Towards a Reconfigurable Wireless Sensor Network for Biomedical Applications

Óscar Gama[1], Celso Figueiredo[2], Paulo Carvalho[3], P. M. Mendes[4]
[1,3]Informatics, [2,4]Industrial Electronics Dept., University of Minho, Braga, Portugal
{[1]osg, [3]pmc}@di.uminho.pt, { [2]celso.figueiredo, [4]paulo.mendes}@dei.uminho.pt

Abstract

This work presents a study on the reconfiguration requirements of wireless sensor network for biomedical applications. It is presented a wireless platform operating in a smart suit, the required monitored parameters and possible monitoring scenarios. The smart suit is currently used only for one scenario, which consists of monitoring handicapped people, with all physiological data being collected and routed to a base station. We propose using the smart suit for different scenarios, such as a patient requiring emergency medical care. The goal is to use the sensorial network in the suit to monitor his health condition, so that the cardio-respiratory rhythm, body temperature, and arterial pressure data can be used to help the diagnostic. We state that this network can be used if reconfigured correctly. This paper also shows that sensors here in use should control external high resolution ADCs using hardware interruptions instead of software interruptions.

1. Introduction

The deployment of wireless sensor networks (WSN) to assist biomedical applications has received much attention recently. These networks are expected to be widely available as soon as they become efficient and cost effective. WSNs can be used to improve quality of life if they support, for instance, glucose level monitors or general health monitors. However, there are still a few challenges to overcome. The system must be low power and the network nodes must operate under limited computation. Since these systems must operate in the human body, they also have some material constraints. Moreover, it is required continuous operation, with high robustness and fault tolerance capability. Recently, the widespread availability of low power sensor devices with physiological monitoring capacity is pushing researchers to include them in smart suits which can be used to monitor biological signals in different situations. Their monitoring application ranges from spatial suits to simple jogging activity [1].

Today, the link between textiles and electronics is more realistic than ever. E-textil is an emerging new field of research that combines the strengths and capabilities of electronics and textiles into one. E-textiles have not only wearable capabilities like any other garment, but also have local monitoring and computation, and wireless communication capabilities. Sensors and simple computational elements are embedded in e-textiles, as well as built into yarns, with the goal of gathering sensitive information, monitoring vital statistics, and sending them remotely for further processing [2].

For integration into everyday clothing, electronic components should be designed in a functional, unobtrusive, robust, small and inexpensive way. Therefore, small single-chip systems are a promising alternative to large-scale computer boxes. There are several portable units now available, which meet the requirements to be used in medical diagnostic [3]. When those monitoring device characteristics are embedded in smart suits, they become suitable for use in medical applications.

In this context, it is possible to think of a “wired” athlete [4], whose performance is being monitored, going on a critical condition that requires emergency medical care. Instead of removing all the sensors and attach new sensor devices, it is preferable to use the in-place sensor network for medical monitoring. In order to improve the application level flexibility of these networks, several solutions have been developed to allow reprogramming of WSNs. Some systems are based on network reconfiguration by simply flashing the instruction memory of the sensors. Others, based on virtual machines, inject mobile agents or scripts into the network to reprogram it. These systems, generically known as middleware, make the bridge between low level and high level abstraction, allowing the network to self-adapt to distinct application requirements [5].

Sensor network applications also require a minimum level of quality of service (QoS) over a certain period of time. This QoS requirement can change along the time, according to the application needs or the sensors availability [6]. For instance, higher quality might be required for some vital signals during a medical emergency, and lower quality might be enough for a clinical situation under control.

Most of the times, the concept of wireless smart suit is developed for monitoring individual physiological data of people during leisure activities, intensive sports training, or therapy. In this work, that concept is expanded to cover emergency medical care. To obtain the maximum benefit from the network in each scenario, usage of network reconfiguration is proposed. This work aims to obtain a smart suit with the ability to monitor critical medical parameters. The suit network may monitor the cardio-respiratory rhythm, the pulse oximetry, body temperature, arterial pressure, body goniometry, electrocardiography (ECG) and electroencephalography (EEG), depending on the clinical scenario.

