes is measured by fixing the power level for different distances. As mentioned earlier, the power levels of a Telosb mote range from 1 to 31 levels. Where, 1 refers to the weakest signal power level and 31 refers to the strongest signal level. RSSI is measured in dBm unit ranging from 0 to -155 dBm. We measure the RSSI for a certain power level using the following steps:

1. Fix the power level to have the variables as Distance and RSSI.
2. Place two motes away from each other by a certain distance X.
3. Send 200 packets by the first mote to the second.

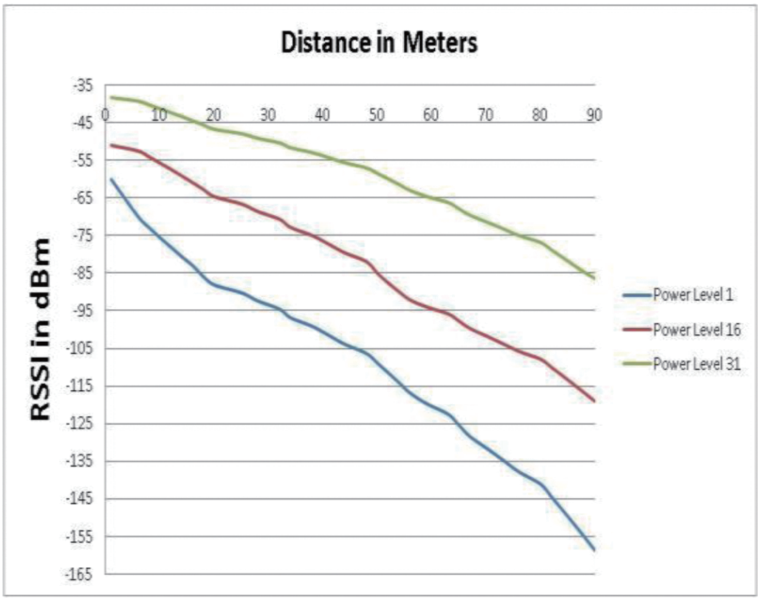


Fig. 8. RSSI readings.

4. Measure the signal strength for each packet received by the second mote.
5. Calculate the average RSSI.

Figure 8 depicts different readings for RSSI for levels 1, 15 and 31. Although the results are intuitive and expected, the main reason of this experiment is to find specific readings of the sensor nodes.

6.2. *Power field distribution*

In many research papers, it is assumed that the transmission power is in circular shape. We assume the same and performed the below experiment to find the power field distribution, using the following algorithm:

1. Place two motes away from each other by 3 meters.
2. Place the receiver to the right of the sender.
3. Face the sender mote towards north (the USB port of the mote is directed towards north).
4. Send 100 packets to be received by the second mote.
5. Measure the RSSI of all received packets
6. Calculate the average RSSI of the received packets.
7. Repeat the previous steps three more times, by having the sender facing south, east and then west.

At the completion of the experiment, we obtained the measurements of RSSI for Telosb mote when it is facing all four directions: north, south, east and west. All derived values are listed in Table 1 and depicted in Fig. 9.

From this experiment, we noticed that the transmission power close to elliptical shape. Also, we found that the RSSI is best when the sender is facing east (the receiver) with the USB port facing the receiver. Hence, it is best to have the USB interface of the sender mote facing towards the receiver.

Table 1
Power reading in different directions

Direction	RSSI (dBm)
North	−80.86
South	−81.12
East	−70.04
West	−83.45

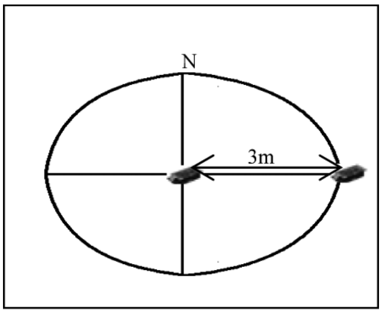


Fig. 9. Power field distribution.

