

AN AD HOC WIRELESS SENSOR NETWORK FOR TELEMEDICINE APPLICATIONS

Tarek R. Sheltami *, Ashraf S. Mahmoud, and Marwan H. Abu-Amara

Department of Computer Engineering

King Fahd University of Petroleum and Minerals

Dhahran, Saudi Arabia, 31261

الخلاصة:

لقد أدى التقدّم في أنظمة الحاسبات المُضمّنة إلى ظهور شبكاتِ المجسّات اللاسلكية التي تحتوي على بطارياتٍ صغيرة ذات إمكانات حوسبية محدودة وإرسال لاسلكي محدود. تمكن شبكات المجسّات اللاسلكية من تجميع وحساب البيانات التي سنُضمّن بعُمق في البيئة الطبيعية المراد دراستها. وعندما تستعمل شبكات المجسّات اللاسلكية المؤقتة الواسعة النطاق بين المرضى المتحركين، يُمكن أن تُزوّد الأطباء المختصين ببياناتٍ مرضى بطريقة تسمح بالمُراقبة المستمرة للمرضى بغض النظر عن الموقع، وذلك عن طريق شبكة الإنترنت أو شبكة الاتصالات الخلوية. كما يمكن هذا النظام في حالة الطوارئ، من الاتصال ألياً بالأطباء أو الممرضات والوصول إليهم من خلال أجهزتهم المحمولة. والشبكة المُقترحة تُشمل مجسّات في الطبقة الأولى وهي مسؤولة عن قياس وتجميع ثم إيصال البيانات سلكياً أو لاسلكياً إلى جهاز التحكم الصغير الذي يمثل الطبقة الثانية. إضافة إلى الطبقة الثانية توصل القراءات لاسلكياً إلى النظام المركزي. والشبكة المُطَبَّقة تُميّز بين قراءات المجسّات الدورية والحرّة حيث إنّ الأولوية تُعطى للأخيرة. وفي هذه الورقة تُطبّق ثلاث حالاتٍ خاصّة لتتبع ومراقبة المرضى والأطباء الذين يُستعملون شبكات المجسّات اللاسلكية، ذلك تحتوي هذه الورقة العلمية على تبيان لأداء الشبكة واسعة النطاق عن طريق النمذجة والتحليل الرياضي.

* Address for correspondence:

KFUPM Box 89, Dhahran 31261, Saudi Arabia

e-mail: {tarek, ashraf, marwan}@ccse.kfupm.edu.sa

Paper Received 5 October 2005; Revised April 2006; Accepted 20 December 2006.

ABSTRACT

Recent advances in embedded computing systems have led to the emergence of wireless sensor networks (SNETs), consisting of small, battery-powered “motes” with limited computation and radio communication capabilities. SNETs permit data gathering and computation to be deeply embedded in the physical environment. Large-scale ad hoc sensor networks (ASNET), when deployed among mobile patients, can provide dynamic data query architecture to allow medical specialists to monitor patients at any place via the web or cellular network. In case of an emergency, doctors and/or nurses will be contacted automatically through their handheld personal digital assistants (PDAs) or cellular phones. In specific, the proposed network consists of sensor nodes at the first layer whose responsibility is to measure, collect and communicate, via wired or wireless interface, readings to a microcontroller representing the second layer of the architecture. Deployed microcontrollers process incoming readings and report to a central system via a wireless interface. The implemented network distinguishes between periodic sensor readings and critical or event driven readings where higher priority is given for the latter. In this paper we implement 3 special cases for tracking and monitoring patients and doctors using SNETs. In addition, the performance of a large scale of our implementation has been tested by means of mathematical analysis.

Key words: Sensor nodes, sensor networks, and mobile patient.

AN AD HOC WIRELESS SENSOR NETWORK FOR TELEMEDICINE APPLICATIONS

1. INTRODUCTION

A typical sensor network consists of a large number of sensor nodes (SNODEs) that are densely deployed either inside the phenomenon or at a close proximity to it. Due to the frequent changes of the sensor network topology, the position of the SNODEs need not be engineered or pre-determined. Furthermore, in addition to being capable of sending the raw data directly to the nodes responsible for the fusion, SNODEs, with the help of the onboard processor, are also capable of locally carrying out simple computations and transmitting only the required and partially processed data. Moreover, sensor networks have self organizing and fault tolerant capabilities. Such capabilities make sensor networks suitable to many applications such as health, environmental, military, security, and home applications [1]. Note that an SNODE is equipped with a limited power [1–4] where the SNODE operation is highly dependent on battery lifetime. Therefore, power conservation and power management is very significant. The main tasks of an SNODE are events detection, data processing and data transmission. Power consumption in an SNODE can be divided into three categories: *sensing, communication and data processing*. While sensing power depends on the nature of the application, periodic updates consume less power than persistent monitoring.

