Mobile Netw Appl
DOI 10.1007/s11036-010-0253-7

Transmission of Patient Vital Signs Using Wireless Body Area Networks

Baozhi Chen · Dario Pompili

Abstract Triage is the process of prioritizing patients based on the severity of their condition when resources are insufficient. Hospitals today are equipped with more and more electronic medical devices. This results in possibly high level of electromagnetic interference that may lead to the failure of medical monitoring devices. Moreover, a patient is usually moved between different hospital settings during triage. Accurate and quick prioritization of patient vital signs under such environment is crucial for making efficient and real-time decisions. In this article, a novel in-network solution to prioritize the transmission of patient vital signs using wireless body area networks is proposed; the solution relies on a distributed priority scheduling strategy based on the current patient condition and on the vital sign end-to-end delay/reliability requirement. The proposed solution was implemented in TinyOS and its performance was tested in a real scenario.

Keywords body area networks · ubiquitous networking · wireless sensor networks · cross-layer design · quality of service · prioritization · healthcare

B. Chen (✉) · D. Pompili
Department of Electrical and Computer Engineering,
Rutgers University, CoRE Building (Room 629),
96 Frelinghuysen Road, Piscataway,
NJ 08854, USA
e-mail: baozhi_chen@cac.rutgers.edu

D. Pompili
e-mail: pompili@ece.rutgers.edu

1 Introduction

The rapid growth of wireless technologies and personal area networks enables the continuous healthcare monitoring of mobile patients using compact sensors that collect and evaluate body parameters and movements. These sensors, limited in memory, energy, computation, and communication capabilities, are strategically deployed on a patient, forming a cluster that is called Wireless Body Area Network (WBAN) [21]. Many works based on WBANs have been proposed that focus on designing wireless sensors for a single vital sign and on developing pervasive healthcare systems to monitor vital signs of multiple patients.

In this article, we focus on providing a networking solution for *in-hospital triage*, which is the process of prioritizing patients based on the severity of their condition. This process facilitates the ability of the medical team to treat as many patients as possible when resources are insufficient for all to be treated immediately. Existing devices for monitoring patient vital signs are mostly wired, often depend on direct user interaction, have a limited analytic capability, require manual archiving even of digital data sources, and have limited capability to propagate data to the next destination on the patient's path. This is particularly critical in the in-hospital setting.

Accurate and reliable monitoring of patient's vital signs during this period is crucial for making efficient and error-free triage decisions. During triage, emergency service providers need to rapidly assess the injured patient and determine the need for trauma center care. In addition to challenges of acquiring patient data, trauma triage is now limited by a reliance on human interpretation of acquired patient data. Using existing

Published online: 06 July 2010

technology, the in-hospital environment lacks effective methods for prioritizing information streams, evaluating time-dependent trends, managing incomplete data, and providing effective alerts. Current limitations of patient monitoring represent an important barrier for developing improved trauma triage methods.

On the other hand, wireless technology has experienced explosive growth over the past decade. Wireless Wide Area Networks (WWANs), mobile phone networks, and Wireless Local Area Networks (WLAN) are being deployed almost everywhere at an incredible speed, resulting in increased spectrum use by a variety of heterogeneous devices, standards, and applications. This holds especially true for the unlicensed Industrial, Scientific, and Medical (ISM) bands that host a number of heterogeneous networks. For example, many devices of different standards for WLANs such as Bluetooth, ZigBee, IEEE 802.11b/g all operate in the same 2.4 GHz ISM frequency band.

Because radio waves centered at the same frequency emitted from these devices interfere with each other, coexistence of them has become an important issue in order to ensure that wireless services can maintain their desired performance requirements. For instance, in a critical environment such as medical emergency scenarios it is extremely important to avoid the failure of the medical devices that may be caused by radio frequency interference.

With an ever-increasing use of electronics in medical devices of all kinds as well as many wireless communication devices in medical environments, some unforeseen problems are coming: the interactions between the products emitting the electromagnetic (EM) energy and sensitive medical devices. Even the devices themselves can emit EM energy, which can react with other devices or products. It has been reported that medical devices may fail to operate correctly due to the existence of electromagnetic interference [25].

To guarantee wireless services in such environments, it is necessary to design a system that can handle such interference. Existing research on wireless healthcare systems has focused on the design of purpose-specific one-BAN system [3, 5] (i.e., system used on one patient), in-BAN data processing/fusion [16, 36], and improvement of network performance metrics such as throughput and energy efficiency [15, 31]. In addition, emergency services have been considered in [17, 30]. While these studies have proposed solutions to access patient healthcare data in real time, no research has focused on prioritizing the transmission of healthcare data over the wireless network under the electromagnetic interference (EMI) environment by *jointly* considering the patient condition *and* the data content, i.e., the type of measurement (temperature, O_2 saturation, blood pressure and pulse, heart rate variability, etc.), which is critical in emergency services.

Furthermore, different types of healthcare data are of different importance during in-hospital triage, as shown in Table 1. In an in-hospital environment, proper prioritization of critical vital signs is crucial for efficient and real-time triage. Moreover, patients in different emergency conditions have different service requirements: a patient who needs more immediate service should be given higher priority, i.e., his/her vital signs should be transmitted with higher end-to-end (e2e) reliability and lower delay.

In order to provide a networking solution that performs real-time in-hospital triage on multiple patients under EMI environment, in this work we propose a new interference-aware WBAN system that continuously monitors vital signs of multiple patients and prioritizes data transmission based on patient's condition and data content. Based on the patient's condition, which in this work we assume to be already diagnosed, patients are categorized into three classes, "*Red*", "*Yellow*", and "*Green*", each indicating the level of treatment

Table 1 Bit rate and delay requirements of healthcare data [11]

Data source	Bit rate [bps]	Delay [s]	Sampling rate [Hz]
Electrocardiogram (ECG)	1–8k	<10	63–500
Blood pressure [mmHg]			
Arterial line	1k	10–30	63
CVP (central venous catheter)	1k	>120	63
Non-invasive cuff	0.05	30–120	0.025
Cardiac output [L/min]	1k	<10	63
Pulse oximeter SpO_2 saturation [%]	1k	<10	63
Patient ID band	0.05	>120	0.0002
Inter-cranial brain pressure [mmHg]	16	10–30	1
CO_2 Concentration (for respiration monitoring) [ppm]	1k	30–120	63
Temperature [°C]	0.3	>120	0.02

needed, i.e., *"immediate"*, *"delayed"*, and *"minimal"*, respectively. Categorization of one patient into one of these three classes can be achieved by using data aggregation algorithms within one BAN. Moreover, these algorithms should jointly consider all the vital signs taken from one patient as using only few vital signs is generally not sufficient to perform the diagnosis of a patient.

For example, injury severity assessment of neurological status (e.g., level of consciousness and motor activity) should be made by looking at both vital signs such as pulse, blood pressure, and respiratory rate, as well as the movement activities (normal vs. abnormal). We have proposed a solution to classify the movements of a patient using multiple sets of triaxial accelerometers (using IMote2 and Shimmer sensors) attached to different parts of the body in [29], which will be further extended for patient status classifications.

In this work, we focus on the e2e transmission of vital signs instead of patient status classifications. Specifically, we aim at maximizing the e2e reliability of these three traffic classes while meeting the delay requirements. Here *reliability* is defined as the ratio of the number of received data packets (containing vital sign information) at the sink and the number of total packets sent from a source node.

Our solution focuses on a wireless communication network with *light* and *moderate congestion*, as categorized in Table 2, where γ_R, γ_Y, and γ_G are the current reliability of "red", "yellow", and "green" patients, respectively. When the network is in *no congestion* status, in fact, the use of standard protocols is enough to guarantee the services to all patients. On the other hand, when the network is in *heavy congestion* status, additional mechanisms are needed to guarantee the service to patients under critical conditions. For example, transmission of vital signs from patients in non-critical conditions can be held on until the congestion becomes less severe (source rate control). Moreover, adaptive sampling techniques consisting in reducing the sampling rates of the sensors deployed on patients under non-critical conditions can be applied in order to reduce the traffic. Once the overall traffic is reduced, our solution, which is tailored for light or moderate congestion, can be applied to guarantee e2e Quality Of Service (QoS). Therefore, we focus on providing a solution for light and moderate congestion; in these states, using our communication solution, the services to the patients under critical conditions can be guaranteed.

Table 2 Network congestion types

Congestion type	Network condition
No congestion	$\gamma_R = \gamma_Y = \gamma_G = 1$
Light congestion	$\gamma_R = \gamma_Y = 1, \gamma_G < 1$
Moderate congestion	$\gamma_R = 1, \gamma_Y < 1, \gamma_G < 1$
Heavy congestion	$\gamma_R < 1, \gamma_Y < 1, \gamma_G < 1$

To sum up, our major objectives are:

1. Provide a communication solution for in-hospital networks with *light* and *moderate congestion*.
2. Maximize the reliability for all three classes of traffic while guaranteeing their e2e delay requirements.
3. Provide partial cognitive radio [19] capability to sensors in order to avoid EMI.

In order to achieve these objectives, a cross-layer communication solution is proposed to offer a prioritization service and maximize reliability while meeting the e2e delay requirement based on the patient's condition and data content. Our solution adopts a *modular design*: the modules include *Medium Access Control (MAC)*, *Routing*, and *Scheduling*. Each module is individually designed to meet the domain-specific requirements; then, the three modules are *jointly* optimized considered to obtain the best performance possible. The quality of multiple channels is considered in the MAC and routing modules, which leads to the interference-aware design of Multi-channel Quality-based MAC (MQ-MAC) and Channel Quality Based Routing (CQBR). Moreover, a two-level data packet scheduling scheme is proposed to maximize the reliability for all three classes of traffic while guaranteeing their e2e delay requirements. These modules are also designed to be of low complexity so that resource-limited sensors can run them.

Note that we aim at providing a solution to situations where the traffic is near to the network capacity. Our solution is based on the well-known Crossbow's wireless sensors IMote2/TelosB, which use the IEEE 802.15.4/ZigBee standard [10]. To improve the network performance, our solution can be easily migrated to other high speed wireless platforms such as 802.11b/g/n.

The remaining of this article is organized as follows. In Section 2, we review the related work for wireless healthcare networks and related scheduling algorithms. In Section 3, the network system that our solution is based on is presented. Our interference-aware prioritization solution is proposed and detailed in Section 4. In Section 5, performance evaluation and analysis are carried out, while conclusions are discussed in Section 6.

2 Related work

In this section, we review the related work with a special focus on three aspects: research in healthcare, multi-channel MAC protocols, and packet scheduling algorithms.

2.1 Wireless healthcare systems

Research on wireless healthcare systems can be divided into three classes: *application-specific sensor system design* [3, 5], *data fusion* [16, 36], and *communication performance improvement* [15] for healthcare. Although comprehensive, these studies are primarily concerned with the design of one BAN (one patient) only, and only few of them have examined wireless healthcare networks that consist of multiple patients or BANs [17, 31]. As an example for multi-patient monitoring, CodeBlue [17] provides an emergency medical care network substrate. However, no prioritization services of the vital signs are provided, e.g., data from patients that need to be treated immediately and data from patients that are not severely injured are transmitted using a best-effort approach. This undifferentiated service cannot satisfy the different need for emergency triage when the network has limited resources. Our solution is complementary to previous solutions and provides a prioritization service based on the patient condition and healthcare data content while guaranteing the e2e delay requirement of the healthcare data for those patients in critical situation. In this way, patients who need immediate medical treatment will be served with higher priority and critical vital signs will be transmitted faster to the sink/base station and with higher reliability. To the best of our knowledge, our solution is the first one that takes the patient's condition and data content into account in wireless healthcare networks.

