Robust Framework for 6LoWPAN-based Body Sensor Network Interfacing with Smartphone

Thomas Gonnot, Won-Jae Yi, Ehsan Monsef and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*—This paper presents the design of a robust framework for body sensor network. In this framework, sensor nodes communicate using 6LoWPAN, running on the Contiki operating system, which is designed for energy efficiency and configuration flexibility. Furthermore, an embedded router is implemented using a Raspberry Pi to bridge the information to a Bluetooth capable smartphone. Consequently, the smartphone can process, analyze, compress and send the data to the cloud using its data connection. One of the major application of this framework is home patient monitoring, with 24/7 data collection capability. The collected data can be sent to a doctor at any time, or only when an anomaly is detected.**

I. Introduction

With the recent development of the Internet of Things, patient health monitoring is gaining more focus in recent research. While a single sensor can be easily connected to a smartphone using Bluetooth, managing several sensors can become complex and inefficient due to the nature of the protocol. The recent release of the Bluetooth Low Energy (BLE) protocol greatly diminished the limitation of the previous versions, but other alternatives are to be investigated before settling on this protocol.

One of the reasons the Bluetooth was chosen for this kind of application is not only the existence of a profile dedicated to health application, but also the compatibility with current smartphones. The presence of Bluetooth on almost all the present-day smartphones makes them a perfect platform for data collection and transmission to the cloud. It is to be noted that all the communications using Bluetooth are encrypted, preventing the unauthorized collection of the data collected by the sensors. One of the minor drawbacks of Bluetooth is the necessity for pairing the devices to a phone which could be solved using different techniques including Near Field Communication (NFC) tags facilitating the pairing process. Furthermore, once paired with one device, the sensor exclusively exchanges data with this device.

Another protocol described in this paper is 6LoWPAN. This protocol is based on IEEE802.15.4 MAC layer, which is also used by the ZigBee protocol. It offers advantages such as the possibility of creating a mesh network, or the compatibility with other packet switched networks, such as internet. Additionally, it offers an optional encryption for the communications using AES algorithm, as well as provides the network discovery and routing capabilities offered by IPv6. Another important aspect is the power consumption, which can be as low as Bluetooth Low Energy (BLE), and offer comparable performances in terms of power consumption and data throughput [1].

In this paper, we describe the implementation of a Wireless Body Sensors Network (WBSN) using simulated data on different 6LoWPAN enabled nodes. A Raspberry Pi board is used as a border router, which collects, organizes and then transmits the data to a smartphone using Bluetooth. Figure 1 shows the organization of this type of network.

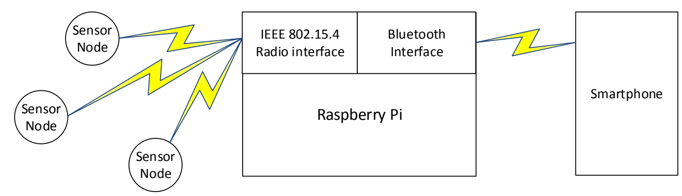

Figure 1. Organization of a Wireless Body Sensor Network with 6LoWPAN

Section II describes the implementation of the different 6LoWPAN nodes used, while section III discusses the reliability of the communications between the nodes and the border router. Section IV describes the implementation of the border router and its data collection functionality. Section V discusses the interfacing with the smartphone, and how the data is formatted. Section VI concludes this paper.

II. 6LoWPAN Nodes

The nodes are the devices containing the sensors placed on the body of the patient. These microcontroller based devices have limited computing and memory capabilities. Therefore, the nodes run a special operating system, called Contiki. Contiki is an open source operating system dedicated to the implementation of communication stacks on devices with limited capabilities, such as low memory availability or non-32 bit architectures.