2. Platform for Physiological Data Acquisition

The physiological data is measured using a smart suit. Each suit sensor sends wirely the raw data to a node that can route it directly to a central station, or pre-process it to send only the required data. Each wireless node has one or more sensors attached.

2.1. Smart Suit with Wireless Sensor Network

The smart suit is lightweight, machine washable, comfortable and easy-to-use, with embedded sensors [7]. Figure 1 shows the general concept of the platform under development. The system is based on a wireless transceiver system, which collects and digitizes the relevant data coming from different sensor types.

To measure respiratory and cardiac functions, sensors are plugged into the shirt around patient's chest and abdomen. A single channel ECG measures heart rate, the accelerometers network records patient posture and activity level, and inductive copper filaments are used for respiratory function monitoring. The monitoring electrodes are sewed in the textile material achieving a good skin contact. In addition, electrodes in the cap allow EEG monitoring.

A two-axis accelerometer senses the patient posture and activity level [7]. A network of CMOS temperature sensors distributed in the suit determines the body temperature. Also, two power-supply lines are available in all electronic regions of the suit. A 3V battery is the supply voltage. The implementation of microelectronic components into clothes and textile structures was implemented in a reliable and manufacturable way. Damage of the components by washing processes and daily use must be avoided. This solution usually requires removal of the electronics before starting the cleaning process.

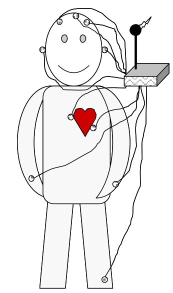

Figure 1. WSN for physiological data acquisition under medical emergency

2.2. Wireless Platform

Each smart suit described previously has embedded one MICAz mote, used for collecting and transmitting data to a base station. It is intended that the smart suits present in the nearby area form a collaborative WSN, so that the whole performance becomes superior to the summation of each element performance. Figure 2 shows the different node cores used in the system development.

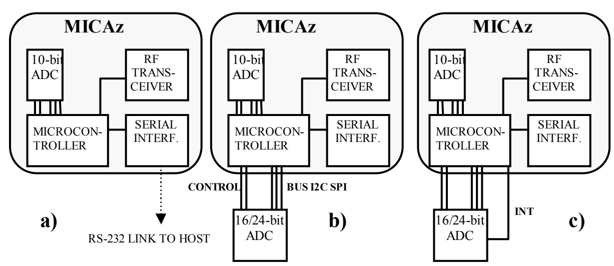

Figure 2. Wireless nodes involved in the physiological data acquisition with: a) internal ADC; b) external ADC driven by SW INT; c) external ADC driven by HW INT

The system can use three different node types. The first is the standard wireless node, which has a microcontroller with an ADC built in. This device allows a maximum data transfer rate of 115.2 kbps, imposed by the RS232 link to the host. On the other hand, the mote allows reading a maximum of 7 differential or 8 single-ended channels at 76.9 kHz with 10 bits resolution.

When more resolution is required, it is necessary to use an external ADC. This is required for high resolution EEG and ECG, for example, to enable the recording of EEG signals for brain-computer interface. It was selected the AD7714 ADC, which connects through the SPI port to the mote´s I^2C bus. This solution allows 16 or 24 bits of resolution, with a maximum sampling rate of 1028 Hz. However, the data acquisition from the external ADC, requires the use of TinyOS commands. The timings of this operating system impose a maximum rate of 1.5 kHz to sample all the channels. Since the microcontroller performs other tasks than sampling, e.g. transmitting data, this rate is not guaranteed. In this way, it was necessary to implement the third solution, which is a modified version of the previous one. Instead of implementing all the external ADC control by software, the critical tasks were implemented using hardware interruptions. Now, it is possible to sample the analogue channels safely at 1028 kHz with 16 or 24 bits resolution. The limitation comes from the ADC channel switching, done by software and taking approximately 16 µs.