In [14], MEDiSN, a system for vital sign monitoring of ambulatory patients, is proposed and evaluated using empirical experiments in the emergency room. Collection Tree Protocol (CTP), a routing protocol provided by TinyOS, is used to forward vital signs to the gateway. Nodes are divided into two classes with one class of nodes called Physiological Monitors (PMs) and the other class of nodes called Relay Points (RPs): these two classes of nodes form a two-level architecture. PMs are used to first sense the physiological data and then forward to RPs; whereas RPs are dedicated to relaying packets between PMs and the gateway (sink). Because PMs are designed to not relay packets on behalf of other nodes, patients are restricted to stay within the area covered by the RPs. Therefore, limited mobility is supported. Moreover the different roles of PMs and RPs need to be pre-specified at deployment. This is not required in our solution, where sensors deployed on the patients can self-organize into a two-tier architecture. Nodes with higher computing power, battery capacity, and networking capabilities are selected as cluster heads, which can aggregate vital signs of one patients. Cluster heads can also relay packets coming from other patients so an external infrastructure (or backbone) is not necessary; hence, seamless mobility can be supported. Finally, CTP uses only one frequency channel and route selection is based only on the link quality indicator (LQI). The authors compared the performance of CTP using two different frequencies: it is shown that CTP using the frequency channel with less interference—and hence higher LQI—provides better performance than that using the frequency channel with more interference. This shows that a solution that can select a frequency channel with higher LQI is preferable. Similar to CodeBlue, the prioritization of different vital signs is not considered, which may not satisfy the QoS requirements of different types of data.

2.2 Multi-channel MAC Protocols

Multi-channel MAC protocols (several of which are summarized and compared in [20, 33, 34]) have been proposed to increase throughput and to reduce signal interference. Depending on the number of transceivers in use, the multi-channel MAC protocols are divided into two classes: those with only one transceiver and those with multiple (≥ 2) transceivers. Performance of the second class is generally better than that of the first class due to the ability of simultaneously receiving and transmitting packets, and the ability of receiving multiple packets; however, this is achieved at the price of higher hardware complexity and cost. Conversely, our solution is designed to be practical and to run smoothly on existing low-cost ZigBee devices, which have only one transceiver. Therefore, our MAC approach falls in the first class; for this reason, in the following we focus on multi-channel MAC protocols using only one transceiver.

Receiver-Initiated Channel-Hopping with Dual Polling (RICH-DP) [28] is a receiver-initiated collision-avoidance protocol that does not require carrier sensing or the unique code assignment for collision-free reception. All nodes in a network follow a common channel hopping sequence for communications.

Unlike RICH-DP, Multi-channel MAC (MMAC) [27] uses a default channel for traffic indication and incorporate an energy-saving mechanism. The multi-channel hidden terminal problem is solved by using temporal synchronization. Time is divided into fixed-time intervals using beacons and have a small window at the start of each interval to indicate traffic and negotiate channels.

Like RICH-DP, Slotted Seeded Channel Hopping (SSCH) protocol [1] employs a channel-hopping scheme. But, unlike in RICH-DP, channel hopping is not only used for control but also for data transmissions. Scheduling packets are employed to arrange hopping schedule so that communications do not interfere with each other, while synchronization techniques are employed to assign traffic to different channels.

Multi-channel MAC (McMAC) [26] is proposed to avoid control channel congestion so that it can use a large number of channels efficiently. Each node-specific MAC address is used as a seed to generate hopping sequence. Consequently, the receiver's hopping pattern can be predicted and different node pairs are able to rendezvous at the same time on multiple channels and communicate with each other. Compared to SSCH, the hopping pattern is chosen at random and careful pairwise scheduling is not needed. Also, unlike in SSCH, network-wide synchronization is not required.

Most of the above MAC protocol solutions are designed to avoid contention or collision without considering channel fading. Although Opportunistic Multiradio MAC (OMMAC) [2] accounts for channel fading, it is evaluated only by simulations and it is not clear how well it would perform in a real environment. These protocols do not take the actual channel quality measurements into account and, hence, cannot avoid using channels hat may be experiencing severe interference or fading. Therefore, the performance of these MAC protocols is not optimized. There are also some proposals such as [32, 37] using the RSSI value to optimize multi-channel communication. However, RSSI only characterizes received signal energy, which does not capture the link characteristics such as channel quality, reliability, and space and time coherence as our solution does.

2.3 Packet scheduling algorithms

In data networks, packet scheduling has been used to ensure QoS as a way to control packet delays. Of all the scheduling techniques, two classes of scheduling are of great importance in data networks: Generalized Processor Sharing (GPS) [22, 23] (also called Weighted Fair Queuing (WFQ) [6]) and Earliest Deadline First (EDF) scheduling [4]. GPS schedules packets of each flow with guaranteed minimum bandwidth according to specified weights. It has been shown that e2e delay requirements can be mapped into a bandwidth allocation problem by appropriate admission control [23]. It has also been shown that the close coupling between delay and rate under GPS in deterministic delay bounds leads to sub-optimal performance and decreased network utilizations [7].

Conversely, an EDF scheduler assigns "deadlines" to packets arriving at the scheduler and then serves the packets in the ascending order of their assigned deadlines. Specifically, every time a packet arrives at one of the queues it is assigned a deadline equal to its arrival time plus the maximum tolerable queuing delay of the packets. Every packet needs to be sorted according to its deadline upon its arrival at the node; the packet with the least deadline is then served first. It has been shown that optimal performance can be obtained with EDF policy for a single switch [4], and certain EDF techniques can achieve better performance than those using GPS [7]. Yet, the implementation of an EDF server is more complicated than that of a GPS one, and proper techniques are needed to make the cost of such a server affordable in practice.

Several hybrid schedulers that combine EDF and WFQ (such as [8] and [38]) are also proposed for IEEE 802.16 (also called WiMax) networks. They apply only either EDF or WFQ but not both to a flow, depending on the traffic QoS requirement. For example, in [38] EDF is only applied to real-time Polling Services (rtPS) (for real-time variant bit rate flows) while WFQ is only applied to the non-real-time Polling Services (nrtPS) (for non-real-time flows that require better than best effort service, e.g., bandwidth-intensive file transfer). Similarly, EDF is only applied to rtPS while WFQ is only applied to nrtPS and Best Effort (BE) service (for best effort traffic such as HTTP with no QoS requirements). These hybrid schedulers cannot satisfy the need of vital sign traffic, which requires both class-based service depending on patient's condition and the flow-based service to guarantee the e2e delay requirements, as detailed in Section 4.4.

3 Proposed network scenario

Our scenario is assumed to be a network in the hospital with large enough triage area and limited number of base stations to collect vital sign data; in such a scenario multihop communication would be beneficial to improve the connectivity and to accommodate more traffic. The whole networking system is assumed to be a *two-tier hierarchical architecture* (Fig. 1) with the high-level tier being the network of BANs (inter-BAN) and the low-level tier being the networks

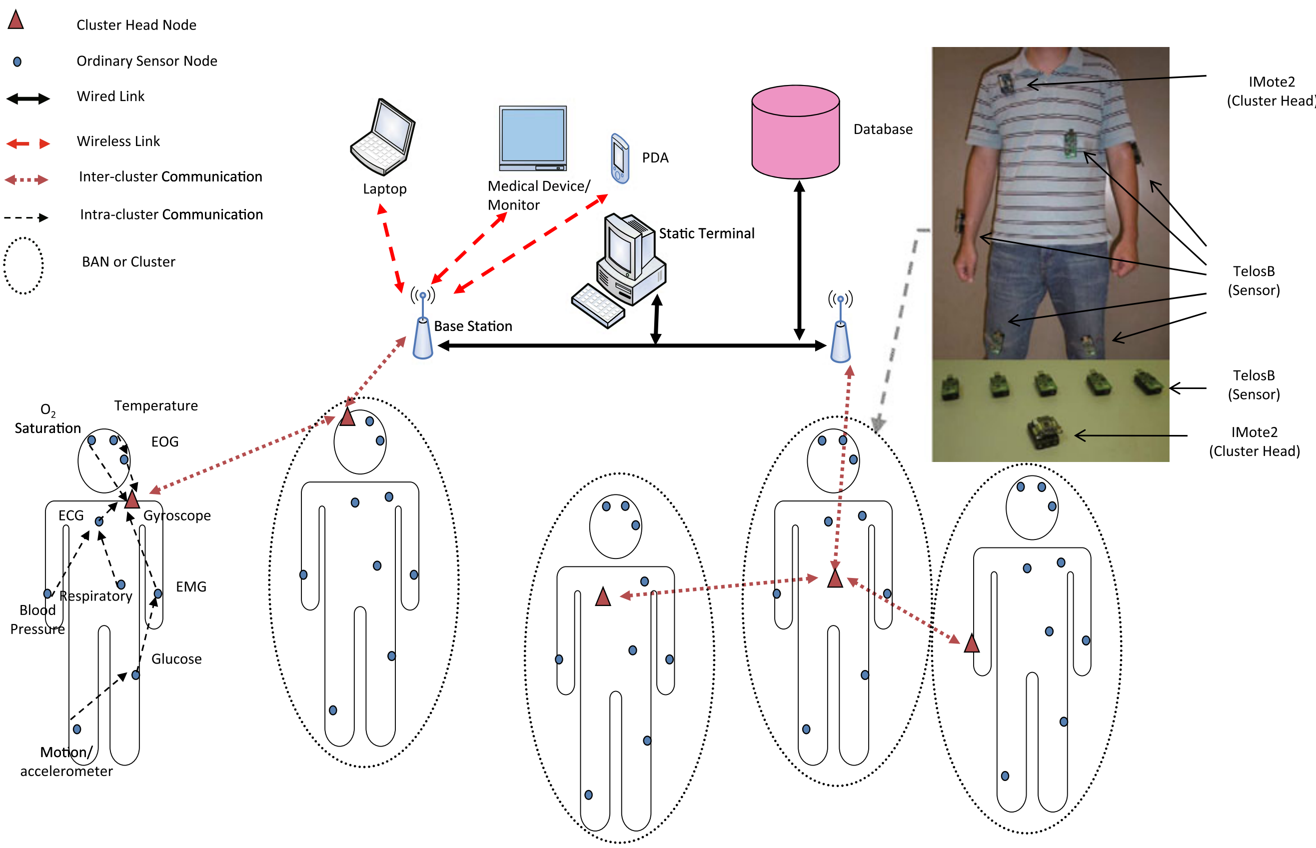


Fig. 1 Proposed physical network architecture

of BAN (intra-BAN). Sensor nodes deployed on a patient form a body area network, which monitors, collects, and pre-processes physiological data of the patient the BAN is associated with. Inside each BAN, a node with higher computing power, battery power, and networking capabilities is selected to play the *logic role* of Cluster Head (CH). This cluster head collects, aggregates, and fuses the data from other sensors in the cluster, performs data-consistency checks, and transmits the processed data to the best base station. The wireless station will then relay the data to a mobile terminal such as laptop, PDA, or medical device, or to a static terminal. Cluster heads, base stations, and mobile and static terminals can also forward each others' packets, share each others' patient information, and access database to obtain, if needed, the patient profile, i.e., the patient medical history.

This two-tier architecture calls for *two different protocol stacks*, one for *intra-BAN* and the other for *inter-BAN* communications. The inter-BAN protocol provides communication between BANs, while the intra-BAN protocol is employed to aggregate patient's vital signs. In this article, we focus on the communications from the BANs to the base stations so cluster heads are assumed to be pre-selected and sinks/destination nodes are assumed to be the base stations.