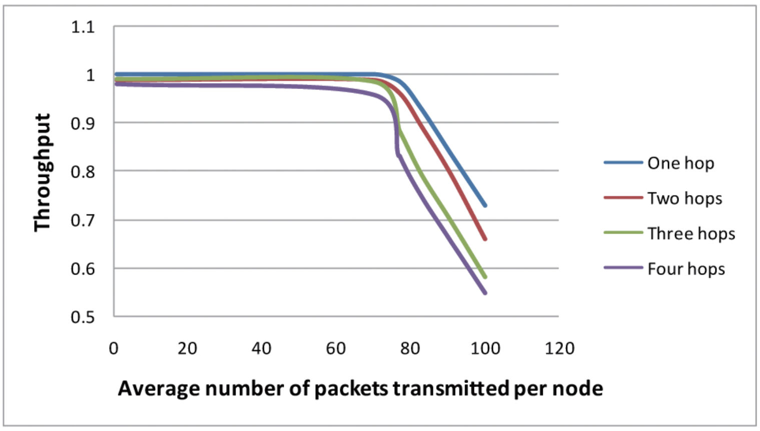


Fig. 10. Throughput vs. average number of transmitted packets.

6.3. Throughput

We utilize one, two, three and four hops topologies to measure the throughput of the transmitted data from and to the base-station. The following algorithm describes how we performed this experiment:

1. Fix the time period of transmitting packets to 1 sec.
2. Set the number of packets to be transmitted.
3. Calculate the throughput (number of received packets over transmitted packets in one second period).
4. Repeat the previous steps for different number of packets.

Figure 10 depicts the throughput of the network vs. the average number of packets transmitted. We notice that the performance of the networks starts to degrade when the average number of packets reaches 75 with sharp degradation when it reaches 80 packets. This is due packets collision.

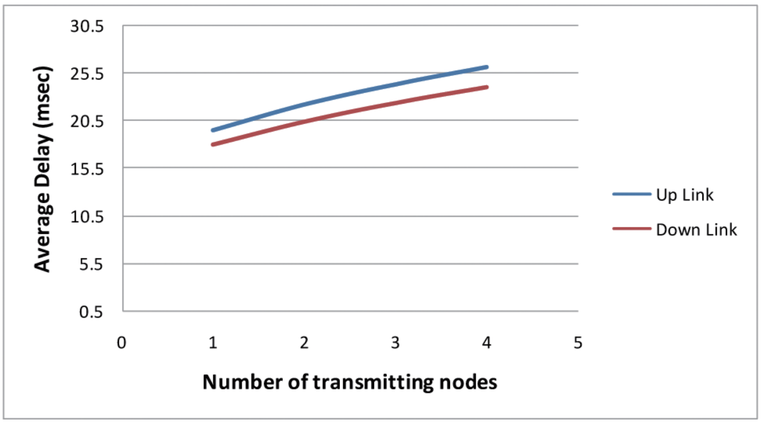


Fig. 11. Average delay (per packet) vs. number of transmitting nodes.

6.4. *Delay*

In order to find the delay of the transmitted packets between any two motes, many messages are transmitted between two specific nodes and then the average delay is calculated. The sender mote incorporates a timestamp with the packet for the receiver to calculate the time of transmission. The delay is compared with the number of nodes transmitted to the base-station. Figure 11 depicts average delay of the transmitted packets versus the number transmitting nodes. As expected, the more the number transmitting nodes the more the delay.

6.5. *A Medical application: Electrocardiography* (*ECG*)

Telemedicine system can be defined as the delivery of medical information over a distance by using a telecommunication means. This system helps and serves the patients and allows their physicians to check them without the actual physical contact, independent of their geographical locations. In this regard, we design a web-based interface between ECG device and any personal computer for the elderly and disabled people at home. Then, the application analyzes the data coming from the ECG device and according to some parameters, it decides whether the patient has an abnormal case or not. As soon as the patient has some difficulties and abnormalities, an email and/or SMS message is sent to the physician defining the patient ID and his situation.

Accurate identification of a heart disorder is vital to saving a life. A physician or an expert like the cardiologist always looks for tools that help in detecting certain heart disorder within the patient. An ECG has provided them the insight to detect these disorders by identifying variations in the functions of the heart. The main parts of the ECG are described in the following paragraphs.

6.5.1. *ECG signal*

An electrocardiogram (ECG) is a recording of the electrical activity of the heart over time produced by an electrocardiograph, usually in a noninvasive recording via skin electrodes. Its name is composed of three different parts: electro, because it is related to electrical activity, cardio, Greek for heart, and gram, a Greek root word meaning “to write”. Electrical impulses in the heart originate in the senatorial node and travel through the heart muscle where they impart electrical initiation of systole or contraction of the heart. The electrical waves can be measured at selectively placed electrodes (electrical contacts) on the skin.