3. Suitability of Network for Emergency Medical Care Assistance

The physiological data that is possible to record with the system developed are movement, cardio-respiratory rhythms, temperature, arterial pressure, ECG, and EEG. In normal operation, these data is sampled at a low rate and sent to the base station for data processing, analysis, and visualization. Moreover, the operator can select only a set of signals for detailed analysis, reducing the necessary data rate that the network needs to handle.

3.1. Wireless Platform Performance

After implementing all the solutions illustrated in Figure 2, it was possible to evaluate them in order to propose a solution to use this network for biomedical applications. The main limitation of the data acquisition systems is the RS232 link between nodes and the base station (maximum rate of 115.2 kbps). The node's microcontroller allows a data rate of 921.6 kbps, and this value can be improved. So, the data rate will be the limited by the ZigBee wireless link (maximum data rate of 250 kbps). Another problem is the impossibility of the system acquiring data when it is sending data through the serial or the wireless link.

Last but not least, each data set is encapsulated in a frame before transmission. The TinyOS frame is loosely based on the PPP in HDLC-like framing described in RFC 1662. The raw data packet is wrapped on both ends by synchronization bytes (7E character), and its header includes information such as packet type, the local and group receiver addresses and the payload length, which by default has a maximum length of 29 bytes. This value was increased to permit transmitting several samples in a single packet. A CRC value is also appended at the end of the packet. The length of the packets in a serial transmission stream may not be constant, although it is defined by a fixed C-language data structure. This happens because an escape byte is used to avoid the case when a byte inside the frame is equal to the synchronization byte or the escape character itself. This turns it easier to search a stream for packets since the 7E character always means the start or end of a packet. So, when analyzing the packet's payload data, every inserted escape character must be eliminated to recover the true data value.

The above limitations are present in all tested solutions, since the platform core is the same for all of them. Now, it is necessary to check the limitations for the solutions with extended resolution.

In the second solution, the bottleneck is the software driven interruption. To test the maximum sampling frequency, a timer was implemented and each sample was sent together with its time-stamp. We have found that the smallest sampling time, *Ts*, is about 1 ms (with a 2 MHz crystal). It is not a timer problem, since when the *Ts* was reduced bellow 1 ms, the system started to miss the deliver of some data points. In this way, the system does not allow to obtain all the signals with maximum sample rate. The third solution, with the hardware driven interruption, was able to deliver 16 bit resolution samples at a maximum data rate of 1028 Hz.

After identifying the throughput limitations of the sensing platform, it was tested to sense the desired physiological signals. During regular operation, the system samples the temperature at 1 Hz. The cardiac rhythm is obtained from an oximeter clip placed on the ear lobe, and is being sampled at 10 Hz. The respiratory rate is sampled at 5 Hz, and also each goniometric sensor is continuously sampled at 5 Hz. All sampled data is forwarded to the base station through a wireless link with a bandwidth of 128 kbps. In normal operation, the goniometric data is the most important data since what is being monitored is the physical condition evolution of the patient. In this case, since the aggregated data rate is 578 bps, the wireless network was able to deliver all the data.

Bandwidth consumption becomes more relevant when it is required to register ECG and EEG signals, even one at each time. Firstly, it was measured a two leads, single channel ECG signal. The maximum data rate was 16 kbps, which allowed for cardiac pulse evaluation. If three lead monitoring is required, it is necessary to use 48 kbps. The most challenging signal to acquire is the EEG, even if we want to monitor only the low frequency components (delta: 0-4 Hz, theta: 4-8 Hz). The problem is the need to use several electrodes, each one attached to one ADC channel. This requires 16 µs to switch between ADC channels. In this way, this sensor network is limited in the number of simultaneous EEG channels it can measure.

Notwithstanding these limitations, the WSN is able to measure several physiological signals used in clinical evaluation. The question is to understand if the WSN adaptation to a medical emergency can be done automatically based on the data being monitored, or manually by an assistant or the patient himself.

