International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2100-2104

New Hybrid Gateway for Wireless Sensor Network

Fihri Mohammed[1] and Ezzati Abdellah [2]

Mathematics and Computer Science Departmen,
LAVETE Laboratory, Faculty of Sciences and Technical Settat, Morocco.
[1] fihrimohammed@gmail.com,[2]abdezzati@gmail.com

Abstract

In this paper we will focus on how to use table driven, on demand and tree architecture protocol at the same time by introducing our hybrid gateway. At first we will describe the principal behind our gateway, how the solution is combination between hardware and software. As an example how we use usb wifi card that support Ad-Hoc mode to connect to each other using either AODV or OLSR, and xbee serie 2 for Zigbee. The broadcasting service is ensured by hostapd, the routing by linux deamon in case of olsr and aodv, however in Zigbee case the protocol is embedded in the radio card its self and we use SPI communication "after all the routing is done" to collect data from root card. Afterwards we describe how the data is treated and assembled in one formatted XML file or simply send in formatted way by SMS. After that, we will put our gateway on stressful environment (worst use case scenario) and see its reaction in terms of bandwidth and energy, and if those results are suitable for some applications types of WSN. In this order we will present two use cases of our gateway in real life although it can support more than that by a simple and easy code refactoring, or simply by using linux features.

Keywords: WSN, OLSR, AODV, ZIGBEE, GATEWAY, HYBRID.

INTRODUCTION

Wireless Sensor Network (WSN) consists of deploying unspecific number of nodes and sometimes in unspecific area in order to create a network using all the nodes or some of them to route the data to the base station. The node can be stable or in movement. Several routing protocol with different approach (active, proactive, hierarchical …) are used in this type of network but every approach has its advantages and disadvantages depending on the use case that it developed for (energy efficient, mobility QoS..). In this chapter, we create a new self-configuration multi-routing protocol gateway that can send collected data via several protocols OLSR, AODV, ZigBee to the server using XML.

By using raspberry, two 2.4 usb wifi card that support Ad-Hoc mod and xbee serie 2, we use aodv and olsr linux demon and we collect data and send it to server as xml file using 3g connection or formatted SMS and of course any Ethernet based communication such as a Vsat gateway to Ethernet.

First, every node, however the routing protocol in use must have an ID declared in database server, this Id will not be used for routing purpose but in order to identify data; if a node do not have already an Id must send a join request to the base station, on it turn, it sends it to the server in order to get an ID for the node and add it to the database server; in case where the connectivity between the base station and the server is lost the base station gives a temporary id to the node until the 3g or sms connections are established. At the moment of receiving data from sensors the base station store the data until it achieve the preconfigured size of the XML file then sent it to the server if the 3g connection is not available for the moment, the base station starts to create xml files depending on the preconfigured size and one the connection is up, the base station starts to update the server with the new files.

For OLSR and AODV we can use a DHCP server on the base station or simply work with static IP and all the preconfigured options can be added or modified from HTTP interface that uses PHP and python to configure the system. For the self-configuration of zigbee, we used some of xbee the pan id also can be modified from the interface. The gateway can also receive orders from the server like forcing interrogation of a sensor or putting it in sleep mode in case of a problem like energy or false information.

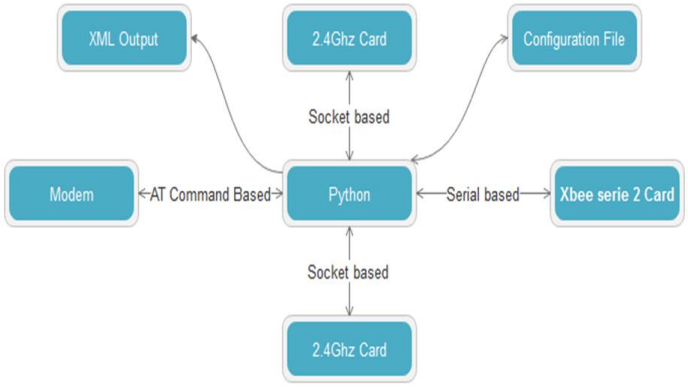

Figure 1. Gateway Aspect

OVERVIEW

The major aspect of WSN is that the sensor nodes are densely deployed in an open space; on a battlefield in front of, or beyond, enemy lines; in the interior of industrial machinery; in a biologically and/or chemically contaminated field; in a commercial building; at homes; or in or on a human body. A sensor node typically has embedded processing capabilities and on board storage, but it's limited in power, computational capacities, and memory [1].