To seamlessly integrate the inter-BAN and intra-BAN protocols, we adopt super-slots (Fig. 2), each of which is divided into two parts: the first part (Inter-BAN slot) with Multi-channel Quality-based MAC (MQ-MAC) scheme (Section 4.2) for inter-BAN packets and the second part (Intra-BAN slot) with adaptive Time Division Multiple Access (TDMA) scheme for intra-BAN packets. FTSP [18], a network synchronization protocol that is implemented and included in TinyOS 2.1, is used to synchronize the slots.

To avoid EMI, inter-BAN communications employ multiple frequency channels. However, the operations

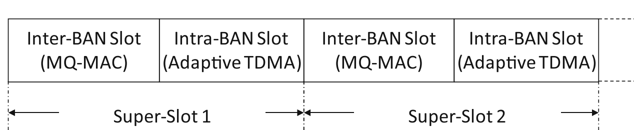


Fig. 2 Super-slot for MAC protocols

are complicated. On the other hand, sensors in a cluster do not need to communicate with other clusters and are relative static. To reduce communication complexity and overhead, the intra-BAN protocol is designed to be simple while aiming at minimizing energy consumption and reducing interference among inter-BAN communications. We give a brief introduction of it here, while the rest of the article will focus on inter-BAN communications.

When the intra-BAN protocol starts, all the sensors in one BAN probe the channels by sending out short messages during intra-BAN slots. Upon reception of these probing messages, CH collects the power and channel information from these sensors, selects the best available channel (i.e., the available channel with the best Link Quality Indicator (LQI) [10] with least interference from neighboring BANs) and appropriate minimal transmission power levels for them, and broadcasts control messages requesting the sensors to tune their channels to the specified channel and adjust their transmission powers. The adaptive TDMA scheme uses the same channel for one BAN, where the intra-BAN slot is divided into sub-slots that are accessed sequentially for data aggregation. The number of sub-slots assigned to one sensor is decided by its sampling frequency. The CH can also broadcast other commands to change other parameters such as sampling frequency of the sensor.

4 Proposed cross-layer solution

In this section, we present our cross-layer inter-BAN communication solution for in-hospital triage. We first provide a brief overview of the solution and then we detail each component of it.

4.1 Overview

To maximize the network performance for in-hospital triage, we adopt a cross-layer design modular approach that jointly considers the interaction of three modules: MAC, routing, and scheduling. With the interaction between these modules, the e2e reliability of the vital signs can be maximized with guaranteed e2e delays.

Cross-layer wireless communication solutions allow for an efficient use of the scarce resources such as bandwidth and battery energy. However, although we advocate integrating highly specialized communication functionalities to improve network performance and to avoid duplication of functionalities by means of cross-layer design, it is important to consider the ease of design by following a modular design approach [24]. This will also allow improving and upgrading particular functionalities without the need to re-design the entire communication system. For these reasons, in our work we rely on the above-mentioned design guidelines and propose a cross-layer communication solution incorporating MAC, routing, and scheduling functionalities. Our cross-layer solution is based on current ZigBee standards and IMote2/TelosB sensors, which means that it can be implemented and deployed without the need for a new standard or hardware platform.

Interactions between the scheduling, MAC, and routing modules are shown as in Fig. 3. Basically, each module operates based on the input from the other two modules and feeds back information in the reverse direction so that the other modules can adjust their operations. The *MAC module* collects the channel quality information and passes it to the routing module, which uses this information to decide the route to the sink. The *routing module* estimates the number of hops N_{sink}, e2e reliability, and delay to the sink, passing them to the scheduling module, which also utilizes N_{tx} (the average number of transmissions to successfully send a packet) to select a proper packet to transmit. Because of the close interaction between these modules, the traffic is served with different priority based on packet's emergency state and vital sign requirements.

4.2 Multi-channel quality-based MAC

Our Multi-channel MAC is designed to have partial cognitive radio capability. In a cognitive radio system, the cognitive process typically starts with spectrum sensing, followed by channel identification and spectrum management [9]. In our case, spectrum sensing is achieved by probing and sensing the quality of the

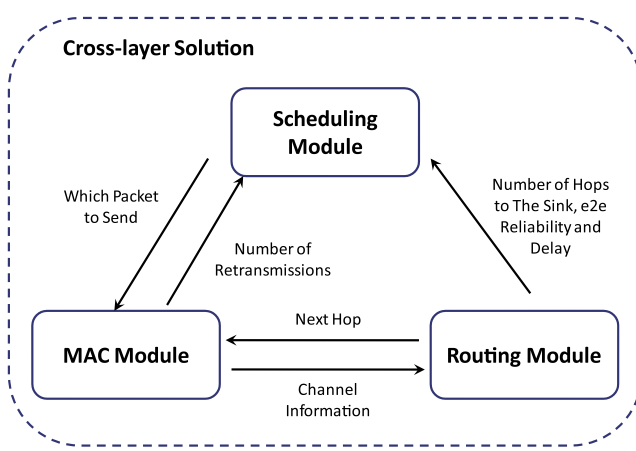


Fig. 3 Interactions between network modules

Table 3 LQI^{RX} (LQI^{TX}) table at receiver (transmitter)

Transmitter (receiver)	DCH11	DCH12	$\cdots$	DCH26
i_1	$LQI_{i_1}^{(1)}$	$LQI_{i_1}^{(2)}$	$\cdots$	$LQI_{i_1}^{(16)}$
i_2	$LQI_{i_2}^{(1)}$	$LQI_{i_2}^{(2)}$	$\cdots$	$LQI_{i_2}^{(16)}$
.	.	.	.	.

16 channels.[1] The measured channel quality serves as a metric to characterize channel activities and variations such as fading. A channel with the best quality, i.e., the channel experiencing the least interference or fading, is chosen for packet forwarding. Moreover, as the channel quality is observed at the receiver, the channel is selected by the *receiver* in our MAC. By using the measured LQI information on multiple channels, our MAC captures channel variations, maximizes channel utilization, reduces interference, and allows simultaneous transmissions.

The spectrum (channel) sensing protocol to probe and track the LQI values of all the channels is designed as follows. A dedicated Common Control CHannel (CCCH) is used for all the nodes to exchange control information. During initialization, a node sends probes and listens on all available data channels. This can be done by competing on CCCH with probes. The winner sends probes over the available channels according to a previously or dynamically agreed schedule. Upon reception of these probes, the receivers build their LQI^{RX} tables (Table 3), which records the LQI values of the channels. LQI^{RX} is also updated upon the reception of a packet on one channel. This LQI^{RX} table is broadcast periodically on CCCH for the transmitters to build up their LQI^{TX} table (Table 3), which will be used by the transmitter for its egress (outgoing) channel information. LQI^{TX} is used by the transmitter for routing, as discussed in Section 4.3.

Our MQ-MAC is shown in Fig. 4. After LQI^{RX} is built, nodes can forward data to each other with MQ-MAC. At the start of a slot, when node i has a data packet for j, it first sends out a Request-To-Send (RTS) packet with the un-preferred channels on CCCH and l, the data packet's length. Here, l is used by the receiver and its neighbors to estimate the time length that the channel will be used. Upon receiving the RTS, j selects the best available channel as,

$$CH_{ij}^{*} = \arg \max_{\substack{CH \in \mathcal{ICH}(j) \\ CH \notin \Lambda(i,j)}} \left\{ LQI_i^{RX} \right\}, \quad (1)$$

[1]IEEE 802.15.4 specifies 16 channels within the 2.4 GHz band, in 5 MHz steps, numbered 11 through 26. Central frequency of channel n is given by [10]: $f_c = 2405 + 5(n - 11)$ MHz, $n = 11, 12, \ldots, 26$.

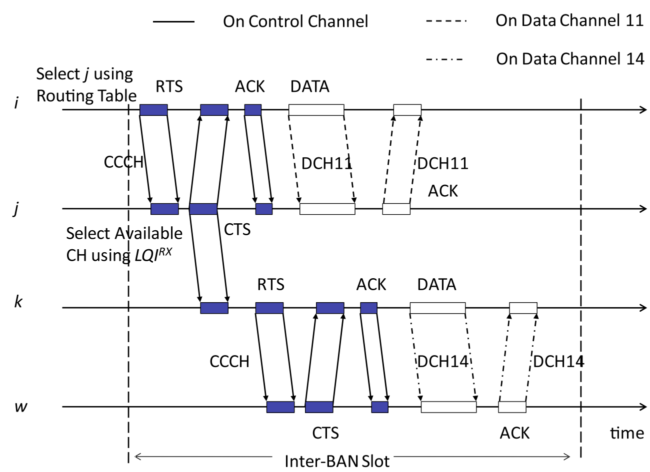


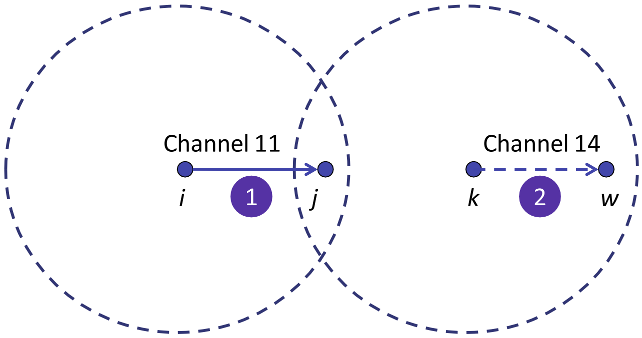
Fig. 4 Proposed inter-BAN MQ-MAC protocol

where LQI_i^{RX} is the set of ingress LQIs from i in table LQI^{RX}, $\mathcal{ICH}(j)$ is the idle channel set (estimated by overhearing RTS/CTS messages) in j's neighborhood, and $\Lambda(i, j)$ is the set of the unavailable channels just received from i and the adjacent channels of those active channels near j. Channels in $\Lambda(i, j)$ are not selected in such a way as to avoid interference at the transmitter and from adjacent channel [13]. Clear-To-Send (CTS) is then broadcast with CH_{ij}^{*} and l on CCCH. If no channel can be selected, CTS is broadcast with channel number 0, telling i to backoff.

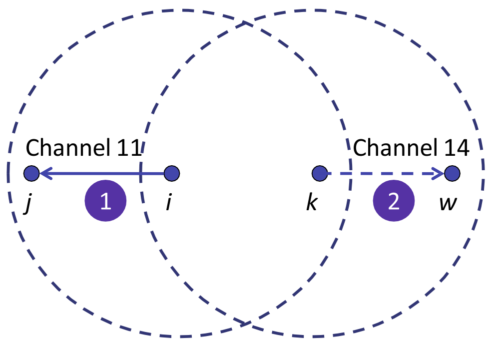
After receiving CTS, i sends back an ACKnowledgement (ACK) message with CH_{ij}^{*} if this channel is not reserved or active in the neighborhood. On hearing this message, i's neighbors will mark CH_{ij}^{*} as reserved. Otherwise, i sends back a new RTS to ask j to re-select another best channel. This negotiation is repeated until the transmitter i and receiver j settle on a common available channel.

Finally, i tunes its channel to CH_{ij}^{*} and starts the transmission of the data packet to j. An ACK message will be sent back to i for acknowledgement. If all these steps are successful, i.e., all messages are received, i and j will tune their channels back to CCCH. Otherwise, retransmission will be made after a timeout expires until the maximum number of retransmissions is reached. The channel will be tuned back to CCCH upon the maximum number of retransmissions or the end of the slot.

Remark 1 Although the quality of egress channels can be obtained in LQI^{TX}, this information may not be up to date. Hence, relying on the receiver to use CTS to reserve the channel is a better way. 2) ACK is used to reserve the channel at the transmitter while the new RTS is for channel renegotiation.



