User Behavior Driven MAC Scheduling for Body Sensor Networks: A Cross-layer Approach

Mao V. Ngo, *Student Member, IEEE,* Quang Duy La, *Member, IEEE,* Derek Leong, Tony Q.S. Quek, *Fellow, IEEE,* and Hyundong Shin, *Senior Member, IEEE*

***Abstract*—In deploying body sensor networks (BSNs), sampling rates might be dynamically tuned to fit application requirements (e.g., monitoring patients' different activities), which helps conserving energy for battery-powered sensors. However, this results in variable data rates among sensors, which further requires an efficient resource allocation to maintain reliable transmission accommodating all traffic loads. We thereby address this joint problem of transmission reliability and energy efficiency, by proposing a BSN system that autonomously detects user behaviors, which in turn trigger dynamic sampling and resource scheduling via an adaptive MAC scheduling scheme. This cross-layer scheme uses time-slotted channel hopping (TSCH) in IEEE 802.15.4, which is a reliable low-power MAC protocol. Specifically, the proposed solution determines the best TSCH slotframe length for specific application requirements, the number of timeslots to be added/removed according to dynamic sampling rates, and then allocates timeslots via an equally spaced timeslot allocation algorithm. We implement our proposed approach on a BSN testbed, which features both state-of-the-art hardware and software architecture. Experimental results are conducted to evaluate our proposed solution in terms of throughput, packet delivery ratio, and energy per bit, which demonstrates that our cross-layer solution ensures reliable data transmission and energy efficiency compared to existing techniques.**

***Index Terms*—Body sensor networks (BSN), adaptive scheduling, media access control (MAC), time slotted channel hopping (TSCH), IEEE 802.15.4.**

I. Introduction

BODY sensor network (BSN) is a promising platform for efficient, low-cost, and comfortable healthcare monitoring systems, with such applications as monitoring the daily conditions of a patient, and providing support for elderly people [1]. Among many low-power wireless communications standards, IEEE 802.15.4 is widely adopted in BSNs for healthcare applications [2]–[5]. In recent years, a great deal of research has focused on medium-access control (MAC) for BSNs to improve energy efficiency, and prolong battery life of wearable sensors. MAC protocols of IEEE 802.15.4 [6] are mainly categorized into two methods, i.e., contention-based and schedule-based protocols. Even though the contention-based method, based on carrier-sense multiple access with collision avoidance (CSMA-CA), is traditionally utilized in many BSN systems [4], [5], [7], it faces several problems such as multi-path fading and interference from other devices using the same 2.4 GHz band [4], [8]. Instead of using only a single channel for communication as in CSMA-CA, an alternative MAC protocol called *time-slotted channel hopping* (TSCH) has been introduced in the latest revision of IEEE 802.15.4 [6] (published 2015). TSCH protocol, which is based on time-slotted access, provides deterministic latency for applications and mitigates the effects of interference. Moreover, when combining with channel-hopping mechanism, TSCH can reduce multi-path fading, ensuring reliability of wireless communications in the presence of other devices using 2.4 GHz. As such, TSCH is well-suited for monitoring applications where sensors transmit data periodically, in proximity to other wireless systems. While TSCH has been gradually adopted in many IoT applications [9]–[11], it is mostly employed for very low data rates (typically one or two packets per minute) compared to patient monitoring which requires sensors to dynamically sample and transmit back to gateway (i.e., the sending rate) at several times per second. To the best of our knowledge, there has not been any research on using TSCH for health-monitoring BSNs under such a high date rate constraint, besides our earlier paper [2].

A challenge of health-monitoring applications with high data rate constraints is that sensors' batteries will be used up quickly due to intensive sampling and communication workload, thus necessitating energy and workload reduction through adaptive sampling driven by user's behaviors [12]. Consider, for example, monitoring a heart disease patient undergoing cardiac rehabilitation with battery-powered electrocardiogram (ECG) sensors on the patients body. Instead of sampling data at frequent fixed intervals, resulting in waking up too often and unnecessary energy usage, the ECG can take measurements at a low sampling rate when the patient is sedentary with a low risk of a heart attack or cardiac event. However, when the patient exhibits a more vigorous behavior with higher risk of a cardiac event, the ECG can switch to

Manuscript received December 20, 2018; revised March 27, 2019; accepted May 04, 2019. This work was supported in part by the SUTD-ZJU Research Collaboration under Grant SUTD-ZJU/RES/01/2016 and the SUTD-ZJU Research Collaboration under Grant SUTD-ZJU/RES/05/2016. The work of M. V. Ngo was supported in part by the Singapore International Graduate Award (SINGA) A*STAR scholarship. The associate editor coordinating the review of this paper and approving it for publication was Dr. Giancarlo Fortino. (*Corresponding author: Tony Q.S. Quek.*)

M. V. Ngo is with the Singapore University of Technology and Design, Singapore 487372, and also with the Institute for Infocomm Research (I^2R), A*STAR, Singapore (e-mail: vanmao_ngo@mymail.sutd.edu.sg).