As presented by Akyildiz *et al.* [1], a typical architecture for a sensor network is shown in Figure 1 and consists of the sensor field that defines the collection of the scattered SNODEs, the sink, and the task manager node. Using a multihop infrastructureless architecture, the data collected from the scattered SNODEs are routed the sink node, where it is communicated via Internet or satellite to the task manager node.

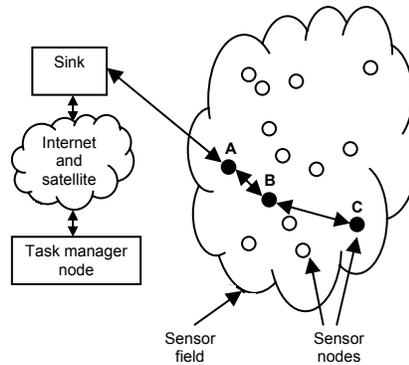


Figure 1. Sensor network architecture

In this paper we propose and present a case study of the use of an ad hoc sensor networks (ASNET) to facilitate the monitoring of the health conditions of patients by deploying SNODEs with each patient and possibly SNODEs with doctors and/or nurses. Each SNODE deployed with the patient has a specific task such as reading blood pressure, temperature, and so on. The resulting mobile patient ASNET allows the tracking and the monitoring of patients and doctors inside or outside the hospital. In the case of an emergency, doctors, and/or nurses will be contacted automatically through their handheld personal digital assistants (PDAs) or cellular phones. Upon being contacted, the doctor/nurse will in turn issue a medical query to the specific mobile patient ASNET [5–9]. More specifically, the proposed ASNET consists of sensor nodes at the first layer whose responsibility is to measure, collect, and communicate, via a wired or wireless interface, readings to a microcontroller at the second layer. Microcontrollers communicate with a third layer devices consisting of either central computer or handheld/notebooks carrier doctors/nurses. The implemented ASNET distinguishes between periodic sensor readings and critical or event driven readings where higher priority is given for the latter. The actual setup of the mobile patient ASNET can result in one of the following three configurations.

1. Communication between SNODEs of the patient and the sensor node of the doctor/nurse without a centralized unit: we assume that each patient has small and light weight sensor nodes connected to a microcontroller also attached to the patient. Doctors/nurses may also carry sensor nodes that allow other doctors to locate them within the hospital. In this case, the patient's SNODEs send the data to the SNODE of the doctor/nurse via Wi-Fi through the patient's microcontroller without the need for a central computer system if the doctor/nurse node is one hop away from the

patient's microcontroller. Otherwise, the hospital access points are responsible for relaying the information to the destination SNODE.

2. Communication between SNODEs of the patient and the sensor node of the doctor/nurse via a centralized unit: Similar to the first case, both patient and doctor/nurse have SNODEs. In this case the communication between the patient's microcontroller and the SNODE of the doctor/nurse is via the centralized unit where the patient SNODEs send their information only to the central computer via WiFi. The doctor/nurse SNODE can invoke the database directly through the web utilizing the wired or wireless network.
3. Communication between SNODEs of the patient and the non-sensor node of the doctor: The doctor's node is a mobile general packet radio service (GPRS) device. The doctor could be inside or outside the hospital. In this case, the communication is between two different networks. The mobile patient's microcontroller sends the data to the centralized system which sends a short message service (SMS) to the doctor.

2. BACKGROUND

Advanced technologies have been developed to eliminate the barriers of distance. Many telehealth programs have been formulated in many countries such as USA, Canada, and other countries where successful telemedicine programs have been implemented [10–12].

Telemedicine solutions can be extended to offer solutions even in suburban areas, where specialists are not available. In some countries, existing communication infrastructures are being used to transmit clinical data to areas where a group of specialists are available to provide advice and consultation [12,13]. One major advantage of telemedicine is that it reduces the cost of medical practice and brings expertise into remote areas [12,14].

Akay *et al.* [15] proposed an application that uses a multiuser, collaborative environment with multimodal human/machine communication in the dimensions of sight, sound, and touch. Users communicate with the application through a fusion agent by eye-tracking, speech, and microbeam pen. The conversational agent answers the user's questions and detects human-related or semantic errors and notifies the user about the results of the image analysis. However, even though channel noise is an influential factor in this application, the authors did not study it.

