

Wireless communication system for a wide area sensor network

Magali Cortez and Jaime Sánchez

Department of Electronics and Telecommunications, Applied Physics Division
Center of Scientific Research and Superior Education of Ensenada
Km. 107 Tijuana-Ensenada Hwy. Baja California, México
{mcortez, jasan}@cicese.mx
<http://www.cicese.mx>

Abstract. In this paper, the design and implementation of a wireless communication system specific to wide area sensor network capable of establishing ad hoc connections is presented. First, the application to which this system is applied is explained, and then a brief description of the system is made. In this system two routing algorithms were implemented, one based on a static links routing table and the other is the M to 1 algorithm proposed by Lou W. in [1]. Tests were carried out with the designed system and the routing algorithms mentioned, accomplishing satisfactory communication between nodes in a test scenario for the first case.

Key words: Sensor network, wireless networks, wide area, microcontrollers, CR10X acquisition card

1 Introduction

Nowadays, wireless sensor networks represent one of the most innovative technologies. In general, this type of networks consists of a certain number of nodes distributed within an area. The nodes are provided with some type of sensor, and thus are used to collect and process data [2]. No matter what the application be, the sensor nodes that integrate this network have similar characteristics: small size, a transmitter-receiver module, a microprocessor and at least one sensor; limited in energy consumption, transmitting power, memory and processing capacity [3]. These nodes communicate by forming an ad hoc network, in which data is being sent from a source node to a destination node through intermediate nodes. This way, the covered range of the network is extended. There are commercial wireless communications systems developed under the IEEE 802.15.4 standard for wireless sensor networks that offer personal area coverage, which are suited in applications where the nodes are distributed in a small area or where the density of sensors per unit area is large. The problem arises when a larger area is trying to get covered, at an urban or greater level, where the nodes are separated by long distances, and to form an ad hoc topology each node requires to transmit with enough power to reach to its neighbors, which

is not possible with commercial wireless sensor network systems due to the fact that these transmit with very little power (some dBm). The Network of Deformations of the Mexicali Valley (RDEFVM) of the Department of Sismology at CICESE [4], presents a scenario where the sensor nodes are being separated by a few kilometers. The collection of data generated by each node is done manually by monthly visits which cause a delay in the analysis of the acquired data, and implies an economic and physical burden, specially on summer months when the Valley's high attainable temperature may reach from 40 to 50 C. A remote recollection of the data with a wireless network would avoid those problems, with a given benefit to the researchers in charge of the RDEFVM.

This report presents the design and implementation of a wireless communication system specific to a wide area sensor network capable of establishing ad hoc communications. A graphical user interface necessary to the control and recollection of data, and a web page for the remote recollection of data via internet are developed. The rest of the document is organized as follows. In Section II, a description of the system designed in this project is given and the routing algorithms implemented are explained. In Section III, the test scenario is explained and the results obtained are presented later. Finally, in Section IV, the conclusions of this work are presented.

2 Structure of the system

This section describes the general architecture of the Wireless Sensor Network, the design methodology, the routing algorithm and the graphic interface developed for the user.

2.1 Wireless long range communication system.

In a wireless sensor network applications where the nodes are separated by long distances, each node is required to transmit with enough power to reach its neighboring nodes in order to create an ad hoc network. To provide wide area coverage to the RDEFVM, each node was equipped with a high power transmitter and a high gain antenna. To establish ad hoc communication within the network, each node had a microcontroller with a routing algorithm loaded previously in its memory. Because this system tries to give a solution to the problem of data collection that the RDEFVM represents, the characteristics of the nodes that integrate this network were considered in the design.

The RDEFVM has different types of sensors, but to begin, only displacement sensors (inclinometers) will be used. Therefore, each node in the network will have an inclinometer, a CR10X data acquisition card from Campbell Scientific and a 20W solar panel capable of powering the node. The sensor is connected to the card, which has the task of acquiring, processing and storing the data generated. The data acquisition card's output is through a serial communications port. Otherwise, to have access to the network from a remote device through the internet, at least one of the nodes must be connected to it. A graphical user

interface is necessary for the remote control of the network and the manual and automatic collection of the data captured by the nodes.