Q. D. La was with the Singapore University of Technology and Design, Singapore 487372 (e-mail: duy.q.la@ieee.org).

D. Leong was with the Institute for Infocomm Research (I^2R), A*STAR, Singapore (e-mail: dleong@i2r.a-star.edu.sg).

T. Q. S. Quek is with the Singapore University of Technology and Design, Singapore 487372, and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do, 17104, Korea (e-mail: tonyquek@sutd.edu.sg).

H. Shin is with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do, 17104, Korea (email: hshin@khu.ac.kr)

a higher sampling rate, so that high-resolution data can be collected for more complex analysis, to provide more accurate observations or predictions. Thus, incorporating an adaptive sampling rate mechanism [2], [12] to BSNs provides several benefits, where we may wish to expend more energy only when high-resolution sensor data is necessary. Similar to [12], we use *human activities* as the basis for *user behaviors* [1] to map with different sampling rates. Leveraging results in human activity recognition (HAR) which is a basic application of BSNs [13], [14], we build a dynamic sampling rate mechanism for different types of healthcare sensors according to specific application requirements.

The default TSCH of IEEE 802.15.4 [6] does not efficiently handle dynamic network traffic loads [9]–[11] which are generated from sensors' adaptive sampling rates. To tackle this issue, an adaptive MAC scheduling scheme will be proposed. Given the available user behaviors extracted from application layer (i.e., HAR) which are then mapped to various corresponding sampling rates for sensors, the proposed scheme will simultaneously change sampling rates of the sensors and dynamically schedule the transmissions in MAC layer based on new traffic demands generated by the new sampling rates. The detailed MAC layer scheduling, on top of the TSCH protocol, will be done systematically, with the goals of achieving energy efficiency while maintaining reliable data transmission (i.e., meeting the application's bandwidth requirements). While part of this work has been previously presented in our earlier work [2], the current paper significantly extends [2] in order to make the health-monitoring BSN-based system more comprehensive, via three key aspects. Firstly, the TSCH slotframe length is determined to accommodate the minimum requirements for different sending rates. Secondly, the number of timeslots to be added or removed for each sensor is evaluated in case it increases or decreases its sending rate according to detected user behaviors. This is a non-trivial problem as allocating the exact amount of requested bandwidth to sensors could easily exceed the maximum supported bandwidth of the system, if some high-priority sensors generate too many data. The proposed scheme can account for such an overload scenario, by considering several factors such as individual sensors' sending rates and overall system fairness, to decide on the most appropriate timeslot allocation. Finally, the actual placement of the extra timeslots in the TSCH slotframe is dealt with via an equally spaced timeslot allocation algorithm, to achieve a smooth sensor data stream to the gateway.

In order to quickly respond to real-time changes in user behaviors, some of the data processing and decision making tasks could be done at or nearby the sources of data (i.e., sensor nodes) rather than in the remote cloud. This approach is aligned with the emerging *edge computing*—the next evolution of computing paradigm, which brings computation, storage resources to the edge of the networks [15]. In the proposed system, we will adopt this principle, by placing a user behavior classifier right at the gateway device for extracting knowledge from the application layer. Based on this real-time knowledge, MAC scheduling decision can be made without sending data to the cloud which could incur long delays, hogging of bandwidth, and waste of power. As a result, the proposed system is capable of not only effectively monitor users in real time, but also reducing network latency and saving bandwidth and power.

Overall, our proposed approach constitutes a cross-layer solution, exploiting the detected user behaviors at the application layer in order to automatically change the sensors' sampling rates and perform dynamic scheduling in the TSCH MAC layer simultaneously. We present an end-to-end system architecture and build a testbed to evaluate the performance of the proposed scheme. Our testbed features BSN nodes and a gateway using Contiki-OS on the OpenMote-CC2538 platform. Notably, we employ *edge computing* design principles by placing the data processing and decision making tasks right at the gateway device. In summary, our main contributions in this paper are as follows.

- Present an end-to-end BSN system based on TSCH for healthcare monitoring, which is first in deploying TSCH for such a high data-rate application.
- Propose an adaptive MAC scheduling scheme for TSCH which employs a cross-layer approach, i.e., leveraging detected user behaviors at application layer to dynamically adjust sensor's sampling rates, then rescheduling TSCH timeslots accordingly while maintaining fairness and bandwidth requirement.
- Develop a testbed to implement and evaluate the proposed scheme against existing techniques, in terms of throughput, packet delivery ratio, and energy per bit. Results demonstrate that our cross-layer solution ensures reliable data transmission and energy efficiency.