The study in [16] suggested a telemedicine system utilizing the Global System for Mobile communication (GSM) cellular network and the Public Service Telephone Network (PSTN) to hospital, however, the authors did not address how the data is gathered from patients. Furthermore, the application implemented in [16] provided limited data rate of 9.6 kbps. In contrast, a high bandwidth application such as the one developed in [17,18] provided wireless multimedia access to medical information in which medical data is sent to a mobile phone for management of patients. The study concluded that the system is applicable in large city hospitals for light traffic scenarios and in suburban hospital areas. The system requires a high bandwidth, which is a major concern in wireless communication. In [19,20] a telemedicine project was developed to monitor climbers on Mount Everest. A satellite link was used to monitor the location, heart rate, skin temperature, *etc.* of the climbers. The study shows the applicability of telemedicine, through satellite, to harsh environments and concluded that other applications, such as monitoring patients at their homes, are also possible. However, this solution is cost prohibitive in terms of the required traffic over the satellite links.

3. CASE STUDY

As stated in the introduction section, the objectives of the case study are to facilitate the use of both medical sensors and wireless mobile network, in health applications. The node of the doctor/nurse could be a wireless ad hoc node, a mobile phone device, or a PDA inside or outside the hospital. By facilitating communication between the patient's microcontroller and the doctor/nurse node the system is capable of:

1. Sending an urgent SMS message to the doctor/nurse node in case of critical sensors reading.
2. Sending an e-mail with the patient profile to the doctor.

The main features of our implementation are as follows:

1. The integration of the infrastructure cellular network, ad hoc network, sensor network, and IP network
2. Availability of communication anywhere via the multi-layer hierarchical structure.
3. Information is made available to experts at any location via the web.

3.1. Medical Sensors

The blood pressure sensor MS5536 shown in Figure 2 is chosen for our implementation. The MS5536 sensor is a family of high resolution factory calibrated pressure sensors. The sensor device has a very low standby current and automatically enters power down mode after each conversion. Also, it requires serial peripheral interface (SPI) programming that uses a 3-wire serial interface and is compatible with most existing microcontrollers. Positive pressure corresponds to higher pressure at port 1, which is the nozzle port on the plastic cap as shown in Figure 3.

3.2. TINI Microcontroller

Our implementation uses the TINI microcontroller shown in Figure 4. The board features a real-time clock (RTC), 1MB flash, 1MB static RAM, and support for an external Ethernet PHY for connecting to a wide variety of networks. In addition, the microcontroller has a built in OS that enables SPI java programming and a full TCP/IP v4/v6 support. Finally, the microcontroller also supports a Wi-Fi interface for wireless connectivity.



Figure 4. Tini microcontroller

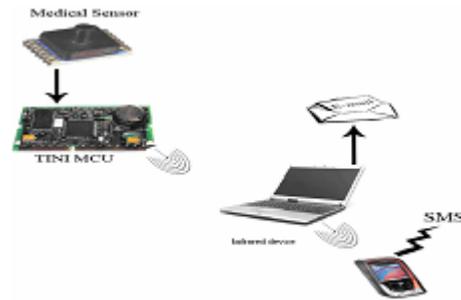


Figure 5. Case study ASNET architecture



Figure 2. Blood pressure sensor MS5536

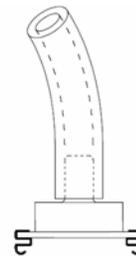


Figure 3. Port 1 of the MS5536

3.3. ASNET Architecture

The ASNET deployment scenario used in our implementation is depicted in Figure 5. As shown, SNODEs measure and communicate reading to the microcontroller in charge which in turn forwards processed reports to the central computer. The patient profile is updated with processed information in the central database. The central computer is then responsible for sending emails and/or SMS messages in cases of emergencies. At the bottom of the hierarchy the SNODEs send two types of reading: periodic messages and critical (or event driven) messages. In our implementation, critical messages are given higher priority over periodic messages. The microcontroller buffer implements this policy when reporting data to the central computer.

3.4. Software

The software component of our implementation can be divided into three main software modules, namely, the e-mail module, the SMS module, and serial peripheral interface module. For the e-mail and the SMS modules, off the-shelf java codes were used to implement the functionalities. The following describes briefly the operation of each of these modules.

3.4.1. E-Mail

In our implementation the e-mail program sends e-mails from the central PC through port 25 to the doctor's/nurse's e-mail in case of emergency and at the same time, a pop-up window will be displayed on the central computer screen. Furthermore, in case of emergency the sensor monitors the patient continuously rather than periodically.

3.4.2. SMS

In addition to the e-mail sent by the central computer, an SMS message is sent to a list of predetermined phone numbers. The system utilizes a GSM modem, or a cellular phone through an infra-red (IR) interface, to connect to the mobile network infrastructure.

3.4.3. Serial Peripheral Interface (SPI)

The MS5536 communicates with the microcontroller digital systems via a 3-wire synchronous serial interface. The serial clock (SCLK) signal initiates the communication and synchronizes the data transfer with each bit being sampled by the MS5536 on the rising edge. The data should thus be sampled by the microcontroller on the falling edge of SCLK and sent to the MS5536 with the falling edge of SCLK. The SCLK-signal is generated by the microprocessor's system. The digital data provided by the MS5536 on the "data out" (DOUT) pin, shown in Figure 6, is either the conversion result or the software calibration data. In addition the signal DOUT is also used to indicate the conversion status. The selection of the output data is done by sending the corresponding instruction on the pin "data input" (DIN).