2.2 Network Architecture.

The architecture consists of the sensor nodes, an access node, a web page server and a remote terminal. The sensor nodes form the basic level of architecture of the system. A sensor node performs the network functions as well as sensing, since it processes and sends the data generated by itself and the rest of the sensors to the access node.

The access node is responsible for collecting and transmitting the information of the sensor nodes to a remote terminal via internet. The access node is a personal computer that communicates wirelessly to the sensor nodes and has specialized software for the recollection of data. The web page server hosts a web page necessary for the remote access to the network via internet. For simplicity, the server can be installed in the access node. Lastly, the remote terminal can be any computer with internet access. Figure 1 shows the architecture described previously.

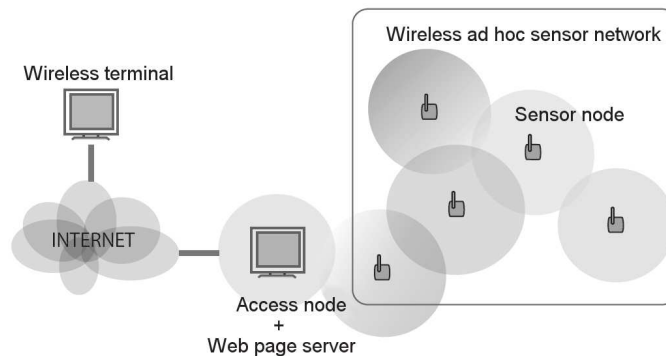


Fig. 1. System architecture

2.3 Design of the sensor node.

Originally, the RDEFVM's nodes were equipped with a sensor, a data acquisition card and a battery connected to a solar panel. The following paragraphs present the design of the electronic circuit that needs to be added to each node in order to establish wireless communication. From this on, the electronic circuit added will be called the wireless interface.

Figure 2 shows the block diagram of the sensor node, name which is endowed to the original node of the RDEFVM once the wireless interface is added. The

wireless interface can be divided into two blocks: one represents the wireless communication and the other the logical control.

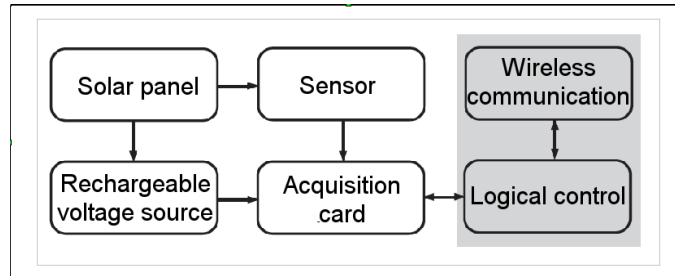


Fig. 2. Sensor node block diagram

Due to the fact that the nodes are separated by distances measured in kilometers, a high power card is required in order for the signal emitted by a node can reach some other neighboring node in the network. The wireless card by which the network is designed is the AC4790 of Aerocomm, which transmits with 1W of power and employs FHSS as the modulation technique [5].

On the other hand, to give greater coverage to the node, a half wave magnetic dipole antenna with a gain of 2 dBi and a 900Mhz operating frequency is used.

Since the AC4790 wireless card does not have the ability to establish ad hoc communications in the network, it is necessary to integrate in each node a microcontroller to handle the data transfer and link information. A program previously loaded in the microcontroller's memory handles the routing as well as the logical control of the entire node. Among its tasks is the communication with the wireless card and the communication with the acquisition card for the recollection of the data generated by the sensors.

The microcontroller used is Microchip's PIC18LF542 [6], which has serial communications through a TX transmission line and an RX reception line. Since it is necessary to establish communication with the acquisition card as well as with the wireless card, a switch is required to choose between those two devices. The SN74CBTLV3257 from Texas Instruments is a digital switcher, better known as a MUX/DEMUX, from one to two lines of four bits, and is the device used for such purpose. Figure 3 shows a block diagram for the connection between those three devices and figure 4 shows a picture of the wireless interface prototype designed.