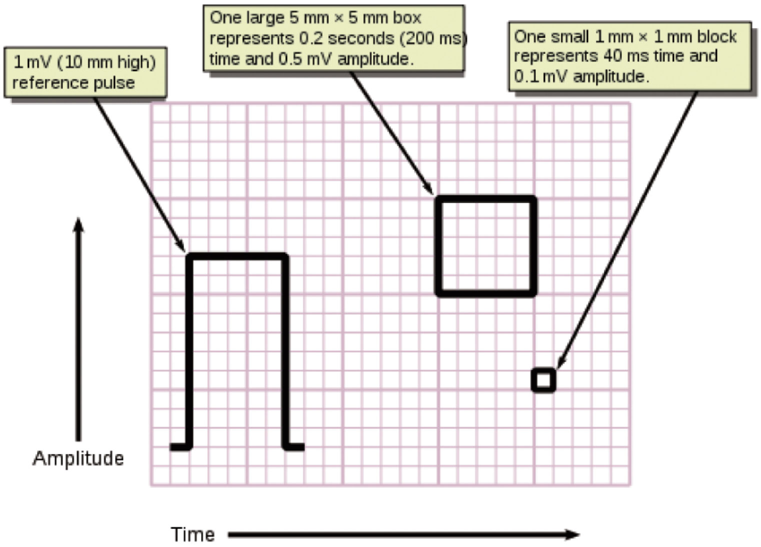

1 mV (10 mm high) reference pulse
One large 5 mm × 5 mm box represents 0.2 seconds (200 ms) time and 0.5 mV amplitude.
One small 1 mm × 1 mm block represents 40 ms time and 0.1 mV amplitude.
Amplitude
Time

Fig. 12. One second of ECG graph paper.

Electrodes on different sides of the heart measure the activity of different parts of the heart muscle. An ECG displays the voltage between pairs of these electrodes, and the muscle activity that they measure, from different directions, also understood as vectors. This display indicates the overall rhythm of the heart and weaknesses in different parts of the heart muscle. It is the best way to measure and diagnose abnormal rhythms of the heart.

6.5.2. ECG graph paper

Timed interpretation of an ECG was once incumbent to a stylus and paper speed. Computational analysis now allows considerable study of heart rate variability. A typical electrocardiograph runs at a paper speed of 25 mm/s, although faster paper speeds are occasionally used. Each small block of ECG paper is 1 mm^2, as depicted in Fig. 12. At a paper speed of 25 mm/s, one small block of ECG paper translates into 0.04 s (or 40 ms). Five small blocks make up one large block, which translates into 0.20 s (or 200 ms). Hence, there are five large blocks per second. A diagnostic quality 12 lead ECG is calibrated at 10 mm/mV, so one mm translates into 0.1 mV. A calibration signal should be included with every record. A standard signal of 1 mV must move the stylus vertically 1 cm, which are two large squares on ECG paper.

6.5.3. Normal ECG waves and intervals

A typical ECG tracing of a normal heartbeat (or cardiac cycle) consists of a P wave, a QRS complex and a T wave. A small U wave is normally visible in 50 to 75% of ECGs. The baseline voltage of the electrocardiogram is known as the isoelectric line. Typically, the isoelectric line is measured as the portion of the tracing following the T wave and preceding the next P wave, as depicted in Fig. 13.

6.5.4. P wave

During normal atrial depolarization, the main electrical vector is directed from the sinoatrial (SA) node towards the atrioventricular (AV) node, and spreads from the right atrium to the left atrium. This turns into the P wave on the ECG, which is upright in II, III, and augmented vector foot (aVF) (since the general electrical activity is going toward the positive electrode in those leads), and inverted in augmented vector