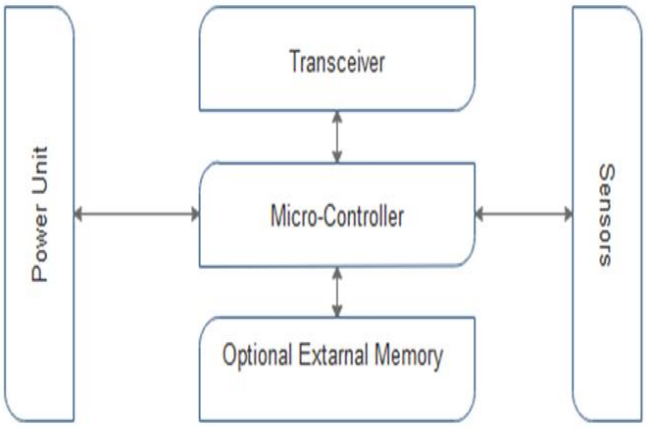

Figure 2. Node structure

In a sensor node, the main functionality of an embedded processor or microcontroller is to be the middleman between the other composites of the node such as sensor devices and transceiver. The microcontroller is also responsible for every computing or running algorithm in order to communicate with sensors with their specific communication protocol, store the measured information in the internal or external memory and process it if needed, and manage the power.

A Wireless Sensor network can be homogeneous or heterogeneous. In the first case, the nodes are similar to each other in every aspect construction, purpose and functions but it can be different in the sensed data type. In the second one, the nodes are different in their construction, and it depends on their functionality or the role that they play. Generally speaking, it possible to divide the nodes to three types, end-device, router or internal-gateway and network coordinator or base-station (also can be called root-access-point or external-gateway).

The end-device is a sensor node with one main functionality, which is to sense and send the results to its parent-network. It should not, in any case, be a part of networking aspects.

The router is the intermediate device between the end-device and the base station. Often it is responsible for the routing aspect and coordination in a part of the deployment area, and sometimes gathering and processing data before sending it to the base station. It can also be equipped with sensors like the end-device.

The base station takes the hand on the coordination of the network like synchronizing the data sending and topology control. Also it is responsible for sending the sensed data to the other type of network where the storage server is located. Usually it is provided by a good and long range transceiver, high level power source if not lined, powerful computing unit and connected to another network such us internet or satellite.

The topology of a sensor networks changes very frequently, due to the death or the sleep mode of some nodes and sometimes due to the mobility of the sensors nodes. There are several types of topologies in WSN Star, Tree, Mesh, Hybrid and even sometimes a customized topology that depends on the specific use but we can classify it under hybrid.

In a tree mode topology, there are several types of nodes, and the function of the node depends on its hierarchical level in the tree. A root node is placed on higher level in the tree and it plays the gateway or base station role. The presence of a relay router is optional, usually it is used in order to extend the range of communication and make treatment functions for the highest and the lowest level nodes. The sensor node is the end-device. This type of architecture reduces the energy consumption and increases the communication range.

The Mesh topology allows every sensor node in the network to connect to its neighbors by symmetric or asymmetric links and be a part of the routing mechanism. In this type of architecture the information can be transmitted in several paths, and the death of a node does not affect the network because of the existence of those different paths. However, when the number of the dead nodes goes high, the network starts to fail. The Hybrid topology, is a combination between two or three of the previously mentioned topologies, it gives us the flexibility to use the right topology in the right part of the deployment area.

Our gateway combine between the tree and mesh topology allowing the topology of the network to become as fallow.