3.2. QoS on medical WSN

When a critical clinical situation occurs and emergency medical care (EMC) is required, the network paradigm changes from delivering all data being monitored to deliver only the critical data, as accurate as possible during the required time period. The data rate can also change from low to moderate or high. Moreover, the computation power is lowered to a minimum since all the data must be forwarded, in opposition to the regular operation where the cardio-respiratory rhythm can be computed on-board before sending it.

To measure the clinical situation of a patient, oximeter, EEG, and ECG sensors are used. Oximeter sensors, for measuring the blood oxygen saturation and the heart rate, are normally used at a low sampling rate, typically 0.5 Hz. Electromyography sensors sample the electrical activity of muscles typically at 1 kHz. To trace an electrocardiogram, ECG sensors collect data at a configurable rate between 120 Hz and 1 kHz. EEG sensors sample the brain activity typically at 1 kHz. Accelerometers and gyroscopes sensors may also be present in medical systems to track patient movements, being each axis typically sampled at 100 Hz.

Table 1 shows the timings associated with the hardware, software, and physiological data for both normal and emergency situations. In addition, the ADC requires some time to acquire and convert each sample, and the software takes some time to detect the data ready condition and to get it. Because it is necessary to correlate signals from multiple sensor devices, data from each node needs to be consistently time-stamped.

From Table 1, it can be observed that the network cannot serve all the sensors simultaneously. Furthermore, in a medical emergency, only a set of vital signals must be monitored in order to follow the clinical situation. The most important data requiring attention are the cardio-respiratory rhythms, the arterial pressure, and the temperature [8]. Also very important, and requiring attention in many situations, are ECG and oximetry data. The EEG can also be important, e.g. for patients with epilepsy historic.

Table 1. Sampling frequency (Hz) in normal and emergency operation

OPERATION	NORMAL	EMERGENCY
Data reading (internal ADC)	76 – 3800	-
Data reading (external ADC interrupt driven)	~1700	-
Data reading (ext. ADC by software)	~3800	-
ECG (3-12 leads)	100	120 – 1000
EEG (2-128 leads)	1 – 80	100 – 1000
Temperature	1	1
Goniometry	5	Not requested
Oximetry	-	0.5
Respiration rate	5	-
Arterial pressure	once a day	100 - 135
Cardiac rate	10	0.5

The most straightforward solution is to design a WSN for every possible scenario. However, if the network is previously configured to monitor several parameters, it is not optimized to monitor the critical ones with the required quality of service. Moreover, the emergency team may want to monitor only one parameter, which is not possible to know beforehand. When the assistant selects one physiological parameter through a scenario configuration, that parameter must be delivered.

One way to alleviate the bandwidth requirements is pre-processing data on-board. However, the algorithms required for data analysis in a medical emergency are usually complex, requiring a large amount of memory and/or high computing power, difficult to obtain with a 4 MHz microprocessor.

The solution proposed is to reconfigure the network in order to meet the diagnostic purpose. To achieve this, the network must be able to deliver the selected data for the selected emergency medical care scenario. For each mode of operation the amount of memory is limited and only the modules for the selected scenario must be in operation and loaded into platform. Since we are dealing with EMC, the WSN must meet two reconfiguration requirements: i) the network must continue to operate during reconfiguration (ideally it should be reconfigured between two data samples), and ii) once started, the reconfiguration must not fail.

The performance of any communication infra-structure, including WSNs, may be characterized in terms of traditional QoS parameters such as data lost or corrupted, transmission delay, jitter, and available bandwidth. As shown next, all these parameters are important to be considered in biomedical systems.