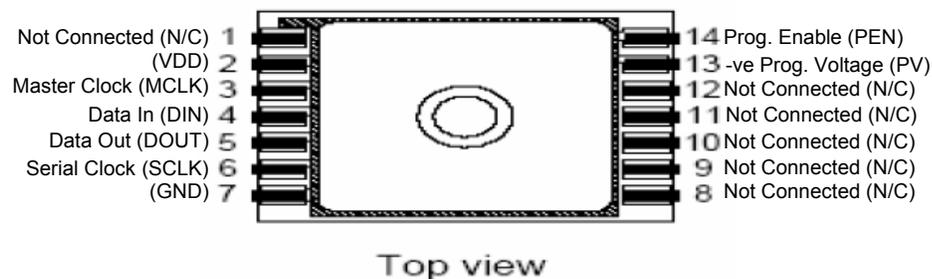


Figure 6. MS5536 pin diagram

3.4.4. SPI Code

The SPI pseudo code listed in Table 1 communicates with the sensors using the SPI protocol. The parameters needed to construct this protocol are as follows:

- **Delay:** which specifies the length of each pulse for the serial clock (SCLK). For the sensor used in our implementation the pulse width is 20 μ s.
- **CPOL:** the clock polarity that indicates which state is considered idle for the SCLK (0 or 1).
- **CPHA:** clock phase which specifies when to latch the data (rising/falling edge).

Finally, Figure 7 shows a sample output as displayed on the central computer. Note that D1 and D2 shown in the figure are two readings from the sensor used to calculate high blood pressure. While the figure focuses on the calculation of the high blood pressure, the calculation of low blood pressure is very similar. The microcontroller sends a predetermined sequence numbers to get D1 and D2.

Table 1. SPI Pseudo Code

<pre> Class spiExample{ change the variables to set the delay and time of fetching void reset (){ send the sequence 10101010101010100000 to sensor } void getW(int x){ if(x==1) \\WORD 1 send the sequence 111010101000 to sensor if(x==2) \\WORD 2 send the sequence 111010110000 to sensor if(x==3) \\WORD 3 send the sequence 111011001000 to sensor if(x==4) \\WORD 4 send the sequence 111011010000 to sensor wait 50 micro-second get the result from sensor }→ cont'd </pre>	<pre> void getD1(){ send the sequence 1111010000 to sensor wait 50 micro-second get the result from sensor } void getD2(){ send the sequence 1111001000 to sensor wait 50 micro-second get the result from sensor } calculate the pressure from D1 and D2 if (pressure ≥ threshold) send alert message to doctor's mobile else continue measuring the pressure </pre>
---	---