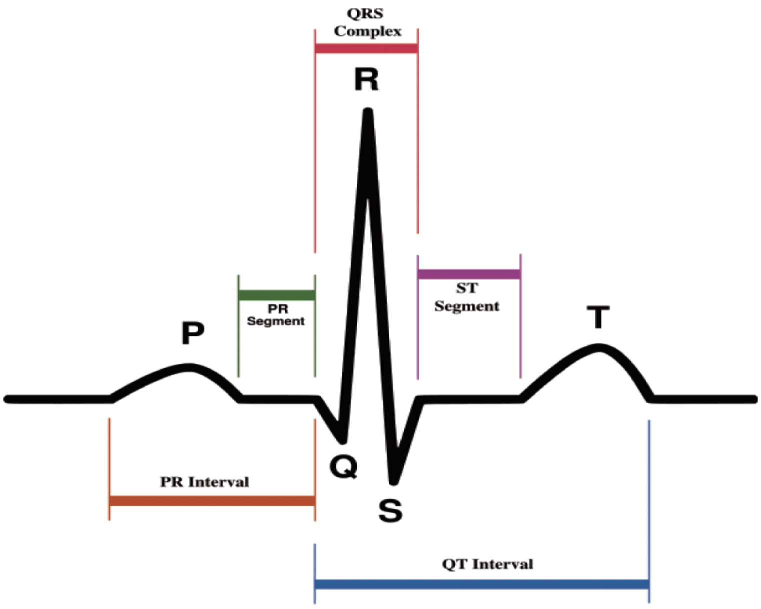


Fig. 13. Schematic representation of normal ECG.

right (aVR) (since it is going away from the positive electrode for that lead). A P wave must be upright in leads II and aVF and inverted in lead aVR to designate a cardiac rhythm as Sinus Rhythm.

- The relationship between P waves and QRS complexes helps distinguish various cardiac arrhythmias.
- The shape and duration of the P waves may indicate atrial enlargement.

6.5.5. QRS complex

The QRS complex is a structure on the ECG that corresponds to the depolarization of the ventricles. Because the ventricles contain more muscle mass than the atria, the QRS complex is larger than the P wave. In addition, because the His/Purkinje system coordinates the depolarization of the ventricles, the QRS complex tends to look "spiked" rather than rounded due to the increase in conduction velocity. A normal QRS complex is 0.06 to 0.10 sec (60 to 100 ms) in duration represented by three small squares or less, but any abnormality of conduction takes longer, and causes widened QRS complexes.

6.5.6. T wave

The T wave represents the repolarization (or recovery) of the ventricles. The interval from the beginning of the QRS complex to the apex of the T wave is referred to as the absolute refractory period. The last half of the T wave is referred to as the relative refractory period (or vulnerable period). In most leads, the T wave is positive. However, a negative T wave is normal in lead aVR. Lead V1 may have a positive, negative, or biphasic T wave. In addition, it is not uncommon to have an isolated negative T wave in lead III, aVL, or aVF.

To demonstrate our approach on a real world application, a Telsob network is tested on Electrocardiography medical environment. This environment consists of Alive ECG sensor [18], Telosb sensor network, application server, patient PC and database, as shown in Fig. 14. The Alive ECG sensor is connected to a patient and sends signals to the Telosb sensor network. The Alive ECG software is tested and ECG data is measured after a Bluetooth connection is established with a Bluetooth enabled device.

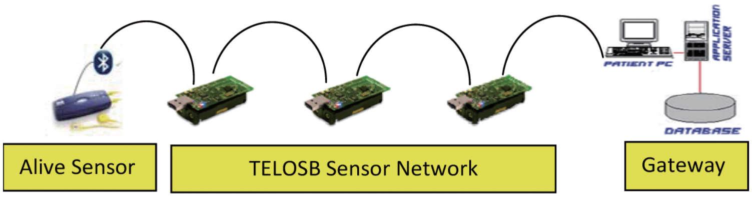


Fig. 14. Network topology of ECG environment.

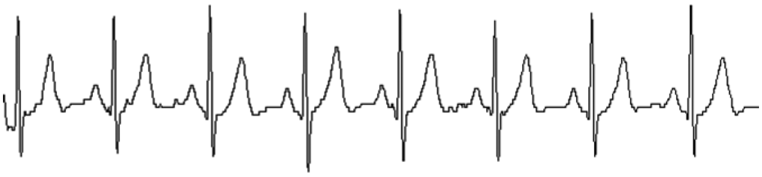


Fig. 15. A normal ECG wave.

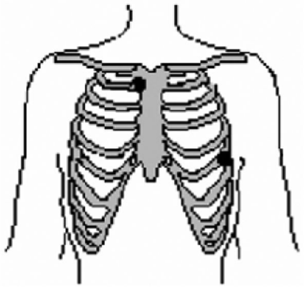


Fig. 16. Normal configure of ECG probes.

We interface one of the Telosb sensors with a Bluetooth connection. This allows us to receive the ECG signals from the Alive sensor, and then the signal is sent via the Telosb network to the gateway. Doctors and nurse are able to access gateway through the Internet.