(a) Hidden Terminal Scenario



(b) Exposed Terminal Scenario

Fig. 5 Hidden/exposed terminal scenario

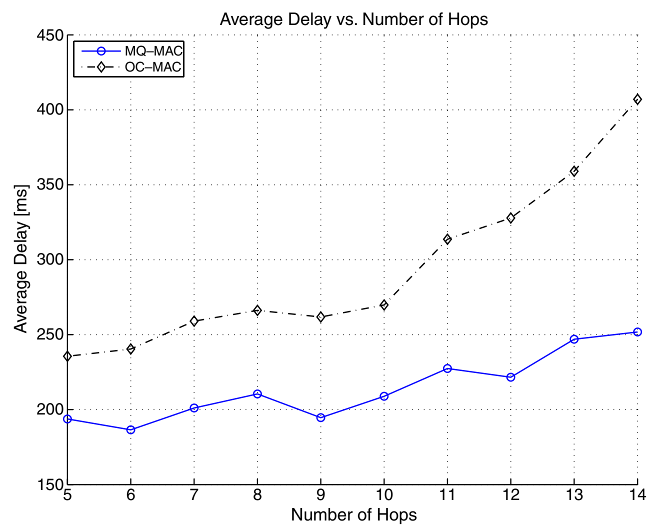
The *hidden terminal problem* is handled as depicted in Fig. 5a. Suppose at first i is sending to j on channel 11 and k needs to send a packet to w. In one-channel MAC protocols, this may result in one of the three cases: neither i nor k can send; collision happens at j; or k will backoff.

In our MQ-MAC, according to the above design, RTS is broadcast with un-preferred channel 11, asking w to select a channel via CTS; w chooses channel 14 according to (1). With the help of CTS and ACK, k can use channel 14 to forward data to w. With different channels, i and j, and simultaneously k and w, can communicate without blocking each other.

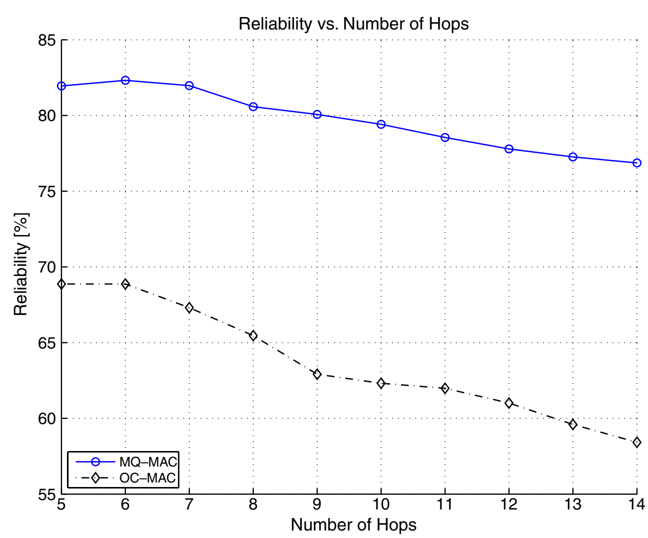
Even another delicate situation can be handled. Suppose that, right after i has sent a RTS to j, k also sends a RTS to w, and then k receives j's CTS before w's CTS, both trying to reserve channel 11. In such a case, as j reserved channel 11 first, k will give up channel 11 and send out another RTS to renegotiate with w. Finally, i and j settle on channel 11 while k and w settle on channel 14. Simultaneous communications via different channels are still possible. Hence, hidden terminal problem is handled.

Furthermore, the *exposed terminal problem* is also handled as shown in Fig. 5b. In one-channel MAC protocols, when this happens, normally either k will not transmit or k still uses the same channel. Yet, interference between both communication pairs still exists due to the usage of the same channel. In our proposed MQ-MAC, by overhearing i's RTS and ACK, k is able to select a different channel for transmitting and, thus, interference will be greatly reduced. Therefore, exposed terminal problem is better handled than in other competing MAC protocols in the same class. To show the advantage of our cognitive radio approach, we compare our solution with the standard one channel CSMA/CA MAC (denoted by OC-MAC). The performance comparison is shown in Fig. 6a, b.

In a multi-hop scenario, comparisons are made with a network that routes on shortest paths, i.e., paths with the least number of hops to the sink. Our results



(a) e2e Delay



(b) e2e Reliability

Fig. 6 One-channel MAC (OC-MAC) vs MQ-MAC show that exploiting channel diversity brings a lot of improvement. The communication protocols are implemented on TinyOS, the embedded operating system designed for networked sensors. These protocols are loaded into the TelosB sensors and tested in our testbed. Experiments were carried on in the aisles of the floor (around $50 \times 20\text{m}^2$) in our building with a sink node located at one corner as shown in Fig. 11a. From the results above, it is clear that our approach, which exploits the channel diversity with measured channel quality, achieves higher packet delivery ratio than those using only one channel.

4.3 Channel quality based routing

These results show that our MQ-MAC protocol using multi-channel and measured channel quality information can achieve better MAC-layer performance than competing schemes. We are interested now in extending it into routing and see how much improvement we can obtain. Motivated by this, we propose a Channel Quality Based Routing (CQBR) algorithm that uses multi-channels and measured channel quality information for routing. Moreover, the MQ-MAC scheme proposed in Section 4.2 is also incorporated into the routing algorithm.

We aim at achieving full utilization of the channels and robustness against interference. The objectives of our CQBR protocol are:

1. Use measured channel quality to select a route with the best quality;
2. Select the optimal channel in terms of link quality;
3. Allow simultaneous transmissions on different channels in the neighborhood.

Let us define the *Route Quality Indicator* RQI_{SD}, a metric to measure the route quality from source S to D, as

$$RQI_{SD} = \min_{(i,j)\in\mathcal{R}_{SD}} \max_{c\in ACH_{ij}} LQI_{ij}^{c}, \tag{2}$$

where (i, j) is the link from i to j, $\mathcal{R}_{SD}$ is the set of links along the route from S to D, ACH_{ij} is the set of available channels for link (i, j), and LQI_{ij}^{c} is the LQI from i to j via channel c.

By combining routing and MQ-MAC, we obtain a cross-layer protocol, CQBR, which selects the next hop based on both the RQI estimated at the *transmitter* and the best LQI channel measured at the *receiver*. When a node needs to route traffic to the sink, it selects the best next hop using RQI in the routing table (Table 4), i.e., data traffic is forwarded to the neighbor that has the best RQI. With the MQ-MAC protocol, the selected next hop commands the transmitter to tune to the optimal channel for data forwarding. Note that the number of hops to sink is used by the hybrid scheduler (Section 4.4) to estimate the maximum tolerable delay for the packets at current node.

Table 4 Routing table at node i

Destination	Next hop	Route quality	Hops to sink
d_1	j_1	$RQI_{ij_1d_1}$	n_1
d_2	j_2	$RQI_{ij_2d_2}$	n_2
.	.	.	.

Let us assume that i needs to route a packet to some destination D; CQBR works by selecting the optimal available next hop with the best RQI as,

$$j^* = \arg\max_{j\in\mathcal{N}(i)} RQI_{ijD}, \tag{3}$$

where $\mathcal{N}(i)$ is the set of i's non-busy neighbors (estimated by looking into the overheard RTS/CTS packets with the data packet length) and RQI_{ijD} is the RQI value of the route from i to sink D via j. Then, it utilizes MQ-MAC to select the best channel and to forward the packet to the next hop.

To illustrate how the routing protocol works, an illustrative example is given in Fig. 7. The maximum LQI values of each communication pair are given in the figure, and the RQI values for route from 1 to sink via 2, from 1 to sink via 4, and from 7 to sink are 60, 100, and 105, respectively, according to (2). When node 1 has data to send to the sink, it selects the route via 4 as the RQI of this route is greater than that of the route via 2. By using a route with high RQI, the node can be sure that there is no bad LQI channel along the route.

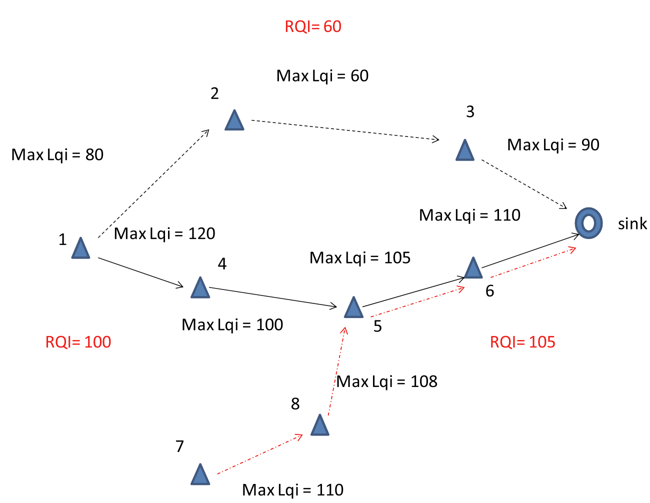


Fig. 7 A CQBR example

The routing tables needed by the protocol to properly operate, LQI^{TX} and LQI^{RX}, are created and maintained as follows:

- **Initialization:** Channels are scanned and LQI^{RX} is created at each node. This table is then broadcast for the neighbors to create their LQI^{TX} tables. Routing table is created from LQI^{TX} with only the entries to neighbors;
- LQI^{RX} is updated upon reception of packets;
- Periodically, LQI^{RX} and routing table are broadcast to neighbors;
- Update LQI^{TX} with LQI^{RX}s from the neighbors;
- Update routing table with LQI^{TX} and routing information from neighbors.

The e2e performance of our MQ-MAC with CQBR routing (MQMAC-CQBR) in terms of delay and reliability is compared with the following protocols:

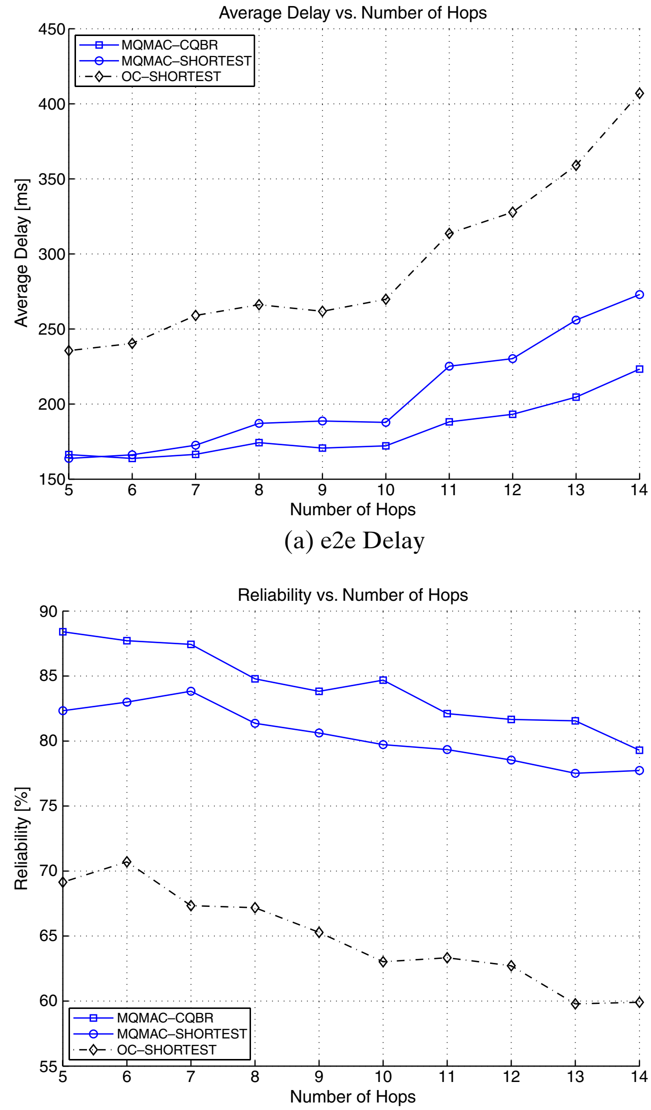
- Protocol stack with one-channel CSMA/CA MAC protocol and routing on the path with the shortest number of hops (OC-SHORTEST);
- Protocol stack with our Multi-Channel MAC and routing on the path with the shortest number of hops (MQMAC-SHORTEST).