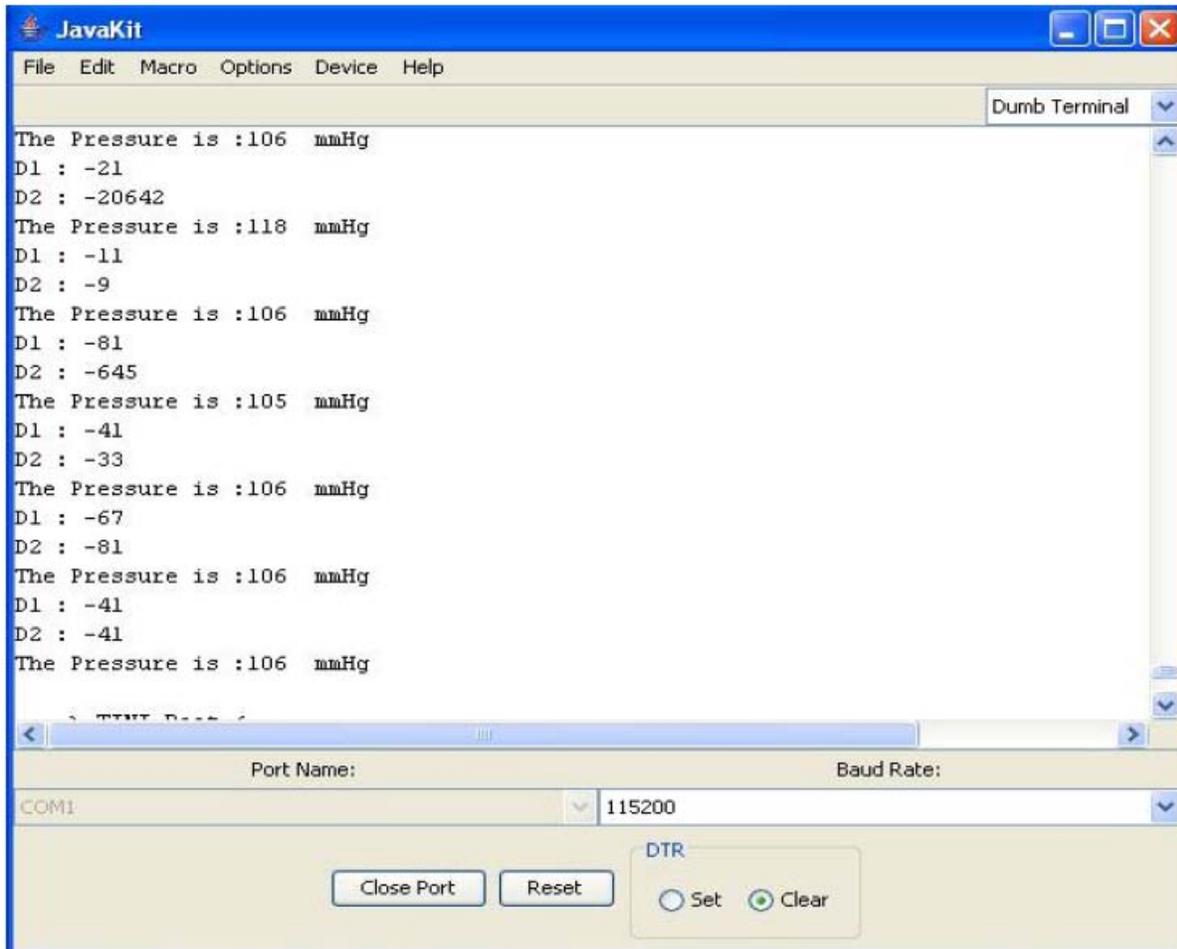


Figure 7. Sample output of SPI code

4. ANALYSIS FOR MOBILE PATIENT SENSORS NETWORK

The proposed architecture enables three communication modes as outlined in the introduction section. In the first and the third cases, signal coverage and power consumption are not issues of concern, since the ad hoc network is only for one hop distance and GSM/GPRS network is reliable and available. The analysis in this section is applied to the second communication mode where the patients' SNODEs communicate their information to the central computer via Wi-Fi. In specific, we focus on the patients' SNODEs to central computer communication link and the performance of the proposed prioritization scheme.

To this end, we model a large scale network of the proposed SNODEs and provide performance figures in terms of the expected message waiting time for both types of messages given the priority policy assumed. A schematic diagram of the assumed network is shown in Figure 8.

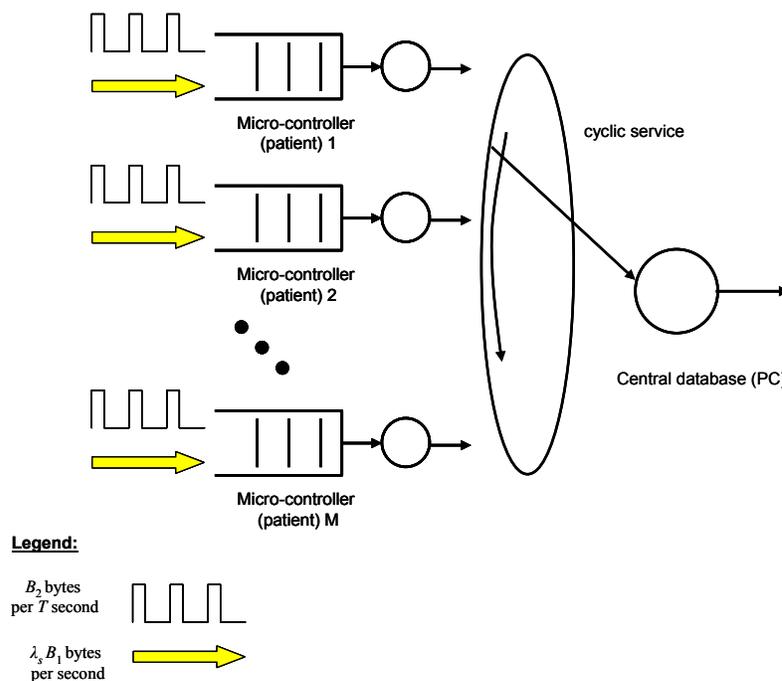


Figure 8. Model of proposed ASNET

The above model assumes the microcontroller operates as a queuing server for the SNODEs while the central server polls the group of M microcontrollers for traffic according to the exhaustive policy. Alternative service disciplines can be implemented on the link between the central system and the microcontrollers.