In medical systems, a great emphasis is placed on data availability. Although intermittent packet loss due to interference may be acceptable, persistent packet loss (due to congestion, node mobility or infrastructure unavailability) would be problematic. Medical monitoring requires a reliable data communication. Delay and jitter are two important parameters to consider when dealing with real-time streaming applications. For instance, ECG signals require a minimum sample rate (250-500 Hz) to guarantee that jitter does not affect the estimation of the R-wave fiducial point, which alters considerably the spectrum [9], and the delay should be less than 3 s for useful real time analysis by the cardiologists [10]. Anyway, a real-time telecardiology system may be implemented tuning properly the well-known TCP transport protocol, and using a reception buffer to deal with delay, jitter and retransmitted data [10].

Depending on the sensors in use, sampling rates may range from less than 1 Hz (e.g. oximeters) to 1 kHz or more (e.g. ECG, EEG), placing heavy demands on the bandwidth available at the wireless channel, especially if used with a high resolution ADC. Since the ZigBee technology, used by the motes, has a transmission rate of 250 kbps, the bandwidth problem may not arise in the link between the sensor and the base station, but rather in the link between the base stations and the monitoring sets (e.g. PDAs) used by the medical staff. Moreover, unlike many WSN applications, medical monitoring cannot use the traditional in-network aggregation since it is usually unmeaningful to merge data from several patients.

3.3. Medical WSN reconfiguration

To allow network reconfiguration, it is essential to guarantee that the network will continue operating after reconfiguration and the time spent in reconfiguration will not conflict with the data acquisition and processing. Several systems have been developed to allow sensor network reconfiguration [5]. In these systems, the new application code is flooded throughout the network or, alternatively, mobile code or agents are injected into the network, and intelligently they move or clone themselves into the desired locations based on network changes. Deluge [11] and Agilla [12] are representative examples of both paradigms. Next, and considering the performance tests carried out by team members of these systems, we will see how appropriate they are for biomedical WSNs.

Deluge, fully implemented in nesC, is a reliable data dissemination protocol for propagating large data objects from one or more source nodes to many other nodes over a multihop WSN. A performance evaluation study was carried out in a testbed with 75 Mica2 motes deployed non-uniformly in a 150’ by 100’ area. The maximum diameter of this network is about five hops, and each node contains a 7 MHz, 8-bit microcontroller CPU and a radio transceiver transmitting at 19.2 kbps.

Ten pages were injected into the testbed from a corner source. Each page is 1104 bytes (48 data packets per page, 23 bytes data payload per packet). In every test, the data was correctly and completely received by all nodes in the network at an average rate of 88.4 bytes/second. From the transmitted packets, 18% are control packets and 82% are data packets. After 10 experiments, all nodes exhibited an individual completion time greater than 220 s and less than 275 s. The completion times for an individual run were greater than 16 s and less than 21.2 s. The transmission time for sending each page through the wireless channel is about 0.5 s, negligible compared with those values. So, there will be no noticeable improvement in the completion times if the tests were performed in a higher transmission platform, like ZigBee. Since dissemination protocols cannot achieve an aggregate bandwidth near the link capacity due to the single-channel radio, spatial multiplexing, and delays necessary for suppression, it appears very hard to significantly improve these values [11].

An Agilla network is deployed without any application previously installed. Here, mobile agents are spread across nodes reprogramming effectively the network or performing application-specific tasks. An agent is either injected into the network by the user, or cloned from another agent already in the network. An agent contains its own instruction memory, data memory, program counter, operand stack, and heap. Agilla executes each agent as an autonomous virtual machine and supports multiple agents on a node, allowing multiple applications to share a network. The migration operation can be weak or strong. In a weak operation, only the code is transferred. The program counter, heap, and stack are reset and the agent resumes running from the beginning. In a strong operation, everything is transferred and the agent resumes executing where it left off. An agent can move or clone itself to any node regardless of the number of hops away. The multi-hop migration is handled by the underlying middleware and is transparent to the user. An agent dies when completes its task, freeing its resources to other agents. With a few exceptions, an instruction is one byte and an agent can have up to 440 instructions. Agilla uses acknowledgements and timers to deal with message loss during migration.