The evaluation results in Fig. 8a, b show that our solution gives better performance both in terms of e2e delay and reliability than the other two competing protocols.

4.4 Two-level scheduling algorithm

The above comparisons show that the cross-layer routing design detailed in Section 4.3 gives performance improvement over conventional protocols using one channel. Yet, protocols so far have no mechanism to guarantee the delay requirements of the traffic. As discussed in Section 2.3, an EDF scheduler is an effective way to bound the packet delay. Generally, a scheduler with only one queue is not an efficient way to do EDF scheduling. In contrast, as discussed in Section 4.4.2, having a number of queues within the EDF scheduler can decrease scheduling complexity and the queuing delay for the packets, meeting the delay requirements of different types of data. However, an EDF scheduler alone only guarantees the traffic delay, and cannot provide differentiated service to distinct classes of traffic, e.g., the red, yellow, and green traffic during triage.

Definition 1 A **flow** $f_{i,j}^{(t)}$ is defined as the traffic of the same type t of sensing data with the same QoS requirement sent from source i to destination j.



(a) e2e Delay

(b) e2e Reliability

Fig. 8 Comparison of the joint MAC and routing protocols

Obviously, the vital signs during triage have two different QoS requirements with the first based on patient's condition and the second from the e2e requirement of different *flows*, as defined in Definition 1. The first requirement is *class specific* while the second is *flow specific*. GPS schemes can be used to provide class-specific service, which assigns different bandwidths to different classes of traffic. However, GPS cannot guarantee the flow delay requirement. On the other hand, EDF scheduling can provide guarantee e2e services to the flows but cannot satisfy the class-specific requirement for triage, where traffic from patients in more critical conditions should be served earlier than that from those patients in safer conditions. To take the advantages of these two scheduling schemes, we adopt a two-level Hybrid Scheduling (HS) approach with the first being EDF scheduling level to meet the e2e requirement of the flows and the second being GPS scheduling level based on the traffic class. Both levels adapt to the traffic variations, thus maximizing the reliability of these classes while meeting the flow e2e requirements. As discussed in Section 2.3, hybrid schedulers such as [8] and [38] can apply either EDF or WFQ but not both to a flow, which cannot satisfy the above traffic requirements.

Consequently, as shown in Fig. 9, two levels of scheduling are employed to select a packet to send. At the first level, *flow-based scheduling*, packets are placed into three classes (red, yellow, and green) of queues according to the associated patient's condition. Then, EDF scheduling is applied to choose a packet for each class in the ascending order of the packets' estimated expiration time. At the second level, *class-based scheduling*, a GPS scheduler is proposed to select one among the chosen packets at the first level for transmission. Suppose the weights assigned to the red, yellow, green classes of queues are ϕ_R, ϕ_Y, ϕ_G, respectively, for the GPS scheduler, then it is clear that we have $\phi_R + \phi_Y + \phi_G = 1$ and $\phi_R \geq \phi_Y \geq \phi_G$ based on the group's significance. Note that these weights are *adjusted adaptively* according to the feedback from the sink.

Compared to the Differentiated Services (DiffServ) approach to provide QoS to traffic, as defined in IETF RFC 2474, our hybrid scheduling algorithm is different in several aspects: 1) our algorithm can guarantee the e2e delay requirements of the vital signs, while DiffServ can only serve the traffic based on the class requirements and, thus, does not in general guarantee the delay requirement of the vital signs; 2) our algorithm is specifically tailored for patient emergency and delay requirements of vital signs, as in Table 1, while DiffServ is designed for different classes of traffic such as voice

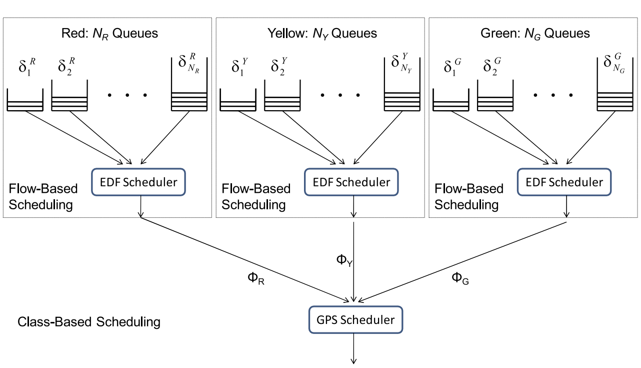


Fig. 9 Hybrid scheduler (HS)

and video with quite different requirements; 3) our algorithm can adjust the traffic class coefficients adaptively using feedback information from the sink, while DiffServ generally does not use any feedback mechanism and therefore is not able to adapt to changes of network traffic. On the other hand, compared with the Integrated Services (IETF RFC 1633) QoS approach, which treats each flow individually without bundling them into aggregates belonging to the same class, our algorithm has much lower complexity and better scalability although a certain "bandwidth inefficiency" is introduced, as will be discussed in Section 4.4.2.

In the following sections, we focus on the feedback mechanism to maximize the reliability of different classes of traffic with guaranteed e2e delay requirements of different flows.

4.4.1 Class-based scheduling with feedback

To maximize the reliability for these three classes of traffic while guaranteeing e2e delay, an adaptive feedback mechanism is used to adjust the GPS weight triple (ϕ_R, ϕ_Y, ϕ_G) at each CH. Feedback packet follows the reverse path from the sink to the source. Depending on the feedback from the sink, which includes the reliability value per class and the average delay per flow, the parameters of the scheduler (number of queues and fraction of capacity to red/yellow/green queues) are adjusted on each node along the path from the source to the destination.

In our work, packets of higher class—red or yellow—are delayed to reliably deliver traffic of lower class—yellow and green—, respectively, so that the reliability of the lower class is maximized. Note that the e2e delay requirements for the higher classes are still guaranteed. A packet is reliably delivered to the destination when it is received before it expires, while expired packets are dropped in the network no matter their class (color).

This mechanism relies on the sink to record statistical data, which includes the numbers of packets and average delay per flow. This data is then sent back to the source CHs to adjust their weight triples (ϕ_R, ϕ_Y, ϕ_G). To improve the efficiency, the statistic data is put into one packet and periodically sent back to the corresponding source CHs. Note that, as feedback packets follow the reverse path from the sink to the source, no additional routing overhead is needed. Finally feedback information can be aggregated, using techniques such as multicasting, which can further reduce the overhead to route the feedback information.

The problem of updating the triple (ϕ_R, ϕ_Y, ϕ_G) for a network with N_B CHs is actually a stochastic network problem whose formulae for the delays or reliability for each flow are hard to derive. Our approach here is to use the following numerical method, which is shown to provide excellent results.

Suppose that the measured e2e reliability γ_R^{e2e}, γ_Y^{e2e}, γ_G^{e2e}, and time difference δ_R^{e2e}, δ_Y^{e2e}, δ_G^{e2e} between corresponding Time-To-Lives (TTLs) and the measured e2e delays are obtained from the sink's feedback information. Our algorithm trades γ_G for γ_R and γ_Y in light congestion condition, and γ_G and γ_Y for γ_R in moderate congestion condition by adjusting (ϕ_R, ϕ_Y, ϕ_G), as shown in Algorithm 1, where $\alpha_R, \beta_R, \beta_Y \in (0, 1)$ are adjustment coefficients, and Δ_R^{e2e}, Δ_Y^{e2e}, Δ_G^{e2e} are threshold values for δ_R^{e2e}, δ_Y^{e2e}, δ_G^{e2e}, respectively. The complexity for this algorithm is $O(1)$.

Algorithm 1 Adaptive GPS Algorithm

```
if Receive Feedback From the Sink then
  // Heavy Congestion
  if γ_R^e2e < 1 and φ_G ≠ 0 then
    φ_R = φ_R + φ_G/2; φ_G = φ_G/2
  end if
  if γ_R^e2e < 1 and φ_Y ≠ 0 and φ_G = 0 then
    φ_R = φ_R + φ_Y/2; φ_Y = φ_Y/2
  end if
  // Intermediate Congestion
  if γ_R^e2e = 1 and γ_Y^e2e < 1 and φ_G ≠ 0 then
    φ_Y = φ_Y + φ_G/2; φ_G = φ_G/2
  end if
  if γ_R^e2e = 1 and γ_Y^e2e < 1 and φ_G = 0 then
    if δ_R^e2e > Δ_R^e2e then
      φ_Y = φ_Y + α_R φ_R; φ_R = (1 − α_R)φ_R
    end if
  end if
  // Light Congestion
  if γ_R^e2e = 1 and γ_Y^e2e = 1 then
    if δ_R^e2e > Δ_R^e2e and δ_Y^e2e > Δ_Y^e2e then
      φ_G = φ_G + β_Y φ_Y + β_R φ_R
      φ_Y = (1 − β_Y)φ_Y; φ_R = (1 − β_R)φ_R
    end if
    if δ_R^e2e ≤ Δ_R^e2e and δ_Y^e2e > Δ_Y^e2e then
      φ_G = φ_G + β_Y φ_Y; φ_Y = (1 − β_Y)φ_Y
    end if
    if δ_R^e2e > Δ_R^e2e and δ_Y^e2e ≤ Δ_Y^e2e then
      φ_G = φ_G + β_R φ_R; φ_R = (1 − β_R)φ_R
    end if
  end if
end if
```

As it is hard to derive the formulae for such a stochastic network, trying to find the optimal ϕ_R, ϕ_Y, and ϕ_G is difficult. Our solution adjusts them based on the network congestion status determined from the feedback information. It is obvious that we should trade the service to the 'yellow' (or 'green') traffic for the service to the 'red' (or 'yellow') traffic. Therefore, the coefficients for the less important traffic are adjusted in a coarser manner and this is taken as a guideline for the design of Algorithm 1. For example, if the network is found to be in heavy congestion (Table 2), ϕ_R is adjusted by adding half of ϕ_Y (or ϕ_G) while ϕ_Y (or ϕ_G) is adjusted by taking half of the original value. If the network is found to be in intermediate congestion (Table 2), the algorithm first tries to degrade the service to the 'green' traffic by adding half of ϕ_G. If there is no 'green' traffic to degrade and the requirement for 'red' traffic is satisfied, it then tries to increase ϕ_Y by decreasing ϕ_R; α_R can be small so that the service to 'red' traffic will not be greatly degraded. For these cases, the coefficients for less important traffic are adjusted in a coarser way by taking half of the original value than that for the 'red' traffic in order to guarantee the service to the 'red' traffic. In the case that the service to the 'red' traffic is guaranteed, the coefficients are adjusted using a smaller step size. In this way, ϕ_R, ϕ_Y, and ϕ_G can approach their optimal values with iterations. Similar adjustments are used in other congestion cases.

Once (ϕ_R, ϕ_Y, ϕ_G) is updated, the scheduler will recalculate *virtual finish time* [22] and schedule the transmission based on the new values. Notice that as only 3 packets (one from each class) are to be scheduled each time, updating virtual finish time is quick. Hence, the *transmission disorder problem* [35] can be easily handled.