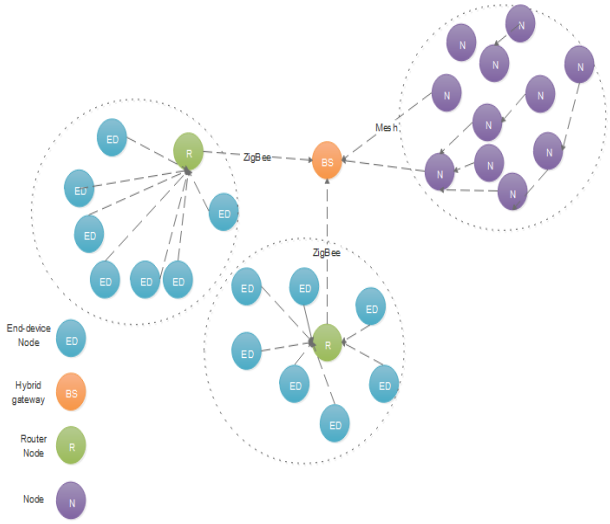

Figure 3. Network Architecture

HARDWARE

The For the gateway computing device we chose to work with raspberry pi, because it provide a powerful CPU with low energy consumption and the most importantly it supports several open-source Linux based operating system. The Linux based systems give us a solid infrastructure to build any solutions we desire using the already provided features by the kernel.

The Raspberry Pi is a nano-SBC ARM processor .This computer. The Raspberry Pi has an ARM11 700 MHz processor. It includes 1, 2 or 4 USB ports, an RJ45 port and

256MB of RAM to the original model, 512MB and 1GB on the latest versions. Its graphics circuit BMC VideoCore 4 decodes Blu-ray full HD streams (1080p 30fps), to emulate older consoles and run relatively recent video games.

Also we did use a powerful directional 16Dbi external Antenna in order to cover a large area. In this order, we chose to work with Ep-8523 wireless lan adapter card. The EP-8523 150Mbps high-power outdoor wireless 802.11N USB adapter is characterized by high gain and long transmission distance. In case there are no obstacles, it offers better transmission effect than other common products. The built-in 16dB directional antenna further improves the performance of this product. For safe wireless connection, the card supports 64/128-bit WEP data encryption, WPA, and other encryption ways. This powerful card will give us a high bandwidth for all multimedia and applications that require high bandwidth.

For all the low data rate demanding sensors we will use XBee digi card. These small "OEM" radio modules in the 2.4GHz band will allow the creation of a ZigBee® mesh wireless network (mesh) with different topologies. These are ideally suited for the management of wireless sensors. The versions with the (pro) termination have a higher "HF" power allowing them to benefit from a better range.

After various experiments, we observed that the behavior of the node is influenced not only by the electric current, but also by the stability of the voltage. So we add an electric current stability circuit that will not only ensures the right voltage but also serve as a safe connecter to the solar panel.

IMPLEMENTATION

The Main idea here is to broadcast a powerful SSID that can reach 1km and 200m indoor from a theoretical point of view. In this order, we used the HOSTAPD open source broadcasting service. This software allows us to broadcast an SSID using a g norm. But in this gateway, we want to achieve more bandwidth then presented before, whence came the use of the n norm. The n norm driver is provided by linux but not supported yet by HOSTAPD, so we use in HOSTAPD configuration file a G norm but we encapsulate it after in N norm driver. But even with n norm, the bandwidth is still not optimized, so we force HOSTAPD to use 40Mhz channel which leads to a significant increase in the bandwidth showed in results section. The use of DHCP server is optional (but for the further represented results the DHCP is used), this latter will allow us to avoid the use of HNA messages because the network configuration is sent to the client node in DHCP communication. The use of DHCP will affect how the data communication is implemented in the gateway configuration file (an iteration on DHCP IP range or table of STATIC IP). Also, the data secure mechanism presented in security chapter will be implemented. The gateway works also with perl programs in order to generate data XML files which will described in the data centralization section. The gateway must provide a route to the server side network. In this order, a forwarding mechanism must be implemented. The forwarding aspect is changed depending on the server side network. If the server side network is IP based (such us Ethernet, other ip Satellite) a simple iptables rules can achieve this but the server side must be a static IP or known domain name. However, if we want to implement a device based forwarding (SMS, Radio…) a specific perl program is used in order to communicate and control the device driver. In the next chapter, a new programming framework we will refactor all of these codes in order to simplify its use.