978-1-4799-8802-0/15/$31.00 ©2015 IEEE

In this paper, the devices used are of two types. The first type is the Z1 platform from Zolertia. It is based on a MSP430 microcontroller, a low power 16-bit CPU, and a CC2420 radio. The second type is the CC2538EM module from Texas Instrument. It is based on a CC2538 microcontroller, which includes a Cortex-M3 core and the IEEE 802.15.4 radio necessary for 6LoWPAN. Both devices implement protocols such as I2C or SPI, and possess at least one ADC to collect data from a variety of sensors. However, in this paper, we do not focus on the acquisition of the data, but rather on its transmission to the border router. Therefore, for prototyping purpose, the device generates a fake signal such as an ECG waveform, and transmits it directly to the border router. Figure 2 shows pictures of both the modules used in this study.

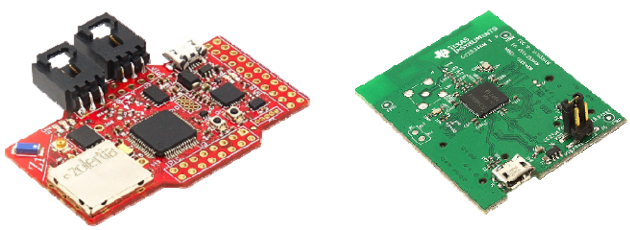

Figure 2. Nodes (devices) used in this study: On the left, the Z1 Mote. On the right, the CC2538EM Module

In order to transmit the data, the node needs to find a suitable border router, which also acts as a server. This is managed directly by the 6LoWPAN protocol. The device has the choice of either wait for a Router Advertisement (RA) message on the network, or request one by sending a Router Solicitation (RS) message. Once the border router is identified, two methods of transmission can be employed. The device can either establish a TCP connection to the router on a predetermined port, and then send the data when available, or send the data directly using UDP. In the first case, each device opens a TCP connection that ensures that the server application receives the data without missing any part and in the right order. However, in the case of the UDP, the data can be lost in the transmission and some Quality-of-Service needs to be implemented on the application level of both the nodes and the border router. It is to be noted that no encryption is used at present, since the Contiki operation system does not support it yet, but in a future implementation, the encryption must be enabled, and a method will have to be devised to share the encryption key between the different devices of the WBSN.

Different nodes in the network should be capable of providing reading from multiple sensors. Similarly, different sensors must be capable of sending different types of data depending on their function. Furthermore, the sensors must allow some parameters to be modified. Therefore, the protocol needs to define how this information will be transmitted to and from the border router. This is implemented using the first byte of the frame that has a header. The first bit of the header defines whether the frame is for configuration or data, and the remaining bits are used to quantify the amount of values in the frame. For example, when one node with 2 sensors sends its configuration to the router, it will send in the first byte the configuration flag set to 1 and the number of configurations set to 2. The configuration itself depends on the sensor: the first byte of a configuration structure defines what type of sensor it is, which impose a certain number of configuration values to be transmitted, following that code. The details of the types of sensors, and various parameters that will be transmitted are not extensively discussed in this paper.

III. Reliability and Quality-of-Service

The robustness of the design primarily depends on the reliability of each of its links, which is an integral factor in the QoS design of the Wireless Body Area Network. Reliability of the node's data corresponds to the ratio of the number of error bits received at the receiver to the number of bits generated by the sender, and is measured by Bit Error Rate (BER). BER varies according to the channel quality, node interference and temporal variation of the signal pertaining to patient movement. A sequence of packet drops could result in severe energy consumption and degrades the overall performance of the sensor nodes. We address these issues in WBAN using different approaches such as routing, contention access and reliable transport layer.

A. Routing

One of the main reasons for high BER in WBAN is the lossy channel between sensor nodes and their coordinator. Therefore, while the channel quality occasionally degrades for some nodes, other nodes can efficiently transmit their data to the coordinator. We can use multi-hop technique to mitigate the packet loss. In multi-hop routing approach, the sensor with lossy channel forwards its traffic to the coordinator, through another sensor with better channel quality. In consequence, the modules implement the *IPv6 Routing Protocol for Low-Power and Lossy Networks*, also referred as RPL. Even though this approach decreases the packet loss, it does not guarantee the full packet reliability.