The rest of the paper is organized as follows. Section II reviews existing work in adaptive MAC scheduling schemes for wireless sensor networks (WSNs). In Section III, we present an overall architecture for the healthcare monitoring system. The adaptive MAC scheduling scheme is described in Section IV. Experimental setup is presented in Section V-A, with the corresponding results shown in Section V-B. Finally, the conclusion is in Section VI.

II. Related Work

MAC protocols of IEEE 802.15.4 standard [6] are categorized into contention-based and schedule-based. For contention-based MAC scheduling, there have recently been notable works which applied machine learning techniques, e.g., Q-leaning method for adaptation of radio sleep-wakeup scheduling to network traffic conditions [16], [17]. This section will nevertheless focus on schedule-based MAC-scheduling, or in particular, the adaptive TSCH protocols in BSNs and WSNs, due to the fact that TSCH protocol shows advantages over contention-based protocols in applications such as healthcare as aforementioned. In fact, the adaptive MAC scheduling schemes based on TSCH protocol can be categorized into decentralized and centralized ones.

Decentralized MAC scheduling schemes were investigated in [9], [11], [18]–[20]. Morell *et al.* [18] adopted the technique

[1]Henceforth, we use "human/user activities" and "user behaviors" interchangeably

of multi-protocol label switching (MPLS) to WSN, then proposed an adaptive MAC scheduler based on the bandwidth, latency, and power requirements of each sensor. Accettura *et al.* [11] proposed decentralized traffic-aware scheduling for the TSCH protocol in multi-hop WSNs. However, this approach faces significant overheads for updating traffic information of each sensor, and has to reschedule the whole network when a new node joins or the traffic load of a node changes. Duquennoy *et al.* [19] proposed a decentralized TSCH-MAC scheduling scheme called Orchestra for multi-hop industrial WSNs. Peng *et al.* [20] proposed an adaptive TSCH (A-TSCH) that uses three dedicated timeslots per slotframe for detecting blacklist channels, resulting in reducing bandwidth of the whole BSN. Palattella *et al.* [9] proposed a decentralized TSCH scheduling scheme for adapting scheduled bandwidth to network traffic requirements. However, these above approaches did not work well for BSNs which support dynamic network traffic loads because reservation bandwidths for multiple sensors simultaneously in a decentralized manner results in high collisions.

Some centralized MAC schedulers were investigated in [21]–[23]. Palattella *et al.* [21] proposed a centralized traffic-aware TSCH scheduling scheme that uses graph coloring techniques to allocate timeslots and channels for a multi-hop WSNs. However, this scheme is unsuitable for adaptive sending rates because the whole slotframe needs to be appropriately reallocated even if only one sensor changes its sending rate, which can disturb the other sensors. Choi *et al.* [22] proposed a centralized TSCH scheduler that reduces overhead of control messages, and avoids rescheduling the entire slotframe when allocating and deallocating timeslots. Jin *et al.* [23] developed a centralized MAC scheduler for TSCH to allocate as many timeslots in a slotframe as possible, aiming to achieve low latency in multi-hop WSNs. However, it allocates too many backup timeslots, leading to the problem of resource hogging.

A. Comparison with our work

These previous works focus on multi-hop WSNs while not paying attention to the dynamic sampling and adaptive sending rates of the BSN. They are therefore not suitable for BSNs with a star topology in which frequency diversity is hard to exploit. In addition, these approaches incur overheads of reporting bandwidth requirements from sensors to gateway, and cannot proactively allocate enough bandwidth right after the sensors switch to sample higher rates and then send at higher sending rates. Moreover, the above works did not address the case of the traffic demand exceeding the network bandwidth capacity. Although few works [9], [21], [23] consider adaptive sending rates, they do not proactively adapt to the network traffic loads generated by the adaptive sampling rate mechanism. Instead of assuming that we know in advance traffic information (e.g., bandwidth requirements) as [9], [11], [21], our cross-layer approach with the information from the application layer (i.e., user behavior) can proactively change the sensors sampling rates, and dynamically allocate MAC scheduling scheme to adapt with the network traffic demands. In addition, with the cross-layer approach, we can reduce overhead of control message for reporting traffic information from sensor nodes since the gateway already has holistic statistical information of the BSN. Overall, our approach fills the existing gap left by current works in the literature by presenting an end-to-end BSN system for healthcare application, employing a cross-layer MAC scheduling approach that can automatically adjust the sensor sampling rates and dynamically allocate an appropriate number of timeslots for requesting sensors, based on the requirements from the application layer (i.e., detected user behaviors). This approach has not been considered by existing works, to the best of our knowledge.