We perform our experiments at the clinic of King Fahd University of Petroleum and Minerals (KFUPM), in Dhahran city, Saudi Arabia. Our patients are connected to the Alive ECG sensor and each patient sends his/her data via the Telosb network to the gateway.

During this experiment, 150 samples of ECG are collected over a period of one year using ECG Alive sensor. Figure 15 shows a normal (i.e. healthy) ECG signals that are captured by ECG Alive sensor.

ECG along with Heart Rate (HR) and three axes accelerometers readings are sent wirelessly using Telosb network. The normal configuration of the probes is shown in Fig. 16. The ECG Alive sensor is good enough for a doctor to detect any abnormal behavior in the ECG and specially for detecting Arrhythmia.

A few samples from the 150 collected samples are demonstrated in Fig. 17.

Figure 17 shows an abnormal ECG signal. The highlighted area shows a signal known as premature ventricular beat arrhythmia in the medical world. It is characterized by the following:

- Absence of a p-wave.
- QRS is wide.
- T-wave is inverted.

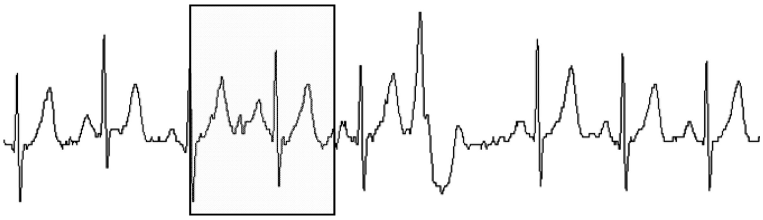


Fig. 17. Abnormal ECG signal.

At the gateway end, the data is analyzed using MATLAB. The data is fetched from a specific location and then based on that data some actions are taken by software that we design specifically for this application. This program has a Graphical User Interface that allows the nurses/doctors to interact with the system. The data is then saved in the server for future needs. It is worth mentioning that the collected data is very close to actual data collected by the ECG equipments. Using our developed system, we can test more than one patient simultaneously if needed. Each one of them will have to be connected with Alive ECG sensor and the rest is taken care of by the Teleosb sensor network.

Acknowledgement

The authors would like to thank King Fahd University of Petroleum and Minerals via projects IN080395 and IN090047 and University of Ottawa via WiSence project for their support. The authors would like to thank Research and Graduate Office of Acadia University and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support.

Conclusions and future work

An application with graphical user interfaces to control the Telosb motes is developed using TinyOS 2. Many interfaces were developed to build an application to carry out different experiments. Using this application, we were able to receive temperature readings from sensor nodes in single and multi hop fashions. Three modes of operations were developed to receive the temperature reading (periodic, threshold and query). Dissemination and Collection Tree protocols were implemented on top of sensor nodes. Collection Tree protocol was used for the uplink (from nodes to base-station), where nodes choose their parents based on the Link Quality Indicator. However, Dissemination protocol was used by the base-station to ensure that every node in the network receives a copy of the data. On the other hand, DYMO protocol was unstable and showed some unreliable behaviors.

An efficient power transmission algorithm was introduced to find out the optimum power of transmission in order to conserve energy. In addition, transmission power strength could be adjusted for each node at the base-station. Exact values of the received signal strength vs. distance, average throughput vs. the offered traffic and delay vs. number of transmitting nodes were measured and will be used in future simulation. Also, an exact shape of the radio power field was found with helical shape.

An ECG medical application was tested at a local clinic at KFUPM utilizing Telosb network and Alive ECG sensors. One hundred and fifty samples of health and abnormal patients were collected for duration of one year. The collected data was tested against the traditional ECG equipments and it was found that there was very small margin of error, due to calibration of the Alive ECG sensor. It is worth mentioning that our application received excellent remarks from cardiologists and nurses at KFUPM clinic.

References

[1] A.B. Waluyo, B. Srinivasan and D. Taniar, *A Taxonomy of Broadcast Indexing Schemes for Multi Channel Data Dissemination in Mobile Database*, the 18th International Conference on Advanced Information Networking and Applications (AINA 2004), Volume 1, IEEE Computer Society, pp. 213–218, 2004.

[2] A.B. Waluyo, B. Srinivasan and D. Taniar, *Optimal Broadcast Channel for Data Dissemination in Mobile Database Environment*, Proceedings of the 5th International Workshop on Advanced Parallel Programming Technologies (APPT 2003), Lecture Notes in Computer Science, Volume 2834, Springer, pp. 655–664 2003.