2.4 AdHoc communication.

The packet flow in an AdHoc network is from a source node to a destination node, using intermediate nodes as relays. This class of communication may be done by either broadcast or routing algorithms. In the broadcast mechanism, packets

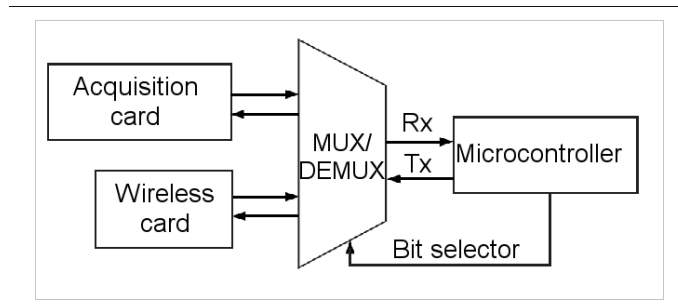


Fig. 3. Chip select function performed by the microcontroller.

are sent from a source node and are retransmitted by the neighbor nodes until they eventually reach their destination.

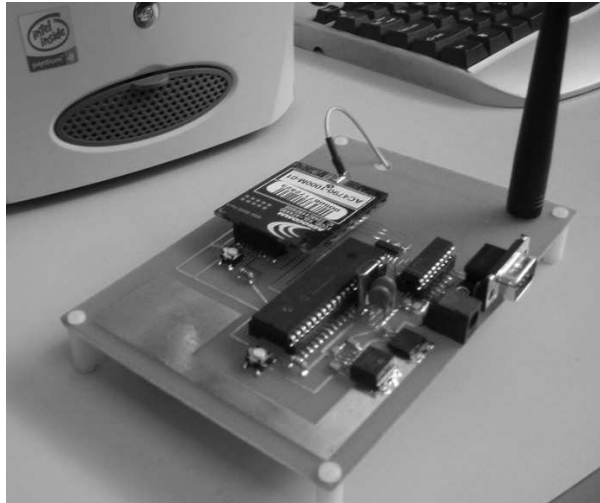


Fig. 4. The wireless interface prototype

There are two options for the broadcast mechanism: flooding and rumor broadcasting. In flooding, each node checks the arriving packets and if it happens to be the destination, it process the packet; otherwise it retransmits the packet to all its neighbors, as long as the maximum number of hopes has not been reached. This mechanism does not need to know the network topology, hence adding or deleting nodes to the network is an easy process. However, this mechanism has several problems, among them is the implosion problem, which means that duplicate packets are sent to the same node, and the superposition that occurs when two nodes send the same packet simultaneously.

Another broadcasting mechanism is called rumor, which is similar to flooding, with the difference that this one retransmits the packets to just one neighbor (which is chosen in a random way), which reduces the implosion problem [7].

The routing algorithms base their operation in a table that contains routes to reach all other nodes. A route is a sequence of hops to other nodes, needed to reach the destination node. Each node has its own routing table, depending of its location inside the network. The network topology must be known by all nodes, and it is necessary a route maintenance mechanism to update the routing tables in case of any change. An example of a routing algorithm with a route maintenance mechanism is the AODV [8], which was designed for Mobile AdHoc Networks where the network topology changes frequently.

In a network with few nodes, the broadcasting mechanisms are more adequate since the packets may reach the destination node with relatively few retransmissions. Otherwise, if a routing algorithm is employed, the power consumption may be optimized since the route to deliver the packets is known in advance, and just the necessary retransmissions are made; however, it needs memory space to store the routing tables. Since the RDEFVM nodes are fixed, and the geographical location of each node is known in advance, it was decided to implement a routing algorithm based on a static routes table; considering that the microcontroller chosen for implementing the AdHoc communication has enough memory to store a small routes table.

2.5 Routing algorithm.

The static routes table used for the network counts with three fields: destination node, next node and number of routes to the destination node. Once the destination node is known, the next node is looked up in the table so the package can reach its destination. The number of available routes is checked in order to get to the destination node, if there is more than one possible route, and the package cannot get to its destiny using the first route, then the remaining routes will be used.