4.4.2 Flow-based scheduling optimization

With the triple (ϕ_R, ϕ_Y, ϕ_G) tuned, the next step is to adjust the number of queues within each class so that the 'bandwidth inefficiency' is minimized while limiting the EDF scheduling complexity. With the EDF algorithm, every packet in a group needs to be sorted according to its deadline when it arrives at the node. The complexity for this sorting operation is $O(\log k)$, where k is the number of packets in the buffer. One original method to reduce the complexity is to maintain a number of queues for the flows into the node. Arriving packets are put into the end of one queue that can guarantee the delay requirement properly. The complexity is reduced to $O(\log N)$, where N is the number of flows arriving at the node.

In a heterogenous environment, such as for wireless vital sign monitoring networks, there are many types of data with different requirements and delay constraints. The maximum tolerable queuing delay by this data can therefore vary depending on the temporal correlation of the vital sign: it is small for vital signs such as ECG readings and larger for temperature readings. Delay constraint is critical for vital healthcare data. On the other hand, it is also necessary to guarantee that the delay insensitive data do not starve, i.e., do not suffer from an uncontrolled delay so to keep certain QoS provisioning.

Hence, in order to handle the heterogenous data, we build an EDF scheduler with N queues where N is the number of the flows. However, when N is large, the complexity $O(\log N)$ is large, which may not be too much of improvement with respect to the case in which the sorting operation is per packet (e.g., if N is not much smaller than k). On the other hand, if we reduce the number of queues to have a lower complexity, a packet with tolerable queuing delay D may be placed in a queue with a smaller queuing delay ($< D$), which introduces certain 'bandwidth inefficiency' resulting from assigning packets with higher tolerable delays to queues with small delays.

Considering the limited processing capability of the sensors, it is necessary to find a solution so that each sensor maintains an appropriate number of queues by trading off admissible complexity and 'bandwidth inefficiency'. To be more specific, we define *bandwidth inefficiency* as follows.

Definition 2 Assume that there is a set of queues

$\mathbb{Q} = \{q_1, q_2, \ldots, q_K\}$

in some node with maximum delays $\mathbb{D} = \{d_1, d_2, \ldots, d_K\}$, respectively, such that $d_1 < d_2 < \cdots < d_K$. If we consider $d_{K+1} = \infty$, the **bandwidth inefficiency** (or simply **inefficiency**) with EDF scheduling for this node is

$$\eta = \sum_{i=1}^{K} \sum_{\substack{\xi \in \Omega, \\ d_i \le d_\xi^{(r)} < d_{i+1}}} R_\xi \cdot \left(d_\xi^{(r)} - d_i \right), \tag{4}$$

where Ω is the set of packets in these queues, $d_\xi^{(r)}$ is packet ξ's remaining maximum tolerable delay at current node, and R_ξ is the bit rate for sending ξ.

Note that the reason for subtracting d_i from $d_\xi^{(r)}$ in (4) is due to the fact of putting ξ into q_i. As $d_\xi^{(r)} < d_{i+1}$, ξ may expire if it is put into queue q_j with $j > i$. On the other hand, if ξ is put in q_j with $j < i$, then it may increase the risk of other packet's expiration. For example, packet ξ' arrives after ξ with $d_{\xi'}^{(r)} < d_\xi^{(r)}$. Yet, ξ must be served before ξ', causing possible deadline expiration for ξ' since ξ may be served at a time that ξ' expires. Hence, it is the right choice to put ξ in q_i. The *maximum tolerable delay at current node* is estimated by dividing the remaining alive time by $N_{sink} \cdot N_{tx}$ (as introduced in Section 4.1).

Our EDF scheduling scheme works by measuring the traffic queueing delay and adaptively adjusting to the observed variations. Suppose that during one measured period the observed minimum packet delay is d_{min} and the maximum packet delay d_{max}. The delay between d_{min} and d_{max} is divided into $M-1$ intervals ($M > 2$) with the end points being $d_i = d_{min} + (i-1)(d_{max} - d_{min})/(M-1)$ for $i = 1, 2, \ldots, M$. These end points correspond to the maximum delays of M queues $q_1, q_2, \ldots, q_M$. A packet ξ is put into q_i if its maximum tolerable delay at this node $d_\xi^{(r)}$ satisfies $d_i \le d_\xi^{(r)} < d_{i+1}$. Suppose that during this observation period, the numbers of packets within these queues are $N_1, N_2, \ldots, N_M$, respectively. Starting from these M queues, an iterative procedure is executed to reduce the number of queues from M to the desired number M_d by combining these queues. Our goal here is to *minimize the bandwidth inefficiency while maintaining the desired number of queues*. Within each iteration, a queue is combined into another with minimum inefficiency, resulting in $M - M_d$ iterations.

As IEEE 802.15.4 has only one bit rate of 250 kbps, R_ξ's are all the same. Hence, we can remove R_ξ from (4). Moreover, the relative inefficiency of the packet within a queue is fixed when this queue is combined into another queue. Therefore, we can ignore this inefficiency within the queue and estimate the relative inefficiency between the queues when we combine the queues with minimized inefficiency. To be more exact, suppose q_j (with N_j packets and maximum delay d_j) is combined into q_i (with maximum delay d_i and $d_i < d_j$), then the *increased inefficiency* for this combination is defined as $\hat{\eta}_{ji} = N_j \times (d_j - d_i)$.

The basic idea of our algorithm is as follows: starting from the current set of queues, we calculate the increased inefficiency of combining a queue with its previous queue. Then, the queue with the least increased inefficiency is selected to combine with its previous queue. This is repeated until we reach M_d queues.

For clarity, Fig. 10a–d show the step-by-step procedure of our algorithm. Suppose we have five queues q_1 to q_5 with the number of packets being 4, 1, 2, 3, 2, respectively. Their maximum delays are $d_1, d_2, \ldots, d_5$ such that $d_{i+1} - d_i = 1$ $(i = 1, \ldots, 4)$. We want to combine these 5 queues into two while minimizing the inefficiency. As shown in Fig. 10a, the increased inefficiency $\hat{\eta}$s of combining only one queue with corresponding preceding queue is calculated as +1, +2, +3, +2. For instance, combining q_4 into q_3 results in increased inefficiency of $\hat{\eta}_{43} = N_4(d_4 - d_3) =$

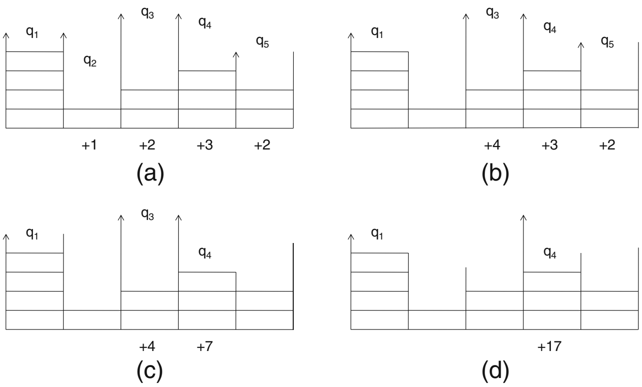


Fig. 10 Queue combination example

$3 \times 1 = 3$. Comparing these $\hat{\eta}$s, we select q_2 to join q_1, as shown in Fig. 10b. Then, the $\hat{\eta}$'s are calculated as $+4, +3, +2$. Note that the displacement of the maximum delay from q_3 to q_1 is now $d_3 - d_1 = 2$. So, the increased inefficiency of combing q_3 to q_1 can be computed as $\hat{\eta}_{31} = N_3(d_3 - d_1) = 2 \times 2 = 4$. As $+2$ is the minimum $\hat{\eta}$, q_5 is combined into q_4, resulting in three queues q_1, q_3 and q_4, as depicted in Fig. 10c. This iteration goes on until the desired number of queues is reached. Note that in Fig. 10c $\hat{\eta}$ from the combined q_4 to q_3 is evaluated as $N_4(d_4 - d_3) + N_5(d_5 - d_3) = 3 \times 1 + 2 \times 2 = 7$, while $\hat{\eta}$ from q_3 to the combined q_1 is evaluated as $N_3(d_3 - d_1) = 2 \times 2 = 4$. In Fig. 10d, $\hat{\eta}$ from the combined q_4 to the combined q_1 is evaluated as $N_4(d_4 - d_1) + N_5(d_5 - d_1) = 3 \times 3 + 4 \times 2 = 17$.

At the beginning of each iteration, χ is used to denote the ascending index set of the combined queues such that χ_i represents the *minimum index of the initial queues* for the i-th combined queue, i.e.,

$$\chi_i = \min\{j | q_j \text{ is combined into the } i\text{-th combined queue}\}.$$

The i-th combined queue has the maximum delay as q_{χ_i}, i.e., d_{χ_i}. As our algorithm merges neighboring queues, this i-th combined queue contains q_j such that $\chi_i \leq j < \chi_{i+1}$. Specifically, at each iteration, the increased inefficiency to combine queues specified by χ_{i+1} to χ_i can be formulated as,

$$\hat{\eta}_{(i+1)i} = \sum_{\chi_i \leq k < \chi_{i+1}} N_k \left(d_k - d_{\chi_i}\right). \tag{5}$$

The EDF scheduler merges the neighboring queues specified by χ_{i^*} and χ_{i^*+1} such that

$$i^* = \arg \min_{1 \leq i < size(\chi)} \hat{\eta}_{(i+1)i}, \tag{6}$$

where $size(\chi)$ is the number of elements in set χ, which is updated by removing χ_{i^*+1}. The algorithm then iterates on the new χ until the desired number of combined queues is reached, as shown in Algorithm 2. The complexity of this algorithm is $O((M - M_d)M)$.

Algorithm 2 Algorithm to merge M queues to M_d queues ($M \geq M_d$)

```
Initialization: χ = {1, 2, ..., M}, d_1 < d_2 < ··· < d_M
for j=M to M_d + 1 do
  //Find the combined queue with minimum increased inefficiency
  η̂_min = +∞; i_min = 0
  for i=1 to size(χ)-1 do
    η̂_i = Σ_{χ_i ≤ k < χ_{i+1}} N_k(d_k − d_{χ_i})
    if η̂_min > η̂_i then
      η̂_min = η̂_i; i_min = i
    end if
  end for
  //Merge neighboring queues
  Remove χ_{i_min+1} from χ
end for
```

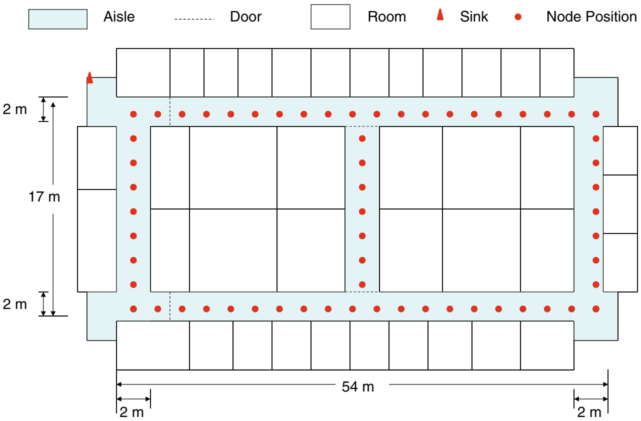
5 Performance evaluation

To evaluate the whole solution presented in Section 4, we implemented the protocols on TinyOS and loaded them into the IMote2/TelosB sensors and tested in our testbed.