[3] A.B. Waluyo, B. Srinivasan and D. Taniar, Research in mobile database query optimization and processing, *Mobile Information Systems* **1**(4) (2005), 225–252.

[4] A. Durresi and M. Denko, Advances in mobile communications and computing, *Mobile Information Systems* **5**(2) (2009), 101–103.

[5] A. Durresi, P. Zhang, M. Durresi and L. Barolli, Architecture for mobile Heterogeneous Multi Domain networks, *Mobile Information Systems* **6**(1) (2010), 49–63.

[6] C. Li and L. Li, Energy efficient resource management in mobile grid, *Mobile Information Systems* **6**(2) (2010), 193–211.

[7] Contiki, http://www.sics.se/contiki/.

[8] D. Son, B. Krishnamachari and J. Heidemann, *Experimental Study of the Effects of Transmission Power Control and Blacklisting in Wireless Sensor Networks*, IEEE SECON, Santa Clara, USA, pp. 289–298 October 2004.

[9] Dissemination, http://www.tinyos.net/tinyos-2.x/doc/html/tep118.html.

[10] E. Shakshuki, H. Malik and T. Sheltami, *Lessons Learned: Simulation Vs WSN Deployment*, The IEEE 23nd International Conference on Advanced Information Networking and Applications (AINA-09), pp. 580–587, Bradford, UK, May 26–29, 2009.

[11] H. Truong and S. Dustdar, A survey on context-aware web service systems, *International Journal of Web Information Systems, Emerald* **5**(1) (2009), 5–31.

[12] I. Chakeres, E. Belding-Royer and C. Perkins, *Dynamic MANET On-demand* (*DYMO*) *Routing*, draft version 04 IETF, Mar. 2006.

[13] I.D. Chakeres and J.P. Macker, Mobile ad hoc networking and the ietf, *SIGMOBILE Mob Comput Commun Rev* **10**(3) (2006), 79–81.

[14] I.F. Akyildiz and I.H. Kasimoglu, *Wireless Sensor and Actor Networks: Research Challenges*, Ad Hoc Networks, pp. 351–367, 2004.

[15] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, *Computer Networks* **38**(4) (2002), 393–422.

[16] J. Jeong, D.E. Cullar and J.H. Oh, *Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks*, Technical Report No. UCB/EECS-2005-16.

[17] M. Kubisch, H. Karl, A. Wolisz, L.C. Zhong and J. Rabaey, *Distributed Algorithms for Transmission Power Control in Wireless Sensor Networks*, IEEE WCNC, New Orleans, Louisiana, pp. 558–563, 2003.

[18] Mao Ye, Chengfa Li, Guihai Chen and J. Wu, *EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Network*, the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC), Phoenix, Arizona, USA, pp. 535–540, 2005.

[19] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis and A. Woo, *The Collection Tree Protocol* (*CTP*), TEP 123, TinyOS Network Working Group, Aug. 2006.

[20] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He and J.A. Stankovic, *ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks*, the 4th International Conference on Embedded Networked Sensor Systems, Boulder, Colorado, pp. 223–236, November 2006.

[21] T. Bokareva, N. Bulusu and S. Jha, *A performance comparison of data dissemination protocols for wireless sensor networks*, in IEEE Globecom Wireless Ad Hoc and Sensor Networks Workshop, Texas, USA, pp. 85–89, 2004.

[22] Telosb, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[23] The Collection Tree Protocol (CTP), http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html.

[24] TinyNode, http://www.tinynode.com/.

[25] TinyOs, http://www.tinyos.net/.

[26] tinyos-1.x, http://www.tinyos.net/tinyos-1.x/doc/.

[27] tinyos-2.x, http://www.tinyos.net/tinyos-2.x/doc.

[28] Vinh Pham, Erlend Larsen, Øivind Kure, Paal Engelstad, Routing of internal MANET traffic over external networks, *Mobile Information Systems* **5**(3) (2009), 291–311.

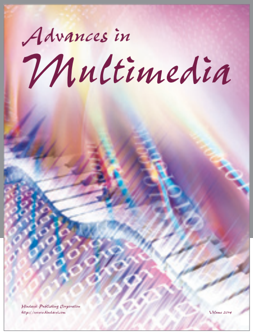
[29] Y. Yu, R. Govindan and D. Estrin, *Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks*, UCLA Computer Science Department Technical Report, Tech. Rep. UCLA/CSD-TR-01-0023, 2001.