4.1. Assumptions

1. The i^{th} ($1 \leq i \leq M$) microcontroller (patient) is connected to N_i sensors where output messages of the j^{th} ($1 \leq j \leq N_i$) sensor are classified into two types:
 - (a) Type 1: critical messages – B_{1j} bytes generated according to a Poisson rate of λ_{sj} message per second. These messages have higher priority over type 2 messages as they represent threshold crossing messages.
 - (b) Type 2: periodic messages – B_{2j} bytes every T_j seconds.
2. A microcontroller process messages at the rate of C_m bytes per second.
3. A number of M microcontrollers are connected to the central system using IEEE802.11 wireless network.
4. The central system polls each peripheral device and processes messages at the rate of C_c bytes per second.

4.2. Analysis of Microcontroller Queuing System

The total offered load, L_i , in bytes per second for the i^{th} ($1 \leq i \leq M$) microcontroller is equal to

$$L_i = \sum_{j=1}^{N_i} (\lambda_{sj} B_{1j} + B_{2j} / T_j) .$$

Therefore, for a stable queuing system the following constraint should be satisfied:

$$C_m > L_i \quad i = 1, 2, \dots, M .$$

The bandwidth available for nondeterministic traffic is therefore given by:

$$\mu_{ms} = C_m - \sum_{j=1}^{N_i} B_{2j} / T_j \quad i = 1, 2, \dots, M .$$

Hence, the microcontroller can be modeled as an M/D/1 queue whose service rate is given by μ_{ms} while the arrival rate, λ , is equal to

$$\lambda = \sum_{j=1}^{N_i} \lambda_{sj} B_{1j}$$

For example, the mean message waiting time for the M/D/1 queue is given by

$$E[W] = \frac{\rho}{2(1-\rho)} E[\tau]$$

where the traffic intensity, ρ , is given by $\lambda / \mu_{ms} = \frac{\sum_{j=1}^{N_i} \lambda_{sj} B_{1j}}{C_m - \sum_{j=1}^{N_i} B_{2j} / T_j}$ and the mean service time, $E[\tau]$ is equal to

$$\frac{E[B_{1j}]}{C_m - \sum_{j=1}^{N_i} B_{2j} / T_j} \text{ in seconds.}$$

Note the above analysis is simplified and does not assume priority for critical messages over periodic messages at this stage.

4.3. Analysis of the Polling Network

Assuming a cyclic exhaustive service over the wireless LAN interfaces, the central server polls each of the microcontroller cyclically and serves all buffer contents. The analysis below is divided into two steps:

1. Computation of the mean cycle time, T_c .
2. Computation of the mean waiting time for messages assuming priority is given to critical messages over periodic messages.

To compute the mean cycle time we use the following argument. For steady state, the mean cycle length, T_c should be equal to

$$T_c = \sum_{i=1}^M (E[X_i] + \delta_i)$$

where $E[X_i]$ is mean service time for all messages found in the i^{th} microcontroller buffer while δ_i is the central service switching time from the i^{th} station to the $(i+1)^{\text{th}}$ station. Let the mean number of type-1 messages and type-2 messages that accumulate during a mean cycle time, T_c be equal to $E[n_1]$ and $E[n_2]$, respectively. It is easy to note that:

$E[n_1] = T_c \sum_{j=1}^{N_i} \lambda_{sj}$ and $E[n_2] = \sum_{j=1}^{N_i} T_c / T_j$. Therefore, the mean cycle time should be given by

$$T_c = \frac{\sum_{i=1}^M \delta_i}{1 - \frac{M}{C_c} \left[\sum_{j=1}^{N_i} \lambda_{sj} B_{1j} + \sum_{j=1}^{M_i} B_{2j} / T_j \right]}$$

Let us denote the traffic intensity due type-1 messages by $\rho_1 = \frac{M}{C_c} \sum_{j=1}^{N_i} \lambda_{sj} B_{1j}$, and the traffic intensity due to type-2 messages by $\rho_2 = \frac{M}{C_c} \sum_{j=1}^{N_i} B_{2j} / T_j$.

To obtain a closed form for the mean message waiting time in buffer, we simplify the analysis by assuming symmetric loading on microcontroller boards. For the general asymmetric case, numerical solutions for the mean message waiting times are still possible as outlined in [21]. For the symmetric case, let $N = N_i$, $B_1 = E[B_{1j}]$, $B_2 = E[B_{2j}]$, and $T = E[T_j]$. This translates to mean cycle time that is given by

$$T_c = \frac{\sum_{i=1}^M \delta_i}{1 - \frac{MN}{C_c} [\lambda_s B_1 + B_2 / T]}$$

where now the traffic intensities due to type-1 and type-2 messages are given by $\rho_1 = MN \lambda_s B_1 / C_c$ and $\rho_2 = MN B_2 / (T C_c)$.