5.1 Experiment setup

Experiments were carried on in the aisles of the floor (around $50 \times 20\text{m}^2$) in our building with a sink node located at one corner (Fig. 11a) and N_B BANs evenly distributed in the aisles to emulate multiple patients to be triaged. Every BAN has 3 TelosB sensors that emulate the sampling of ECG, CO_2, and CVP blood pressure data, respectively, and forward them to one IMote2 sensor acting as a CH, forming a BAN. As shown in Fig. 11b, the IMote2 on top of the cup is the CH with 3 TelosB around it as intra-BAN sensors. Each CH will forward the data for itself or other BANs to the sink. Hence, the entire network is composed of several BANs, overall forming a two-tier architecture.

These ECG, CO_2, and CVP blood pressure vital signs are segments of raw data taken from one patient, representing delays of *low*, *medium*, and *high*, respectively. Sampling rates are chosen to be 100, 63, and 63 Hz according to Table 1, respectively, while delays are chosen to be 10, 70, and 120 s, respectively. Every 30 min every node changes randomly into one of the 'Red', 'Yellow', and 'Green' status. Queuing



(a) Floorplan

(b) Sensor Deployment

Fig. 11 Experiment scenario

parameters are set as: $\alpha_R = 0.3$, $\beta_R = 0.3$, $\beta_Y = 0.3$, $\Delta_R^{e2e} = \Delta_Y^{e2e} = \Delta_G^{e2e} = 1$ s, and $M_d = 4$. Initial number of queues for these 3 types of patients are set to 10, while initial values for ϕ_R, ϕ_Y, ϕ_G are set to 0.5, 0.35, 0.15, respectively.

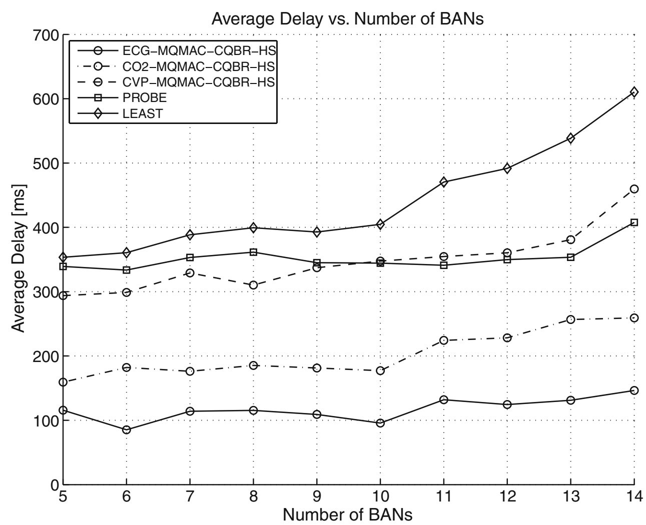
As our solution exploits channel diversity and quality information to avoid interference, we are interested in comparing it with approaches using only *one channel* and those that do not rely on link-quality information. Specifically, our solution is compared against the following two protocols:

1. One-channel protocol without any LQI information (LEAST), which only knows if a link to another node is available or not and routes on the path with the least number of hops;
2. One-channel protocol with probing (PROBE), which probes for the best channel first, then stays on this channel for communications, and routes with the maximum RQI (defined as the minimum LQI value among the links along a route, i.e., $\min_{(i,j)\in R_{SD}} LQI_{(i,j)}$).

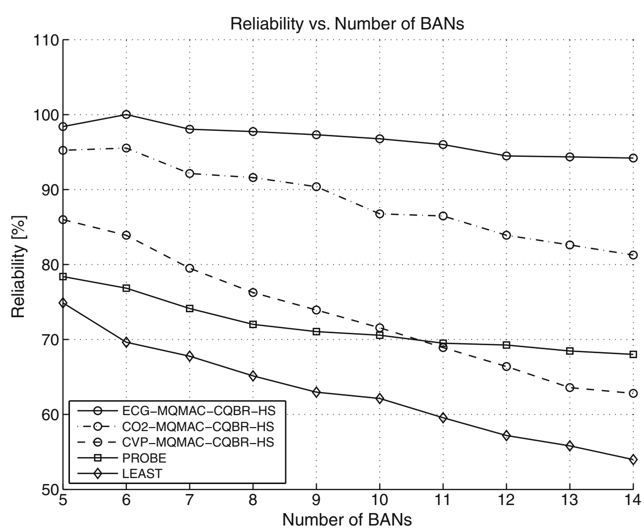
We are interested in measuring and comparing the e2e performance including delay, reliability, and fairness associated with delay and reliability. In this paper, we adopt Jain's fairness measure [12], which is defined as $f(x_1, x_2, \ldots, x_n) = \frac{(\sum_i x_i)^2}{n(\sum_i x_i^2)}$ for a set of values $x_1, x_2, \ldots, x_n$, and compare these protocols. It ranges from $1/n$ (worst case, i.e., most unfair) to 1 (best case, i.e., maximum fairness for all these n values).

5.2 Delay and reliability

The curves for our solution in the following figures are denoted by "X-MQMAC-CQBR-HS", where X represents the patient's data type (ECG/CO_2/CVP). As for e2e delay, our experiment (Fig. 12a) shows



(a) e2e Delay

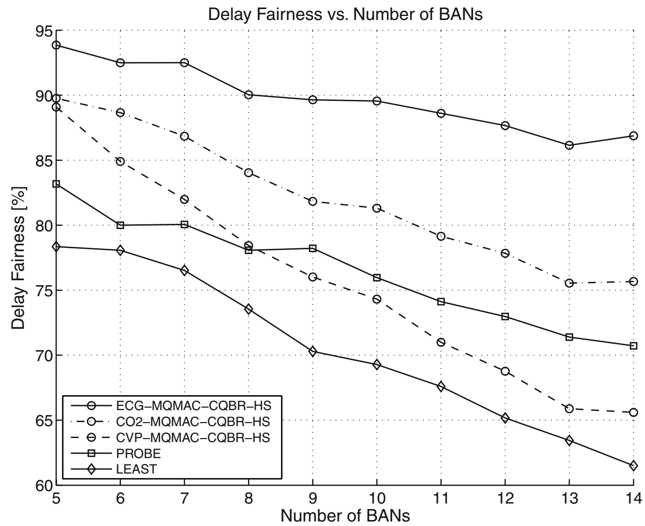


(b) e2e Reliability

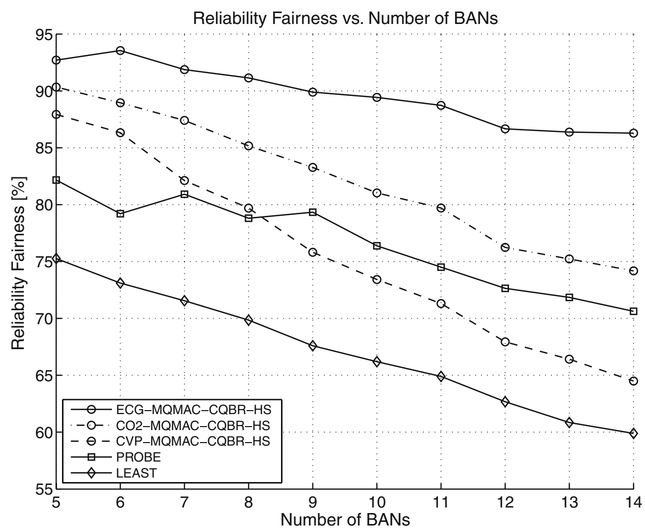
Fig. 12 Delay and reliability comparison that MQMAC-CQBR-HS < PROBE < LEAST. On average, MQMAC-CQBR-HS has the least e2e delay, while LEAST has the greatest delay, whit PROBE in the middle. The reason why MQMAC-CQBR-HS is better than PROBE is that it has a hybrid scheduler to guarantee delay; consequently, the Packet Error Rate (PER) is reduced by exploiting channel diversity to allow simultaneous transmissions and avoid interference, and by selecting the best route with best link quality; conversely, PROBE does not have the scheduler and can only select the best route with good link quality without the option of choosing better channels. On the other hand, PROBE is better than LEAST as the latter chooses the route with the shortest number of hops, leading to the severest interference from neighbor nodes while PROBE tends to select better links for packet forwarding.

Our proposal provides effective priority services based on the vital sign's e2e delay requirements. As for e2e delay, we have ECG < CO_2 < CVP. Note that when N_B is small, the delay of the CVP data is less than that of PROBE as it can choose better channels while PROBE cannot. When N_B becomes bigger, the lowest priority offered to the CVP data due to largest delay requirement introduces much delay that cannot be offset by using good LQI channels so the delay is greater than that with the non-priority PROBE.

As far as the e2e reliability is concerned (Fig. 12b), as N_B increases, all these protocols show decreasing reliability due to an increased interference and a higher number of hops to the sink, which result in increased PER. Due to similar reasons as above, MQMAC-CQBR-HS has the highest reliability while LEAST has the worst, and data with stricter delay requirement get better reliability, too.



(a) e2e Delay



(b) e2e Reliability

Fig. 13 Fairness comparison

5.3 Fairness

As depicted in Fig. 13a, MQMAC-CQBR-HS has better e2e delay fairness than the other two competing protocols. It has more distributive routes because when certain link or channel becomes bad, it can select another one with higher RQI or LQI to forward packets. On the other hand, LEAST does not offer such flexibility so traffic cannot get through so quickly, resulting in decreased fairness for distant nodes. PROBE only has the option to dynamically select routes but not the channel so its delay fairness lies in between the other two protocols. Data with stricter delay requirement obtain better fairness because they are forwarded in a timely manner.

Similar case happens when reliability fairness is taken into account as shown in Fig. 13b. With options to dynamically choose routes (based on RQI) and channels (based on channel quality), MQMAC-CQBR-HSi gives the best reliability fairness, while PROBE has the medium performance with only one option for dynamic route selection and LEAST provides the worst fairness due to the inflexibility w.r.t. both options. For a similar reason as above, our solution provides better fairness to vital signs with stricter delay requirement.

5.4 Effects on vital signs

Finally, in order to see the effects of our solution on the patient's vital signs, we compared the received signals with the generated one. We plot a sample of the received ECG signals at the sink from the same node

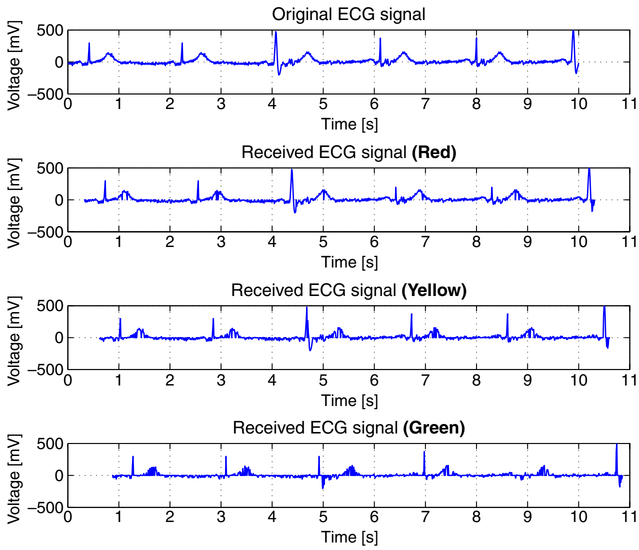


Fig. 14 ECG signals ($N_B = 14$)

in three different states when N_B is 14, as depicted in Fig. 14. The delay of these patient signs can be found by looking at the delay on time axis, while the error due to loss of data can be identified by noticing the voltage of the signals. Compared with the original signal, it is clear that 'Red' patient experiences the least loss and delay, while 'Green' patient has the largest loss and delay. Sign errors for different classes due to packet loss is compared in Fig. 15.