B. Contention Access

One main issue with wireless communication is the different devices trying to access the radio at the same time, creating contingency. This can lead to packet loss if not properly dealt with. The IEEE 802.15.4 includes the implementation of an anti-collision algorithm to prevent such issue, however, it consumes more energy and thus reduces the battery life of the system. Another method that can be used is the IEEE 802.15.4 MAC Superframe [2]. It defines a contention access period managed by the coordinator acting as a beacon to synchronize frames. The superframe provides Guaranteed Time Slots (GTS) that helps the devices to access the radio. It also provides contention free periods, where some devices have an allocated access. Without having to check for collision, the battery life of some nodes can be improved.

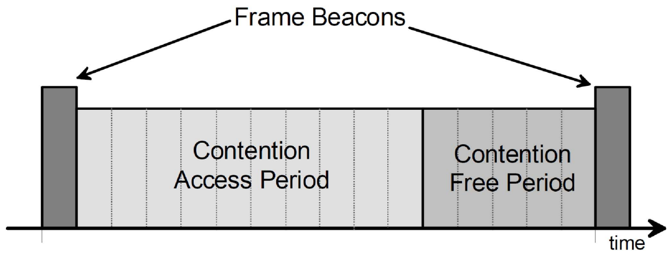

Figure 3. IEEE 802.15.4 Superframe structure with GTSs [3]

C. Reliable Transport Layer

The IPv6 protocol proposes both TCP and UDP, already widely used in IPv4 for all IP data communications. In order to be more efficient, the 6LoWPAN protocol proposes compressed version of these two protocols in order to reduce the complexity and increase the battery life of the nodes. They directly implement the process of resending the data in case the receiver does not acknowledge the transmission. However, it requires maintaining synchronization between the transmitter and the receiver. However, the UDP protocol does not implement such a process. If a packet is lost, there is no retransmission. Although simpler, it requires more program code in the application layer to implement the reliability features. This leads to a more flexible implementation that could, for example, hold the data until the next scheduled transmission, rather than directly resending the data. It is to be noted that the IEEE 802.15.4 protocol already implements an acknowledgement feature in the MAC layer that is used on some system to retransmit the data automatically.

IV. Border Router

The border router used in this study is based on a Raspberry Pi 2 board. It uses a Bluetooth dongle to connect with a smartphone, and an IEEE 802.15.4 radio module for the 6LoWPAN communications. The Linux distribution used is Raspbian, in which the kernel is recompiled to include the 6LoWPAN stack and drivers. It also includes the linux-zigbee tools. Despite the name, these tools allow the IEEE 802.15.4 radio to be used as a network interface.

Furthermore, the border router implements the server that collects the data from the nodes. It monitors any UDP or TCP connection and automatically detects new devices in the network. On the first connection of the node, the server requests information about the type of sensor connected. When the smartphone is connected, it gets a list of connected sensors. From this list, a particular sensor or a set of sensors can be selected, from which the data will be collected.

The Raspberry Pi 2 board is equipped with a 900MHz quad-core CPU, and therefore can efficiently execute signal processing algorithms. This means that despite the data collection, the server can perform the required data processing. For example, from ECG data, the server could derive the heart rate and save bandwidth on the link to the smartphone. Another improvement could be to compress the data collected from many sensors before transmitting on the single Bluetooth link.

Additionally, the border router can be equipped with Wi-Fi so that it can upload the data directly to a remote server when it is within a known network, without the use of a smartphone.