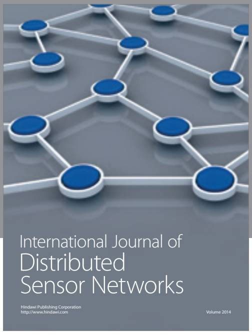
[30] Zigbee, http://www.zigbee.org.

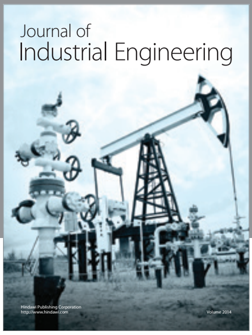
Tarek Sheltami is currently an associate professor at the Computer Engineering Department at King Fahd University of Petroleum and Minerals (KFUPM) Dhahran, Kingdom of Saudi Arabia. He joined the department on August 26, 2004. Before joining the KFUPM, Dr. Sheltami was a research associate professor at the School of Information Technology and Engineering (SITE), University of Ottawa, Ontario, Canada. He worked at GamaEng Inc. as a consultant on Wireless Networks (2002–2004). He was the Principle\Co Investigator of several research projects in the area of Ad hoc, Sensor Networks and Pervasive and Ubiquitous Computing.

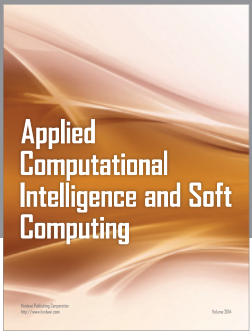
Elhadi M. Shakshuki is a professor at the Jodrey School of Computer Science at Acadia University, Canada. He is the founder and the head of the Cooperative Intelligent Distributed Systems Group at the Computer Science Department, Acadia University. He received the BSc degree in computer engineering in 1984 from El-Fateh University, and the MSc and PhD degrees in systems design engineering respectively in 1994 and 2000, from the University of Waterloo, Canada. Dr. Shakshuki is an Adjunct Professor at Dalhousie University, Canada. He manages several research projects in his research expertise in the area of intelligent agent technology and its applications. He is a member of IEEE, ACM, AAAI and APENS.

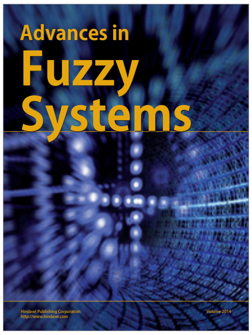
Hussein T. Mouftah joined the School of Information Technology and Engineering (SITE) of the University of Ottawa in 2002 as a Tier 1 Canada Research Chair Professor, where he became a *University Distinguished Professor* in 2006. He has been with the ECE Dept. at Queen's University (1979–2002), where he was prior to his departure a Full Professor and the Department Associate Head. He has six years of industrial experience mainly at Bell Northern Research of Ottawa (became Nortel Networks). He served IEEE ComSoc as Editor-in-Chief of the IEEE Communications Magazine (1995–97), Director of Magazines (1998–99), Chair of the Awards Committee (2002–03), Director of Education (2006–07), and Member of the Board of Governors (1997–99 and 2006–07). He has been a Distinguished Speaker of the IEEE Communications Society (2000–07). Currently he serves IEEE Canada (Region 7) as Chair of the Awards and Recognition Committee. He is the author or coauthor of 6 books, 32 book chapters and more than 950 technical papers, 10 patents and 140 industrial reports. He is the joint holder of 12 Best Paper and/or Outstanding Paper Awards. He has received numerous prestigious awards, such as the 2008 ORION Leadership Award of Merit, the 2007 Royal Society of Canada Thomas W. Eadie Medal, the 2007–2008 University of Ottawa Award for Excellence in Research, the 2006 IEEE Canada McNaughton Gold Medal, the 2006 EIC Julian Smith Medal, the 2004 IEEE ComSoc Edwin Howard Armstrong Achievement Award, the 2004 George S. Glinski Award for Excellence in Research of the U of O Faculty of Engineering, the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario (PEO), and the Ontario Distinguished Researcher Award of the Ontario Innovation Trust. Dr. Mouftah is a Fellow of the IEEE (1990), the Canadian Academy of Engineering (2003), the Engineering Institute of Canada (2005) and the Royal Society of Canada RSC: The Academy of Science (2008).











Hindawi

Submit your manuscripts at
http://www.hindawi.com