Note that the received signs are raw received data and the lost data are padded with zeros. Sophisticated signal processing techniques can be applied for better signal recovery. In this paper we focus on providing a solution for in-hospital networks so using better signal processing techniques is left as future work.

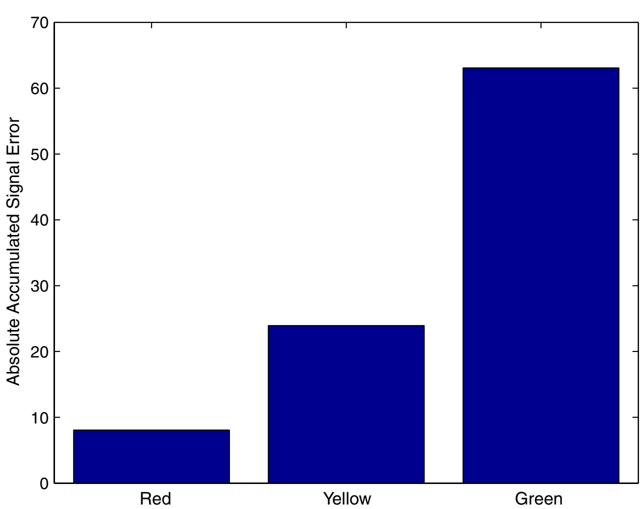


Fig. 15 Error of the received ECG signals

6 Conclusions and future work

Many solutions to avoid interference and to guarantee Quality Of Service (QoS) requirements have been proposed for wireless sensor networks in the past years. These solutions, however, are not tailored for medical applications for healthcare monitoring, where vital signs should be forwarded to the medical team reliably within a specified delay in order to save lives in critical situations such as in-hospital triage.

In this article, we proposed, implemented, and evaluated a novel in-network solution to prioritize the transmission of patient vital signs for in-hospital triage using wireless body area networks. A cross-layer communication solution jointly considering three functionalities—routing, medium access control, and scheduling—was designed to support the interference-aware prioritization services based on both patient condition and healthcare data content. By providing better wireless channels to patients who need more immediate services and scheduling packets based on the patient's condition and data requirements, our solution offers higher reliability and lower delay for patients who need immediate medical treatment. Field experiments show that our solution offers better performance than conventional wireless protocols.

Our proposed solution provides a networking solution to forward vital signs to the sink(s) reliably while guaranteeing the QoS requirements, which depend on the patient's triage condition and the data type. Our solution is based on the assumption that patients have been categorized into three triage classes. This assumption can be removed by integrating with data aggregation techniques that can provide high-level assessment of the physiological status and movement activities using the sampled sensor readings.

Future work will be to evaluate the performance of our solution in real medical environments such as emergency rooms in the hospital. Moreover, algorithms to categorize patients into "red", "yellow" and "green" classes by jointly considering different types of vital signs will also be designed and developed. Advanced signal processing techniques will also be researched and applied to recover the lost data or even the original vital sign signals at the receiver.

References

1. Bahl P, Chandra R, Dunagan J (2004) SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 Ad-Hoc wireless networks. In: Proc. of ACM annual

international conference on mobile computing and networking (MobiCom), Philadelphia, PA, USA
2. Chen F, Zhai H, Fang Y (2008) An opportunistic MAC in multichannel multiradio wireless ad hoc networks. In: Proc. IEEE wireless communications and networking conference (WCNC), Las Vegas, NV, USA
3. Chen H, Black JP, Khan OZ, Jamshaid K (2008) Data-centric support of a smart walker in a ubiquitous-computing environment. In: Proc. of international workshop on systems and networking support for healthcare and assisted living environments (HealthNet), Breckenridge, CO, USA
4. Chiussi F, Sivaraman V (1998) Achieving high utilization in guaranteed services networks using early-deadline-first scheduling. In: Proc. of IEEE Sixth international workshop on quality of service (IWQoS), Napa, CA
5. Dabiri F, Vahdatpour A, Noshadi H, Hagopian H, Sarrafzadeh M (2008) Ubiquitous personal assistive system for neuropathy. In: Proc. of international workshop on systems and networking support for healthcare and assisted living environments (HealthNet), Breckenridge, CO, USA
6. Demers A, Keshav S, Shenker S (1989) Analysis and simulation of a fair queueing algorithm. In: Proc. of ACM SIGCOMM, pp 1–12
7. Georgiadis L, Guérin R, Peris V, Sivarajan KN (1996) Efficient network QoS provisioning based on per node traffic shaping. IEEE/ACM Trans Netw 4(4):482–501
8. Gidlund M, Wang G (2009) Uplink scheduling algorithms for QoS support in broadband wireless access networks. J Commun 4(2):133–142
9. Haykin S (2005) Cognitive radio: brain-empowered wireless communications. IEEE J Sel Areas Commun (JSAC) 23(2):201–220
10. IEEE Computer Society (2006) IEEE standard 802.15.4—2006: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). IEEE Computer Society, Washington, DC
11. IEEE Engineering in Medicine and Biology Society (2008) ISO/IEEE Standard 11073: health informatics—PoC medical device communication—Part 00101: guide–guidelines for the use of RF wireless technology. IEEE Engineering in Medicine and Biology Society, Piscataway
12. Jain R, Chiu D, Hawe W (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Tech. Rep. TR-301, DEC Research
13. Jonsrud GE (2006) Application note AN041 (Rev. 1.0)—CC2420 coexistence. Texas Instruments
14. Ko J, Gao T, Terzis A (2009) Empirical study of a medical sensor application in an urban emergency department. In: Proc. of international conference on body area networks (BodyNets)
15. Li H, Tan J (2007) Heartbeat driven medium access control for body sensor networks. In: Proc. of international workshop on systems and networking support for healthcare and assisted living environments (HealthNet), Puerto Rico, USA
16. Li H, Tan J (2008) Body sensor networks based sensor fusion for cardiovascular biosignal predications. In: Proc. of international workshop on systems and networking support for healthcare and assisted living environments (HealthNet), Breckenridge, CO, USA
17. Malan D, Fulford-Jones T, Welsh M, Moulton S (2004) CodeBlue: an Ad Hoc sensor network infrastructure for emergency medical care. In: Proc. of international workshop on wearable and implantable body sensor networks, London, UK
18. Maróti M, Kusy B, Simon G, Lédeczi A (2004) The flooding time synchronization protocol. In: Proc. of international conference on embedded networked sensor systems (Sensys)
19. Mitola J, Maguire G (1999) Cognitive radio: making software radios more personal. IEEE Pers Commun 6(4):13–18
20. Mo J, So HSW, Walrand J (2008) Comparison of multichannel MAC protocols. IEEE Trans Mob Comput 7(1):50–65
21. Norgall T, Schmidt R, vod der Grum T (2004) Body area network—a key infrastructure element for patient centered telemedicine. Stud Health Technol Inform 108:142–148
22. Parekh AK, Gallager RG (1993) A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Trans Netw 1(3):344–357
23. Parekh AK, Gallagher RG (1994) A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. IEEE/ACM Trans Netw 2(2):137–150
24. Pompili D, Vuran MC, Melodia T (2006) Cross-layer design in wireless sensor networks. In: Mahalik NP (ed) Book on sensor network and configuration: fundamentals, techniques, platforms, and experiments. Springer, Germany
25. Silberberg J (1993) Performance degradation of electronic medical devices due to electromagnetic interference. Compliance Eng 10(5):25–39
26. So HW, Walrand J, Mo J (2007) McMAC: a parallel rendezvous multi-channel MAC protocol. In: Proc. of IEEE wireless communications and networking conference (WCNC), Hong Kong
27. So J, Vaidya NH (2004) Multi-channel MAC for Ad Hoc networks: handling multi-channel hidden terminals using a single transceiver. In: Proc. of ACM international symposium on mobile ad hoc networking and computing (MobiHoc), Tokyo, Japan
28. Tzamaloukas A, Garcia-Luna-Aceves JJ (2001) A receiver-initiated collision-avoidance protocol for multi-channel networks. In: Proc. of IEEE twentieth annual joint conference of the IEEE computer and communications societies (INFOCOM), Anchorage, Alaska, USA
29. Varkey JP, Pompili D (2009) Movement recognition using body area networks. In: Proc. of IEEE global telecommunications conference (GLOBECOM)
30. Varshney U (2008) Bluetooth 2.1 based emergency data delivery system in healthnet. In: Proc. of IEEE wireless communications and networking conference (WCNC), Las Vegas, NV
31. Varshney U (2008) Improving wireless health monitoring using incentive-based router cooperation. IEEE Computer 41(5):56–62
32. Voigt T, Osterlind F, Dunkels A (2008) Improving sensor network robustness with multi-channel convergecast. In: Proceedings of 2nd ERCIM workshop on e-mobility, Tampere, Finland
33. Wang H, Zhou H, Qin H (2008) Overview of multi-channel MAC protocols in wireless networks. In: Proc. of international conference on wireless communications, networking and mobile computing (WiCOM), Dalian, China
34. Wang JP, Abolhasan M, Safaei F, Franklin D (2007) A survey on control separation techniques in multi-radio multi-channel MAC protocols. In: Proc. of international symposium on communications and information technologies (ISCIT), Sydney, Australia
35. Wang Y, Fan L, He D, Tafazolli R (2008) Solution to weight-adaptive fair queuing. IEEE Electron Lett 44(5):385–386
36. Weisenberg J, Cuddihy P, Rajiv V (2008) Augmenting motion sensing to improve detection of periods of unusual

inactivity. In: Proc. of international workshop on systems and networking support for healthcare and assisted living environments (HealthNet), Breckenridge, CO, USA
37. Won C, Youn JH, Ali H, Sharif H, Deogun J (2005) Adaptive radio channel allocation for supporting coexistence of 802.15.4 and 802.11b. In: Proc. of IEEE vehicular technology conference (VTC), Dallas, TX, USA
38. Wongthavarawat K, Ganz A (2003) Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. Int J Commun Syst (Wiley) 16:81–96

Baozhi Chen received his B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2000. He received his M.S. degree from Columbia University in New York City in 2003. He is now pursuing the Ph.D. degree in Electrical and Computer Engineering at Rutgers, the State University of New Jersey—New Brunswick, under the supervision of Prof. D. Pompili. His research interests lie in underwater communications and networking, wireless communications and networking, with emphasis on cross-layer optimization in wireless ad hoc networks, wireless sensor networks, and wireless body-area networks. He is an IEEE and ACM student member.

Dario Pompili joined the faculty of the Department of Electrical and Computer Engineering at Rutgers, The State University of New Jersey, as Assistant Professor in Fall 2007 where he is the Director of the Cyber Physical Systems Laboratory (CPS Lab). He is also Site Director of the Center for Autonomic Computing (CAC), a NSF Research Center funded by the I/UCRC program that combines resources from universities, private companies, and the federal government to conduct fundamental research on making computer systems and applications more reliable, secure, and efficient. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in June 2007 after working at the Broadband Wireless Networking Laboratory (BWN-Lab) directed by Prof. I. F. Akyildiz. In 2005, he was awarded Georgia Institute of Technology BWN-Lab Researcher of the Year for "outstanding contributions and professional achievements." He had previously received his 'Laurea' (integrated B.S. and M.S.) and Doctorate degrees in Telecommunications Engineering and System Engineering from the University of Rome "La Sapienza," Italy, in 2001 and 2004, respectively. His research interests include ad hoc and sensor networks, underwater acoustic communications, wireless sensor and actor networks, and network optimization and control. He is author and co-author of many influential papers in these fields. He is in the editorial board of Ad Hoc Networks (Elsevier) journal and in the technical program committee of several leading conferences on networking. He is also member of the IEEE Communications Society and the ACM.