Roughly, the microcontroller constantly checks the serial port, waiting for a packet to arrive. If the address of the node receiving the packet matches the packet destination address, the remaining of the header is processed, and the corresponding actions are executed on the acquisition card. In this case, an internal packet is constructed and sent to the acquisition card, asking for an answer to a command. Once the acquisition card performs the asked task, the microcontroller receives the answer and constructs a packet to be sent to the Access Node. The constructed packet includes the next node address for transmitting the packet to the destination node, that is obtained by consulting the routing tables.

The constructed packet is sent by the node through the wireless communications card, and waits for an Acknowledge (ACK) from the next node. Once the ACK is received, the microcontroller initializes all variables and registers used and waits for the next packet (from the wireless card) to be processed.

In case of receiving a negative ACK, the constructed packet will be retransmitted until a positive ACK be received or until the number of retransmissions is reached. The retransmission opportunities include the search for a new route, if the retransmission number has been reached for a particular route.

Whenever an alternate route to the destination node exists, which means that the number of routes field in the routing table is greater than one, the packet is sent again until acknowledged or until the retransmission number is depleted. If still the packet is not received correctly, the session is closed and the registers and variables are reinitialized, and the process remains waiting for a new packet. When the packet destination address does not match that of the node receiving it, the packet is retransmitted to the next node en route to the final destination. To avoid the possibility of remaining in a waiting state forever, a timer is initialized at the beginning of the interaction process of the microcontroller with the acquisition card. In case the timer expires (after 2 seconds), and no answer be received from the acquisition card, the card is considered damaged and the message error in the acquisition card is generated and sent to the access node. Once the error message is sent, the microcontroller choses the antenna signal and returns to the waiting loop, checking for the reception of a packet through the serial port, returning to a stable situation.

Packets used to implement the routing algorithm. The structure of the packets used for implementing the routing algorithm is shown in figure 5. Each packet contains the field name and the number of bytes for each field. Three of the packets use a common 7 bytes header, which is needed for the radio chip (AC4790) to configure some parameters for the payload transmission. The header contains the following information: packet header identifier (81H), payload size (128 byte maximum), session duration (time during which the communication among nodes is active), number of retransmissions (256 maximum), and MAC address of the next node.

Command type, code 0x01 (7 bytes header + 45 bytes)

| 81 | Payload Data Length | Session count refresh | Transmit Retries | Next hop MAC | Message type | Destination MAC | Command Type | Available bytes |
|----|---------------------|-----------------------|------------------|--------------|--------------|-----------------|--------------|-----------------|
| 1 | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 40 |

Data type, code 0x02 (7 bytes header + 128 bytes)

| 81 | Payload Data Length | Session count refresh | Transmit Retries | Next hop MAC | Message type | Destination MAC | Data + MPTR |
|----|---------------------|-----------------------|------------------|--------------|--------------|-----------------|-------------|
| 1 | 1 | 1 | 1 | 3 | 1 | 3 | 124 |

Error type, code 0x83 (7 bytes header + 4 bytes)

| 81 | Payload Data Length | Session count refresh | Transmit Retries | Next hop MAC | Message type | Destination MAC |
|----|---------------------|-----------------------|------------------|--------------|--------------|-----------------|
| 1 | 1 | 1 | 1 | 3 | 1 | 3 |

ACK type, code 0x82 (4 bytes)

| 82 | 0x00: Failure, 0x01: Success | RSSI | RSSI* |
|----|------------------------------|------|-------|
| 1 | 1 | 1 | 1 |

Fig. 5. Packets used for implementing the routing algorithm in the system.

Inside the payload field is added another header, which is used by the micro-controller to identify four types of packets: Instruction, Data, Error, and ACK. The first byte value of this header identifies the type of packet and the next three bytes specify the destination address of the packet. Each packet may include other fields on its header, depending on the packet type.

The M-to-1 algorithm. A routing table is easy to implement and is suitable for scenarios where few fixed nodes are involved. However, if a node ceases to be a part of the network or if a new node is added (in other words, the topology of the network changes), an updating mechanism for the routing table is needed, otherwise the communication of the network would be hampered because the nodes would not count with valid routes; for this reason, an M to 1 algorithm proposed in [1] was used.