III. User Behavior Driven Health-Monitoring System

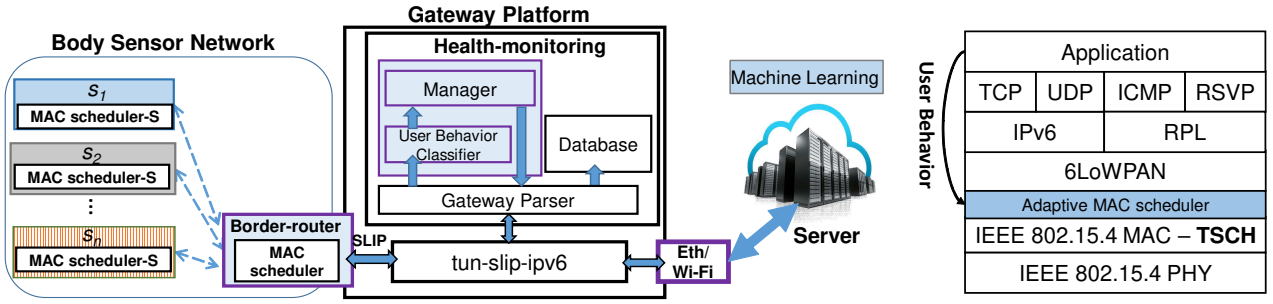
Fig. 1a shows the software architecture of the health-monitoring system, which consists of three main parts: a BSN with star topology, a gateway, and a server.

The **BSN**, which may correspond to an individual patient whose conditions are being monitored, consists of several healthcare sensors connected to a gateway device via one-hop communication. The wearable sensors periodically sample the physiological signals of the wearer, such as ECG, heart rate, blood pressure, body temperature, activity, and oxygen saturation (SpO_2), and then send the data to the gateway using low-power wireless communications, such as IEEE 802.15.4.

The **gateway** normally possesses greater storage and processing power than sensor devices. The gateway receives raw sampled data from all sensors at the *border-router* via a serial line Internet protocol (*SLIP*) interface. The raw sampled data, which is passed through the *tun-slip-ipv6* module [24] and is parsed to readable format by the *gateway-parser*, can be analyzed at the *health-monitoring* module, stored in the *database*, or forwarded to the server (which happens only if the gateway is charging). At the health-monitoring module, the *user behavior classifier* automatically detects user behaviors in the form of human activities. If the detected user behavior is changed, e.g., from a normal to a high-risk activity, the *manager* module issues instructions to the border router with a combined decision that consists of *new sampling rates* and *numbers of timeslots to be added/removed* for the appropriate sensors based on their new required bandwidths (Section IV-C). From the number of timeslots to be added/removed, the border-router chooses actual allocated timeslots by using an *equally spaced timeslot allocation algorithm* (Section IV-D). Then, the border-router cooperates with the sensors to change their sampling rates; and the *MAC scheduler* module at the border router cooperates with the *MAC scheduler-S* modules at the sensors to add or remove timeslots. In summary, to provide the required bandwidths, the gateway employs a cross-layer approach using the user behaviors extracted from application layer as a driven input to an *adaptive MAC scheduler* located right on top of the IEEE 802.15.4-TSCH MAC layer, as shown in Fig. 1b.

The **server** is used for building the user behavior classifier as a machine learning model based on our collected data, and analyzing the network performance. The details of training the machine learning model is presented in Section V-A4.

In traditional BSN-based and other IoT systems, sensory data analysis is typically done in the remote cloud. How-



(a) System software architecture (b) TSCH stack with cross-layer approach

Fig. 1. (a) System software architecture, and (b) the TSCH protocol stack incorporating our cross-layer approach for user behavior driven health-monitoring system.

ever, in the present BSN system for healthcare monitoring, there are emergency situations which require decisions to be responded promptly. To tackle this requirement, we employ edge computing's design principles. Specifically, the decision making modules (i.e., manager and user behavior classifier modules) are placed in the gateway device, right next to the sensor devices. This will allow for real-time detection of user behaviors without sending data back to the remote cloud. Subsequently, MAC scheduling decision can also be made simultaneously without incurring long delays. Therefore, the system can respond quickly to changes in user behaviors and ensure reliable transfer of sensor data in emergency situations.

IV. Adaptive MAC Scheduling Scheme

To save energy for BSNs in healthcare applications, we propose a user behavior driven MAC scheduling scheme, in conjunction with a dynamic sensor sampling rate mechanism, to adapt the required bandwidths of some sensors that increase or decrease their sending rates while ensuring reliable communications for all sensors (i.e., including sensors that do not change their sampling rates). Our MAC scheduling scheme is based on the time-slotted channel hopping (TSCH) protocol of IEEE 802.15.4 [6].