To study the evaluation performance of Agilla, a Mica2 platform was used with 25 nodes deployed uniformly in a 5x5 grid. The results suggest that the quickest an agent can migrate to the next hop is once every 0.3s, becoming naturally longer as the agent length and the multi-hop distance increases. The results also show that the probability of a message being lost increases with the distance, although there is almost no loss for distances of one or two hops [12].

Having in mind the previous explanations, Deluge seems unsuitable for reprogramming medical WSNs in critical situations because of the long delay taken by an image to be transferred over the network, increasing consequently the network energy levels. Since successful migration is essential, Agilla shows a tradeoff between latency and reliability, which may be unacceptable to reprogram real-time systems, such as ECG or EEG.

4. Conclusions

Monitoring of physiological signals is a very challenging task to be implemented with current WSNs. The challenge becomes even greater when the WSN is required to change his monitoring paradigm to assist in a medical emergency scenario. In this situation, data sources and their rate may change according to the clinical situation. To cope with this, the network must reconfigure itself to deliver all the required data and fulfill the required service quality.

Existing performance tests suggest that middleware solutions based on the typical paradigms represented by Deluge and Agilla are unsuitable for EMC due to their underlying reconfiguration timing and/or mechanism. This paper, being a contribution towards the reconfiguration of biomedical WSNs, stresses the need to include specific QoS metric and mechanisms in order to ensure proper network service quality.

Additionally, this paper also shows that if some critical biomedical application using MICAz motes requires data sampling at 1028 Hz with a resolution of 16 or 24 bits, then external ADC control interrupts must be done preferably by hardware than by software.

10. References

[1] D. Marculescu, et al. “E-textiles: ready to ware,” IEEE Spectrum, Volume 40, Issue 10, pp. 28 – 32, October, 2003.

[2] J. H. Correia, P. M. Mendes “Wireless hydrotherapy smart suit for monitoring handicapped people,” 2004 SPIE conference in Smart Materials, Nano-, and Micro-Smart Systems, pp.385-392, Sydney, Australia, 2004.

[3] H. Ren, et al. “Physiological Information Acquisition through Wireless Biomedical Sensor Networks,” Proc. 2005 IEEE Inter. Conf. on Information Acquisition, Macau, 2005.

[4] Jones, W.D, “Taking body temperature, inside out,” IEEE Spectrum, Volume 43, Issue 1, pp. 13 – 15, Jan. 2006.

[5] S. Hadim, N. Mohamed, “Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks”, IEEE Distributed Systems, Vol. 7, No. 3; March 2006.

[6] D. Chen et al. "QoS Support in Wireless Sensor Networks: A Survey", International Conf. on Wireless Networks, Las Vegas, 2004.

[7] Rocha, L.A.; Correia, J.H., “Wearable Sensor Network for Body Kinematics Monitoring,” 10th IEEE International Symposium on Wearable Computers, pp. 137 – 138, 2006.

[8] M. Paksuniemi, et al., “Wireless sensor and data transmission needs and technologies for patient monitoring in the operating room and intensive care unit”, 27th Annual Inter. Conf. of Eng.. in Medicine and Biology Society, 2005.

[9] Pinna, et al., “The accuracy of power-spectrum analysis of heart-rate variability from annotated RR list generated by Holter systems,” Physiol. Meas., nr.15: pp. 163-179, 1994.

[10] A. Iglesias, et al., “Performance Study of Real-Time ECG Transmission in Wireless Networks,” International Special Topic Conf. on Information Technology in Biomedicine (ITAB 2006), Ioannina, Grecia, 2006.

[11] A. Chlipala, et al., “Deluge: Data Dissemination for Network Reprogramming at Scale”, Proc. of 2nd Inter. Conf. on Embedded Networked Sensor Systems, pp. 81–94, 2004.

[12] Fok, et al. “Rapid development and flexible deployment of adaptive wireless sensor network applications”, Proc. of 24th Inter. Conf. on Distributed Computing Systems, 2005.