Using the above notations, the mean message waiting time of type- k , $E[W_k]$, ($k=1$ or 2) is equal the mean waiting time while other buffers are being served, $E[W_a]$, plus the mean waiting time while messages within the i^{th} buffer are being served, $E[W_{bk}]$. The former component is simply half the mean intervisit time for the i^{th} station while the latter is obtained using the M/G/1 analysis assuming a non-preemptive priority-based service for the two classes of traffic. That is,

$$E[W_k] = E[W_a] + E[W_{bk}] \quad k = 1, 2$$

Messages of both types experience the same $E[W_a]$, while the second component is different because of the priority scheme. The first component of mean message waiting time is given by

$$E[W_a] = [1 - (\rho_1 + \rho_2)] T_c / 2$$

To compute the second component of message waiting time we resort to the following approach. Using the M/G/1 as a reference, the mean waiting of type-1 messages is given by

$$E[W_{b1}] = \frac{E[R]}{1 - \rho_1}$$

while the mean waiting time of type-2 messages is given by

$$E[W_{b2}] = \frac{E[R]}{(1 - \rho_1)(1 - \rho_1 - \rho_2)}$$

where $E[R]$ is the mean residual service time for the customer in service. The mean residual service time when the server is busy is given by $E[R / busy] = E[\tau^2] / (2E[\tau])$, and therefore, using the above definitions for ρ_1 and ρ_2 one can show that the mean residual time is equal to

$$E[R] = \frac{MN [B_1 \lambda_s + B_2/T]^2}{2C_c^2 (\lambda_s + 1/T)}$$

substituting in the $E[W_{bk}]$ equation yields the required delay figure.

4.4. Numerical Example

Assuming a DSS IEEE802.11 setup where channel rate is 1 Mb/s. The polling overhead ($T_{poll} + T_{ack} + 2 \text{ SIFS}$) is equal to 0.72 ms. Furthermore, assume the sensor's payloads for periodic and critical messages are both equal to 1 kbyte, *i.e.* $B_1 = B_2 = 1024$ bytes. The number of sensor nodes connected to each microcontroller, N , is given by 10.

As derived in the previous section, the total traffic intensity is given by $\rho = \frac{MN}{C_s} (\lambda_s B_1 + B_2/T)$. Figure 9 depicts

typical message waiting times for both types of traffic versus the network traffic intensity. It is clear that critical messages experience less waiting time compared to periodic messages, which complies with our priority scheme. Using our setup, the second component of the message waiting time (*i.e.* that due to queuing) is negligible compared to the intervisit time for most of the operating region in terms of traffic intensity. This is due to the low offered load by the SNODEs in terms of volumes (*i.e.* small B_1 or B_2) or in terms of frequency of arrival (*i.e.* low λ_s or $1/T_i$) compared to the speed of the communication channel. For most of the operating range of traffic intensity, the message waiting time is linear with the number of patients or microcontroller devices.

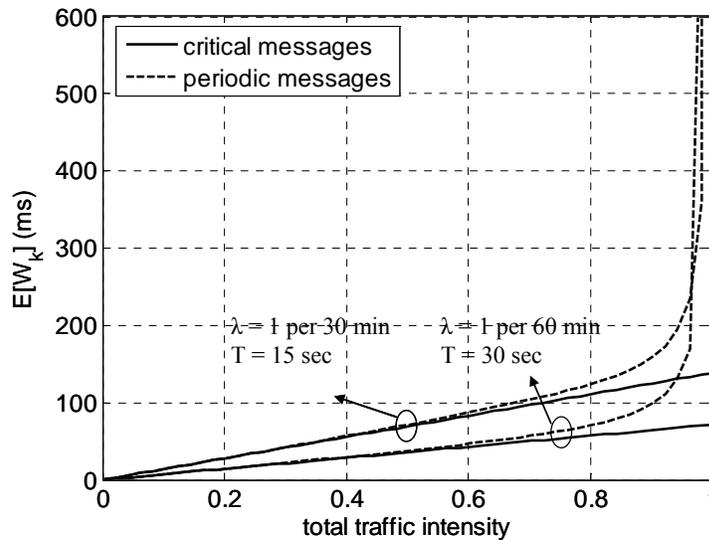


Figure 9. Message waiting time

5. CONCLUSIONS

In this paper we have explored three telemedicine scenarios where ASNET is utilized to communicate medical reading of patients to a central system through the intervention of middle hierarchy devices. In our implementation microcontrollers were used as the second layer. The proposed architecture that consists of three layers was implemented using off-shelf components and packages. An SPI-based algorithm was developed for communication between the sensor nodes and the microcontroller. A wireless LAN network has been created to connect microcontrollers to the central system and handhelds/notebooks carried by doctors/nurses. The paper also provides a model that quantifies the expected performance of the network. A priority-based polling scheme between the central server and the group of mobile patients has been modeled and analyzed. The preliminary results indicate that the priority scheme manages to keep the mean waiting time for critical message low compared to periodic messages sent by the SNODEs. However, the mean waiting time, for the selected implementation, is dominated by the polling intervisit time and not greatly affected by the queuing time experienced in microcontroller buffer.

6. REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, **38(4)**, (2002), pp. 393–422.
- [2] A. S. Porret, T. Melly, C. C. Enz, and E. A. Vittoz, "A Low-power Low-Voltage Transceiver Architecture Suitable for Wireless Distributed Sensors Network," in *Proc. IEEE International Symposium on Circuits and System*, May 2000, vol. 1, pp. 56–59.
- [3] L. Li and J. Y. Halpern, "Minimum-Energy Mobile Wireless Networks Revisited," in *Proc. IEEE International Conference on Communications*, June 2001, vol. 1, pp. 278–283.
- [4] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy, "PicoRadio Supports Ad Hoc Ultra-low power Wireless Networking," *IEEE Computer Magazine*, **33(7)** (2000), pp. 42–48.
- [5] F. Hu, J. Tillett, J. Ziobro, and N. K. Sharma, "Secure Tree-Zone-Based Wireless Sensor Networks for Telemedicine Applications," in *Proc. of IEEE GLOBECOM*, 2003, pp. 345–349.
- [6] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks," in *Proc. MOBICOM*, September 2002, pp. 12–23.
- [7] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad hoc Networks," *Mobile Computing and Communications Review*, **7(1)** (2003), pp. 193–209.
- [8] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring Behavior in Home Using A Smart Fall Sensor," in *Proc. IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, October 2000, pp. 607–610.
- [9] M. Ogawa, T. Tamura, and T. Togawa, "Fully Automated Biosignal Acquisition in Daily Routine through 1 Month," in *Proc. 20th Annual International Conference IEEE-EMBS*, October 1998, vol. 4, pp. 1947–1950.
- [10] V. R. Singh, "A Strain Gauge Jaw Force Transducer," in *Biotelemetry III*. New York: Academic Press Inc., 1976, pp. 263–266.
- [11] P. A. Jennett, R. E. Scott, L. A. Hall, D. Hailey, A. Ohinmaa, C. Anderson, R. Thomas, B. Young, and D. Lorenzetti, "Policy Implications Associated with the Socio-Economic and Health System Impact of Telehealth: A Case Study from Canada," *Telemedicine Journal and e-Health, Special Canadian Issue*, **10(1)** (2004), Spring (2004), pp. 77–85.
- [12] K. Singh, "Biotelemetry: Could Technological Developments Assist Healthcare in Rural India," *Rural and Remote Health*, **5(2)** (2005), article no. 234.
- [13] D. C. Jeutteur, "Telemetry of Biomedical Signals," *Medical & Life Sciences Engineering*, **13** (1994), pp. 131–145.
- [14] M. Burrow, "A Telemedicine Tested for Developing and Evaluating Telerobotic Tools for Rural Health Care," in *Medicine Meets Virtual Reality II: Interactive Technology and Healthcare, Visionary Applications for Simulation Visualization Robotics*. City: Aligned Management, 1994, pp. 15–18.
- [15] M. Akay, I. Marsic, A. Medl, and G. Bu, "A System for Medical Consultation and Education Using Multimodal Human/Machine Communication," *IEEE Transactions on Information Technology in Biomedicine*, **2(4)** (1998), pp. 282 – 291.
- [16] B. Woodward and M. F. A. Rasid, "Wireless Telemedicine: The Next Step?," in *Proc. 4th Annual International IEEE-EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, April 2003, pp.43–46.
- [17] A. Rodriguez, J. L. Villalar, M. T. Arredondo, M. F. Cabrera, and F. Del Pozo, "Transmission Trials with a Support System for the Treatment of Cardiac Arrest Outside Hospital," *Journal of Telemedicine and Telecare*, **7(1)** (2001), pp. 60–62, 2001.
- [18] J. Bates, B. Demuth, V. Pendleton, and K. P. Tonkin, "Wireless Applications in Mobile Telemedicine," *Telemedicine Journal and e-Health*, **8(2)** (2002), pp. 137.

- [19] R. Satava, P. B. Angood, B. Harnett, C. Macedonia, and R. Merrell, "The Physiologic Cipher at Altitude: Telemedicine and Real-Time Monitoring of Climbers on Mount Everest," *Telemedicine Journal and e-Health*, **6(3)** (2000), pp. 303–313.
- [20] P. B. Angood, R. Satava, C. Doarn, and R. Merrell, "Telemedicine at the Top of the World: the 1998 and 1999 Everest Extreme Expeditions," *Telemedicine Journal and e-Health*, **6(3)** (2000), pp 315–325.
- [21] H. Takagi, "Queueing Analysis of Polling Models", *ACM Computing Surveys*, **20(1)** (1988), pp. 5–28.