A. TSCH - Time Slotted Channel Hopping

In the TSCH protocol of IEEE 802.15.4, all nodes synchronize on a periodic *slotframe* which contains a number of timeslots. A timeslot can be assigned to a pair of transmitting and receiving nodes. A TSCH link between two nodes is described by two parameters: *timeslot index* in the slotframe and a *channel offset* that results in different communication frequency in the frequency hopping list. Fig. 2 shows an example of allocating timeslots to the border-router and three sensor nodes. There are two type of links – downlink and uplink. In a downlink timeslot, the border-router sends control packets to one destination sensor among associated sensors. In an uplink timeslot, the sensor node sends data packets to the border-router. There are two type of uplinks – dedicated and shared uplinks. The dedicated uplink is allocated to a single sender-receiver pair at a timeslot (i.e., contention-free); whereas the shared uplink is allocated to multiple senders transmitting to a receiver (i.e., contention-based with back-off mechanism). Fig. 2 presents an example of adding dedicated uplinks from sensor 3 to the border-router, which are denoted as TX with an arrow below, instead of TX(S) for a shared uplink initially allocated based on the sensor's MAC address when joining the network.

B. Slotframe Length

In the TSCH-based BSN, all sensors and the border-router are set up with the same slotframe length that is not changed during deployment. Hence, the system designers need to choose the slotframe length (denoted by $|\text{SF}|$) carefully according to the sending rate of each sensor. Consider a BSN that consists of N sensors under K different user behaviors (e.g., normal, urgent behaviors) which correspond with K sending rates, i.e., s_{ik}, where $i \in \{1,\ldots,N\}$ and $k \in \{1,\ldots,K\}$. Let us denote $s_i^* = \min_k\{s_{ik}\}$ as the minimum sending rate of sensor i under all user behaviors, which is also the initial sending rate of sensor i in the normal behavior. We denote N_{SF} as the *number of slotframes repeated per second* given by

$$N_{\text{SF}} = \frac{1}{|\text{SF}| \cdot |\text{TS}|}, \tag{1}$$

where $|\text{TS}|$ is timeslot duration in seconds. We set $|\text{TS}|$=10 ms as per the recommendation of IEEE 802.15.4 [6]. The original TSCH scheme allocates *only one timeslot per slotframe* to a sensor, so that the maximum sending rate of the sensor is not higher than the number of slotframes repeated per second. In addition, if N_{SF} is high enough to support the sensor with maximum sending rate, i.e., $N_{\text{SF}} \geq \max_i(s_i^*)$, then the other sensors with lower sending rates are also supported automatically. Hence, the constraint of slotframe length is given by

$$|\text{SF}| \leq \frac{1}{\max_i(s_i^*) \cdot |\text{TS}|}. \tag{2}$$

We set $|\text{SF}|$ equal to the upper bound for the flexibility of future allocation when the BSN requires more bandwidth, and reducing wake-up time of the transceivers at the allocated timeslots without sending data packet.

C. Determination of the Number of Timeslots

Suppose at a given time, each sensor i under a user behavior k has been allocated N_{ik} timeslots per slotframe. Let $N_{\text{free},k}$ be

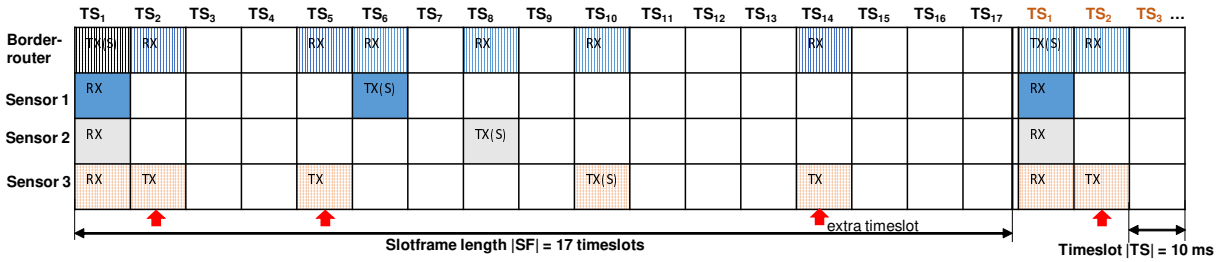


Fig. 2. Allocation of timeslots in a TSCH slotframe to sensors with equal space extra timeslot algorithm. The timeslot with an arrow below denotes the extra timeslot in a high-risk activity.

the total number of available (unused) timeslots in a slotframe at the user behavior k, which is given by

$$N_{\text{free},k} = |\text{SF}| - (\sum_{i=1}^{N} N_{ik} + 1), \tag{3}$$

where $(\sum_{i=1}^{N} N_{ik} + 1)$ is the total number of timeslots allocated to N associated sensors (uplinks) and the border-router (downlink). If the BSN already utilizes all timeslots, i.e., $N_{\text{free},k} = 0$, the system rejects any request for more timeslots. Otherwise, i.e., $N_{\text{free},k} > 0$, the system allocates extra timeslots as follows.

Suppose the BSN changes from the user behavior k to k', and the manager module at the gateway (refer to Fig. 1a) changes the sensors' sending rates from s_{ik} to $s_{ik'}$. Note that, there is no action required for the sensors that do not change their sending rates. We hence consider that sensor i either increases or decreases its sending rate.