This algorithm takes advantage of the simple flooding by adding two mechanisms which, at the end of the flooding allow the nodes to know alternative routes to get to the base node, which is the node that initiates the flooding.

The M-to-1 algorithm bases its operation in the fact that at the end of a simple flooding, a tree is obtained, which is headed by the base node with branches formed by the closest nodes in which each branch is independent. This algorithm tries to unite the mentioned branches to provide more return routes for the base node to get to each node.

The message structure employed is type, mid, nid, bid, cst, route, where type is the type of message, in this case it's RPRI; mid is the number of the sequence of the updating route; nid is the message sending node's identifier; bid is the branch identifier of the node that sent the message that is the nid of the nearest node to the base node, cst is the cost of the route that could be the number of jumps; route is the path by which the package has traveled. For additional details regarding the M to 1 algorithm, consult [1].

Integration of the M-to-1 algorithm to the designed system. The M-to-1 algorithm allows the nodes to have at least one route back to the base node, or one-way communications. In order to provide the system two-way communications, it is necessary to add another algorithm that allows the base node to know the route(s) to get to a determined node. The algorithm added for that purpose uses the following messages: RREQ and RREP.

RREQ is a routing request sent by the base node. RREP is the response to that petition that the destination node sends to the base node when it receives RREQ. RREQ is a broadcast message while the RREP is sent following a route discovered previously by using the M-to-1 algorithm.

The RREQ message counts with the fields Type, destination (broadcast), number of jumps, sequence number, destination node, previous node. The RREP message includes the fields Type, destination, sequence number, origin node, message sender node. When the z node receives a RREQ message, it updates the number of jumps and checks the destination node, if it is its neighbor it sends a RREP placing in destination the address of the neighbor node and in

previous node it's own ID. If the destination node is not its neighbor, then it resends the message, updating the number of jumps and placing it's own ID in the previous node field. When the RREQ reaches it's destination node, this node sends a RREP, in which destination is the base node, it's own ID in the origin node, and in the sending node it places it's own ID. This RREP is sent to the node which is indicated in the previous node field of the RREQ message being answered. When the z node receives a RREP message, it updates it's routing table with the information contained in the message (destination node = origin node; next jump = sending node). The z node places in the sending node field it's own ID, searches in the routing table a route to get to the base node (previously discovered with M-to-1 algorithm) and it sends a RREP message.

The process consists of flooding the network with the RPRI message. After that, the base node sends a RREQ message to know the route to get to the z node. The base node waits for a RREP message sent by the z node. After a t period, if the RREP message has not been received, a RREQ is sent again with a new sequence number; if the RREP message is received, a request of data is sent to the z node using the route contained in the RREP message. To ask for data from the acquisition card to the z node, a REQUEST message is employed.

2.6 Access node.

The access node controls the collection of data generated by the nodes in the network. This node consists of a personal computer that communicates wirelessly with the sensor nodes and has software designed for the collection of data from the nodes. It's installed in a location that has 120V AC power and internet acces available. To endow wireless communication to the access node, a multiport card was used included in the designer's kit by Aerocomm. That card includes an AC4790 card and communication options via RS232, RS485 and USB.

2.7 Graphical user interface for the control and collection of data and a Web page for remote access.

The program of the access node was made in LabView 7.1. The program sends the intructions necessary for the download of the data from the acquisition card, using packets that contain appropriate header to travel thourgh the wireless network to get to their destination. In the following paragraphs the program for control and collection of data is described.

The program performs the control funcions and the collection of data for the sensor network. The collection of data can be manual or automatic. In manual mode, the user has to select the desired node to connect, configures some parameters of the serial communication and begins to download. With automatic download, the user has to program beforehand a list of tasks. The list of tasks indicates the hour and the node number from which the data needs to be collected. In both cases, manual and automatic, at the end of each succesful download the data collected is being sent via email to the registered users of the system. Figure 6 shows the GUI developed.