In our solution the data is divided to two major types active (or accumulated) and reactive. In accumulated data communication type, the gateway react as a client and the nodes as servers. The gateway demands a specific information from specific node using perl (IP socket programing in case of AODV or OLSR and SPI communication in case of Zigbee) by sending the formatted resource order (get_temp, get_hum, get_gaz, get_state,get_all…) the node reacts to the order by executing the responsible command function and send the data. After that, the gateway receive the data, an insertion is done in temporary XML file until the right amount of data is achieved. This operation is repeated periodically and for all the joined nodes, the ip or the ids node are collected from a configuration file in case of a static identifier or is iteration on a range of ids in case of dynamic identifier. The gateway can iterate in 5mn(more or less with few seconds) 250 nodes without using any thread or fork concept, the communication program is started with new parameters for each node and die after receiving the data, a thread concept can be implemented easily but with the 5mn response time we judge that is not needed. In the second type of data collection (reactive), the node plays this time the roll of client and the gateway plays the server roll. In this approach it's the node who decide when the information is sent following an event like door open or max gas reached or simply a periodic send. The sensed information is sent with a priority type (critical, major, and warning), the critical and the major information are forwarded to server immediately but in case of concurrence between those two the critical is sent first. As for the warning information is added to the other accumulated data.

COMMUNICATIONS AND RESULTS

First of all, we will describe how we collected the results and how those last ones are convenient to the WSN applications. In order to monitor the global power consumption in a node, we used an external Arduino board as a volt meter (this board does not intervene in any way with the solution). The generated log from Arduino is stored in the connected computer hard drive. As for the bandwidth, we will use in each client two iperf execution; one working with tcp connection trying to send nonstop periodic data, and the other one to send an AVI large file. As for Zigbee, we use a flooding attack in order to simulate big rate of sending data, in our case, it can not exceeds 255Kbits.

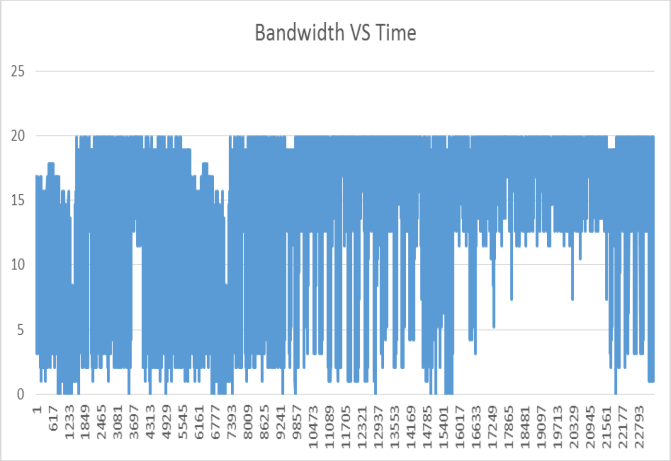

Figure 4. Bandwidth MBits to the server side network at 2 OLSR hopes using the n norm and 40 MHz