- In the *decreasing sending rate* case, i.e., $N_{ik'} < N_{ik}$, we will release a number of timeslots to the pool of the available timeslots, i.e., $N_{ik',\text{remove}} = N_{ik} - N_{ik'}$.
- In the *increasing sending rate* case, i.e., $N_{ik'} > N_{ik}$, we need to allocate extra timeslots to sensor i, the number of which depends on the constraint of total available timeslots given by eq. (3), as well as fairness consideration, as follows.

Among all N sensors, as only a certain number of them increase their sending rates, we use l to denote the indices for these sensors, i.e., $s_{lk'} > s_{lk}, l \in \{1, \ldots, L\}$ and $L \leq N$. Let $N_{lk',\text{add}}$ denote the required number of extra timeslots for sensor l under user behavior k'. We further categorize the timeslot allocation process into two cases, as follows.

If the total timeslots to be added **do not exceed the system's resource capacity**, i.e., $\sum_{l=1}^{L} N_{lk',\text{add}} \leq N_{\text{free},k}$, all requesting sensors are given their number of required timeslots, i.e.,

$$N_{lk',\text{add}} = \left\lceil \frac{s_{lk'}}{N_{\text{SF}}} \right\rceil - N_{lk}, \quad \forall l \in \{1, \ldots, L\}. \tag{4}$$

Any remaining timeslots will be available for future requests in case of either a higher-risk user behavior occurring, or when a new sensor joins the BSN.

If the total timeslots to be added **exceed the system's resource capacity**, which is also called the *request overload regime*, i.e., $\sum_{l=1}^{L} N_{lk',\text{add}} > N_{\text{free},k} > 0$, not all requests can be satisfied. In such a case, we would like to maintain *fairness* among all sensors with increasing sending rates to make sure that resources are distributed evenly across the sensors. To this end, we propose a way to determine the number of extra timeslots for each requesting sensor that aims to achieve *approximately identical ratios* $r_{lk'}$ *between real and ideal throughputs* among these sensors. Here, throughput is defined as the total number of bits successfully transmitted per second (see also Sec. V-A2). Specifically, the throughput ratio is given as

$$r_{lk'} = \frac{TP_{lk'}^{\text{real}}}{TP_{lk'}^{\text{ideal}}} = \frac{N_{lk'}^{*} \cdot N_{\text{SF}} \cdot D}{s_{lk'} \cdot D} = \frac{N_{lk'}^{*} \cdot N_{\text{SF}}}{s_{lk'}}, \tag{5}$$

where $TP_{lk'}^{\text{real}}$, and $TP_{lk'}^{\text{ideal}}$ are the actual achievable throughput after timestlot allocation, and the ideal throughput assuming the required number of extra timeslots can be met, respectively; $N_{lk'}^{*}$ is the actual number of allocated timeslots (after allocation); and D is the size of a data packet in bits.

Inspired by Jain's fairness index [25], we introduce the fairness index F to quantify the fairness of our allocation mechanism in the request overload regime, where

$$F(\{r_{lk'}\}) = \frac{(\sum_{l=0}^{L} r_{lk'})^2}{L \cdot \sum_{l=1}^{L} {r_{lk'}}^2}. \tag{6}$$

The closer to 1 the fairness index F (an index of 1 is achieved by making all $r_{lk'}$ identical), the fairer the allocation among the requesting sensors. Thus, in the request overload regime, as we try to achieve the highest fairness index, the throughput ratios of the requesting sensors should be made similar according to the following condition:

$$\frac{N_{1k'}^{*}}{s_{1k'}} \approx \cdots \approx \frac{N_{Lk'}^{*}}{s_{Lk'}} \approx \frac{\sum_{l=1}^{L} N_{lk'}^{*}}{\sum_{l=1}^{L} s_{lk'}} = \frac{N_{\text{free},k} + \sum_{l=1}^{L} N_{lk}}{\sum_{l=1}^{L} s_{lk'}}. \tag{7}$$

The last equality of eq. (7) suggests that the sum of all the number of allocated timeslots to sensors in request overload regime is capped at the system's maximum capacity. From eq. (7), the new total number of allocated timeslots for sensor l under the new user behavior k' is given by

$$N_{lk'}^{*} = N_{lk'}^{o} + \delta_{lk'}, \text{ where} \tag{8}$$

$$N_{lk'}^{o} = \max\left\{ N_{lk}, \left\lfloor \frac{(N_{\text{free},k} + \sum_{l=1}^{L} N_{lk}) \cdot s_{lk'}}{\sum_{l=1}^{L} s_{lk'}} \right\rfloor \right\}. \tag{9}$$

After allocating to all requesting sensors as in eq. (9), if there are still some available timeslots remaining, i.e.,