V. Bluetooth Smartphone Interface

The Bluetooth protocol is one of the wireless communication options available on all standard smartphones available today. The Bluetooth is easy to connect from one device to another, and offers multiple concurrent connections among devices [4]. Especially when forming a personal area network for the WBSN, the Bluetooth protocol is adequate to collect body sensor data to the smartphone. Generally, the Bluetooth connection between one device to another uses Serial Port Profile (SPP) which is defined to set up virtual serial ports over the Bluetooth transceivers [5]. Through this profile, binary data and framing techniques can be employed for efficient network communication. In our previous work regarding sensor data collection in the WBSN environment, the system was operating on this profile when collecting body sensor data through the Bluetooth protocol [6]. This protocol was suitable enough for real-time body sensor data transmission.

By incorporating the standard Android smartphone, we utilize another standard feature on the device to provide more efficient and simple method to establish Bluetooth connection. The Bluetooth requires a special sequence called ‘pairing’ to be able to communicate between two different devices. The pairing sequence needs to be completed by the user prior to transmitting any data through the Bluetooth protocol. However, this manual process can be altered to an automatic process by utilizing NFC tags. The NFC reader on the Android smartphone is not originally designed to play a role in the Bluetooth pairing process. By recording a simple data (Bluetooth MAC of the target to be connected to the phone) on an NFC tag, and reading this tag on the Android smartphone enables the automatic Bluetooth pairing process. In our design, the NFC tag will hold the Bluetooth MAC address of the Bluetooth transceiver installed on the border router.

Body sensor data collected through the Bluetooth can be displayed on the smartphone screen using a user-friendly interface. This helps the user to completely identify their current health status. Furthermore, the Android smartphone is interoperable in multiple communication protocols. Thus, the body sensor data can be transmitted to a distant location in real-time, via the Internet.

Since the border router is collecting data from different kinds of body sensors, the Android application software on the smartphone must be able to distinguish and identify the data types. This means that the information sent through the Bluetooth from the border router must contain additional information apart from the body sensor data. This depends on the Android application implementation, and the sensor data rules between the body sensors and the border router. On the body sensor level, additional data that contains the identity of the body sensor must be included and transmitted all the way to the Android application software. However, this data should not impact the data transmission performance over the 6LoWPAN environment, and the Bluetooth connection.

VI. Conclusion

In this paper, we presented the design of a robust framework using a combination of 6LoWPAN and Bluetooth to interface a Wireless Body Sensor Network to a smartphone. It uses a Raspberry Pi as an intermediate between the 6LoWPAN network of sensor nodes and the Bluetooth link for the smartphone, while capable of sorting, processing and compressing the data from the sensors. Furthermore, a Quality-of-Service is implemented to ensure that no data will be lost and also to efficiently manage the power consumption.

References

[1] R. Tabish, A. Ben Mnaouer, F. Touati and A. Ghaleb, "A comparative analysis of BLE and 6LoWPAN for U-HealthCare applications," *2013 7th IEEE GCC Conference and Exhibition,* pp. 286-291, 2013.

[2] Rudiyanto and R. Sari, "Analysis of the effect of beacon order and Superframe Order value to the performance of multihop wireless networks on IEEE 802.15.4 protocol," *2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS),* pp. 89-94, 2012.

[3] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements," *IEEE Std 802.15.4-2003,* Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.

[4] B. SIG, "Specification | Adopted Documents | Bluetooth Technology Special Interest Group," 2015. [Online]. Available: https://www.bluetooth.org/en-us/specification/adopted-specifications?_ga=1.157873779.873430462.1423967818. [Accessed 14 2 2015].

[5] B. SIG, "Serial Port Profile | Bluetooth Development Portal," 2015. [Online]. Available: https://developer.bluetooth.org/TechnologyOverview/Pages/SPP.aspx. [Accessed 14 2 2015].

[6] W.-J. Yi, O. Sarkar, S. Mathavan, J. Saniie, "Wearable Sensor Data Fusion for Remote Health Assessment and Fall Detection," *2014 IEEE International Conference on Electro/Information Technology,* pp. 303-307, 2014.