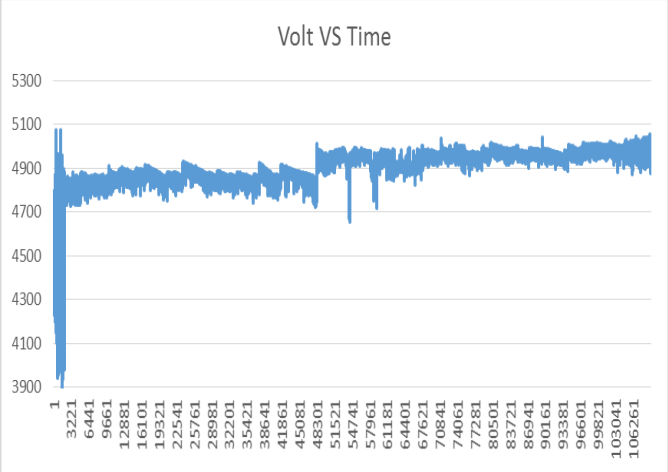

Figure 5. Energy in node without stabilization

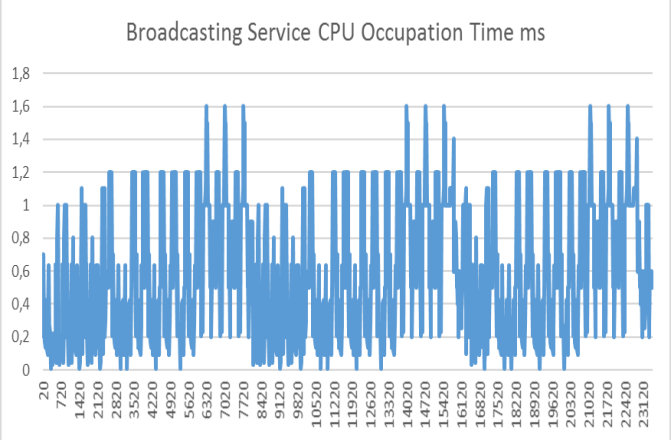

Figure 6. Broadcasting Service CPU Occupation Time ms

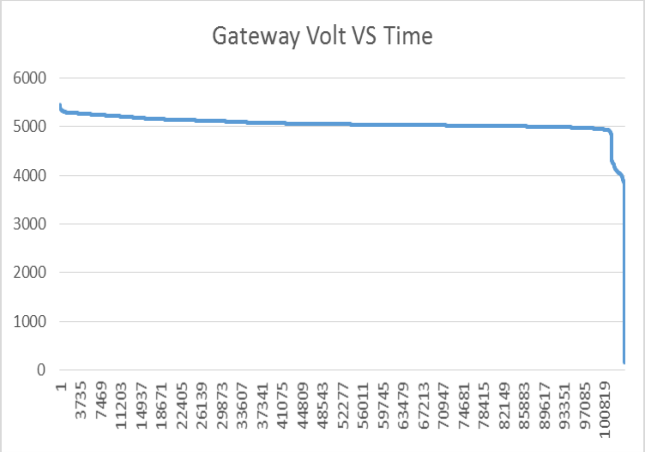

Figure 7. Gateway Volt with stabilization

As the above figures shows, our gateway can work in stressful and nonstop sending environment for at least 28hours with our customize 20000mah battery (the charging time of this battery is 7hours with 1A). So in case of energy constraint applications we can use a cheap 2A out solar panel using power stabilization card that ensure the right voltage all time, the thing that allows a 24h working time in the worst case scenario of WSN(no Sleep or Idle mode). Also for the multimedia we provided a 20Mbits/s and it is more than enough for full HD video/audio transmission.

CONCLUSION

The main goal of our hybrid gateway is to provide the right routing protocols for the right use case at the same time. The energy and the bandwidth results shows that our gateway can provide a good service for several applications that need a high data rate and at the same time a hierarchical architecture for small data rated aspect application such as one state (one or zero, dry contact) application (commanded doors, physical intrusion…), where all we need is to receive a state of an equipment or control it by a small digit.

REFERENCES

[1] Xiao, Y., Chen, H. and Li, F.Handbook on sensor networks. 1st ed. Hackensack, NJ: World Scientific.2010. Print

[2] Akyildiz, I. and Fuat. Wireless Sensor Networks. Chichester: Wiley, 2011. Print.

[3] Robert Faludi.Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing. O'Reilly Media, 2011. Print.

[4] Matthijs Kooijman. Building Wireless Sensor Networks Using Arduino (Community Experience Distilled). Packt Publishing - ebooks Account, 2015. Paperback.

[5] MACIEJ KRANZ .BUILDING THE INTERNET OF THINGS:

IMPLEMENT NEW BUSINESS MODELS, DISRUPT COMPETITORS, TRANSFORM YOUR INDUSTRY. WILEY, 2016. PRINT.

[6] HANES, D., SALGUEIRO, G., GROSSETETE, P., BARTON, R. AND HENRY, J. IOT FUNDAMENTALS: NETWORKING TECHNOLOGIES, PROTOCOLS, AND USE CASES FOR THE INTERNET OF THINGS. CISCO PRESS, 2017. PRINT.

[7] Conti, Marco, Jon Crowcroft, and Andrea Passarella. Multi-hop Ad Hoc Networks from Theory to Reality. New York: Nova Science, 2007. Print

[8] SWATI BHASIN, ANKUR GUPTA, PUNEET MEHTA Comparison of AODV, OLSR & ZRP Protocols in Mobile Ad-Hoc Network on the Basis of Jitter, International Journal of Applied Engineering Research (IJAER) ,Volume 7, Number 11 (2012)

[9] C T. CLAUSEN, P. JACQUET. RFC3626: OPTIMIZED LINK STATE ROUTING PROTOCOL (OLSR)

[10] SEEMA VILAS BHUJADE, PROF. S. D. SAWANT, ―EVALUATION AODV, DSR AND DSDV PROTOCOL OF MANET BY USING NS-2‖, IJET, VOLUME 4 ISSUE 8- AUGUST 2013.