ALGORITHM 1: Equally spaced timeslot allocation

Data: $t_{l,\text{cur}}, N_{lk',\text{add}}, |\text{SF}|$
Result: $\mathcal{S}$ // set of timeslots to be allocated
1 $\mathcal{S} \leftarrow \emptyset$; $step \leftarrow \left\lfloor \frac{|\text{SF}|}{N_{lk',\text{add}}} \right\rfloor$; $t_{\text{tem}} \leftarrow t_{l,\text{cur}}$; $u \leftarrow 0$;
2 **while** $|\mathcal{S}| < N_{lk',\text{add}}$ & $u < |\text{SF}|$ **do**
3 $t_{\text{tem}} \leftarrow (t_{\text{tem}} + step) \mod |\text{SF}|$;
4 **if** t_{tem} is available **then**
5 $S \leftarrow t_{\text{tem}}$;
6 **else**
7 $m \leftarrow 0$;
8 **while** $m < step$ **do**
9 $t_{\text{neighbor}} \leftarrow (t_{\text{tem}} \pm m) \mod |\text{SF}|$;
10 **if** t_{neighbor} is available **then**
11 $\mathcal{S} \leftarrow t_{\text{neighbor}}$;
12 **break;**
13 **end**
14 $m \leftarrow m + 1$;
15 **end**
16 **end**
17 $u \leftarrow u + 1$;
18 **end**

$N_{\text{free},k'} + \sum_{l=1}^{L} N_{lk} - \sum_{l=1}^{L} N^{o}_{lk'} > 0$, then we distribute these remaining timeslots among the requesting sensors in descending order of the sending rates. Hence, in eq. (8), the term $\delta_{lk'} \in \{0, 1\}$ indicates whether an extra timeslot is given to sensor l or not. Finally, the actual number of extra timeslots is given by $N_{lk',\text{add}} = N^{*}_{lk'} - N_{lk},\ l \in \{1, \ldots, L\}$. Note that as $N^{*}_{lk'} \geq N_{lk}$, the new allocation does not decrease bandwidth of the sensors that increase their sending rates.

D. Equally Spaced Timeslot Allocation Algorithm

In the previous sections, we specify how many extra timeslots $N_{lk',\text{add}}$ should be allocated for the sensor l at the new behavior k'. In this section, we present an equally spaced timeslot allocation algorithm to specify the exact placement, i.e., which *timeslot indices* for these timeslots, to provide a smooth sensor data stream to the gateway and the health-monitoring module.

Algorithm 1 describes the details of the timeslot allocation process. The equally spaced timeslot allocation algorithm is not trivial because the allocation has to consider sensor l's new placement in relation to its own existing timeslots and the other sensors'. The inputs of the algorithm are $t_{l,\text{cur}}$ which is the index of the pre-scheduled timeslot in TSCH slotframe currently dedicated to the sensor node l, $N_{lk',\text{add}}$ which is the required number of extra timeslots, and $|\text{SF}|$. The allocation attempts to select new timeslots that altogether are equally spaced, relative to the pre-scheduled timeslot, which can be done easily if the allocated timeslots is free as shown in lines 3 – 5. However, when a chosen timeslot is occupied, we carefully select the nearest available timeslot (i.e., at either some upper or lower index away from the initially attempted timeslot) as shown in lines 6 – 15. Fig. 2 gives an example of the allocation for sensor 3 in the TSCH slotframe with $|\text{SF}| = 17$ timeslots. As in Fig. 2, the current timeslot of sensor 3 is TS_{10}, i.e., $t_{l,\text{cur}} = 10$, and it requires to allocate four extra timeslots, i.e., $N_{l,\text{add}} = 4$. The output of the Algorithm 1 is $\mathcal{S} = \{14, 2, 5\}$, i.e., TS_{14}, TS_{2}, TS_{5} as shown in Fig. 2 with red arrows below. The complexity of Algorithm 1 is $\mathcal{O}(|\text{SF}|^2/N_{lk',\text{add}})$.

E. Cross-layer Adaptive MAC Scheduling Scheme

Combining all the previous subsections, we present a cross-layer adaptive MAC scheduling scheme based on user behavior. In the normal activity, the sensors sample at the normal sampling rates, and send with the normal sending rates. When the health-monitoring module detects a high-risk activity, it generates an "emergency" signal to the border-router which includes the following information: (i) list of the emergency sensors, (ii) their new sampling rates $s_{lk'}$, and (iii) the appropriate number of extra timeslots $N_{lk',\text{add}}$ that needs to be allocated to them, presented in Section IV-C. Based on this information, the MAC scheduler at the border-router runs the equally spaced timeslot algorithm procedure given by Algorithm 1 to find a set of extra timeslots for the emergency sensors. Subsequently, the border-router adds the RX timeslots to itself, and sends a control signal to each of these sensors with the following parameters: *signal type (i.e., add or remove), new sampling rate $s_{lk'}$, and set of added (or removed) timeslots*. Once the emergency sensor l gets the control signal, it adds the TX timeslots (by MAC scheduler-S module), increases its sampling rate to the new value $s_{lk'}$, and acknowledges the border-router in a piggy-back fashion.