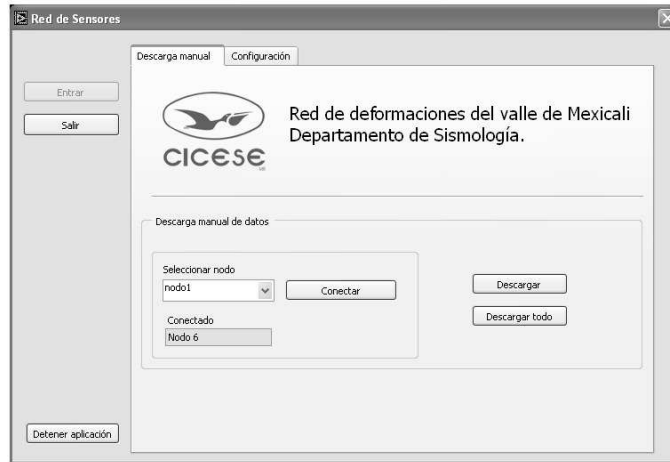


Fig. 6. Graphical user interface for the control and collection of data.

To have remote access to the wireless sensor network via internet, a web page as well as a web page server are needed. For the sake of simplicity, the web page server included in LabVIEW version 7.1 was used, which allows to publish web pages that contain Virtual Instruments. This way, the user can have access to the same program installed in the access node through a web page. The web page server is installed in the computer designated as the access node, so it is necessary for that computer to have a public IP address so that the users can access the web page.

3 Test scenario and results.

The system was tested making downloads of data from some of the sensor nodes which were placed inside the CICESE campus, where an ad hoc topology was employed and a GUI ran on a personal computer. The transmitting power of the AC4790 cards was 10 dBm and the maximum distance between the nodes was 300 m. Of course, the output power of the AC4790 can be set up to 1W to achieve greater coverage.

The AdHoc communication between the nodes, using the algorithm based on a static routing table was satisfactory, achieving 100 % of connections. Nevertheless, when testing the system with the M-to-1 algorithm and the forward communicating algorithm, problems caused by collisions between broadcast messages appeared. This problem is related to the AC4790 card, which cannot resolve the collisions generated by sending broadcast messages at the same time by different nodes. The solution to that problem was not addressed in this work, and is a pending task.

4 Conclusions

The wide area wireless communication system designed provides ad hoc communication using devices intended for point to point links. The results obtained were satisfactory, achieving multi jump communication in a testing network platform. In principle, the system designed tries to give a solution to the problem of data collection imposed by the RDEFVM. However, the system developed can be used in any application that requires long range wireless communications and that employs the CR10X data acquisition card from Campbell Scientific.

Acknowledgments. The authors want to thank CONACYT, for their support for project No. 7410 "Deformaciones de la Corteza en el Valle de Mexicali", in charge of Dr. Ewa Glowacka, from which this work was originated.

References

1. Lou W.: An efficient N-to-1 multipath routing protocol in wireless sensor networks, 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Washington DC. (2005).
2. Agrawal D. P., Biswas R., Jain N., Mukherjee A., Sekhar S. y Gupta A.: Sensor systems: state of the art and future challenges. En: Wu J. (ed.). Handbook on theoretical and algorithmic aspects of sensor, ad hoc wireless, and peer-to-peer network. Auerbach Publications, New York (2005) 318–345.
3. Haenggi M.: Opportunities and Challenges in Wireless Sensor Networks. En: Ilyas M. y Mahgoub I. (ed.). Handbook of sensor networks: compact wireless and wired sensing systems. CRC Press, Florida (2005) I-1.
4. RDEFVM: <http://sismologia.cicese.mx/Redes/Rdefvm.html>(2006)
5. Aerocomm: AC4790 900 MHz OEM transceivers. Users Manual, version 1.3 (2006).
6. Microchip: <http://ww1.microchip.com/downloads/en/DeviceDoc/39631B.pdf>(2006)
7. Heinzelman W.R., Kulik J. y Balakrishnan H.: Adaptive protocols for information dissemination in wireless sensor networks. 5th annual ACM/IEEE international conference on Mobile computing and networking (1999) 174–185.
8. Perkins C. E. y Royer E. M.: Ad hoc On-Demand Distance Vector Routing. 2nd IEEE Workshop on Mobile Computing Systems and Applications (1999) 90–100.