When the high-risk activity is over, the emergency sensors reduce their sending rates to the original values, remove the extra TX timeslots, and inform the border-router about this change. Consequently, the border-router removes the appropriate RX timeslots to release these timeslots for future allocation.

V. Performance Evaluation

A. Experimental setup

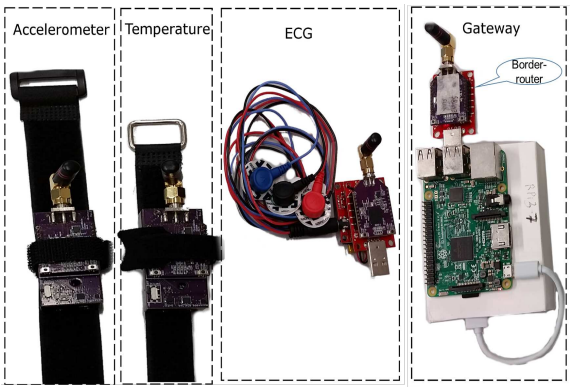


Fig. 3. Experimental testbed of the BSN.

1) Testbed and Evaluation Scenarios: Consider a healthcare scenario that a heart disease patient is undergoing cardiac rehabilitation. The health-monitoring BSN for the patient consists of three sensors and a gateway as shown in Fig. 3.

We consider three different types of **sensors**: accelerometer (ADXL346), temperature and humidity (SHT21), and ECG sensor plugged into a SparkFun AD8232 board. These sensors

use the OpenMote-CC2538[2] platform as the radio communication module.

Raspberry-Pi 3 is used as the **gateway** which handles several tasks such as collecting sensor data, processing, storing data locally, and making a decision of simultaneously changing sampling rate and adjusting network bandwidth accordingly. The sensors are connected to the gateway via the *border-router* which is another OpenMote-CC2538 platform plugged into the gateway via serial connection. The sensors and border-router use the operating system Contiki OS [24] for programming, and TSCH for the MAC layer and (RPL + 6LowPAN) for the network layer (see Fig. 1b). We modify the `tunslip6` module in the Contiki OS to handle triggering commands from the application running on the gateway to the MAC scheduler module running on the border-router (refer to Fig. 1a).

We will evaluate the BSN system under different user behaviors that map to different user activities and the sensors' sending rates, which are given in Table I. In each row of Table I which describes each user behavior, each sensor samples data at a frequency f (Hz) which in turn sends fixed-size data packets at a sending rate s (packet/s) to the gateway. Note that the frequency f is higher than the sending rate s; hence, each sensor must store raw sampled data in a buffer, then only send a data packet if the size of accumulative sampled sensor data reaches the size of the sensor's sending packet. Specifically, the packet size[3] of accelerometer, temperature, and ECG sensors are 115 bytes, 63 bytes, and 83 bytes, respectively. In monitoring a heart disease patient, we let the accelerometer and ECG sensors sample at a low rate when the patient's behaviors are *normal* (e.g., while sitting or standing). However, when the patient exercises more vigorous activities, with higher risk of a cardiac event (e.g., walking and running, sub-divided into *urgent-medium* and *urgent-high* behaviors, respectively), the sensors switch to higher sampling rates, obtaining higher-resolution data for more complex analysis.

In addition, we conducted separate experiments in the *request overload regime* as given in the last row in Table I. In these experiments, the user changes from the normal behavior to the "overload-regime" (by a manual trigger at the border-router) that samples extremely high sampling rates[4] generating extremely high traffic loads which overload the bandwidth capacities of the BSN.

2) Performance Metrics: We use the following metrics to evaluate the performance of the BSN.

Packet delivery ratio (PDR) is an indicator for the network *reliability*. PDR is defined as the fraction of data packets successfully delivered from the sensor nodes to the gateway, i.e., PDR $= C_{\mathrm{P}}^{*}/C_{\mathrm{P}}$, where C_{P} and C_{P}^{*} be the total number of transmitted data packets and the number of successfully delivered data packets, respectively.

[2]www.openmote.com

[3]The packet size includes sensor data size and network header size (i.e., 6LowPAN header).

[4]In the implementation, due to hardware limitation on the sampling rate for the three sensors, when their desired sampling rates exceed the sensor's capability (as marked * in Table I), we send dummy packets having the same length as the usual packet to measure network performance.

TABLE I
USER BEHAVIOR PROFILES WITH SAMPLING RATE f (HZ) AND SENDING RATE s (PACKET/S)

User behavior	User activity	Accelerometer		Temperature		ECG	
		f	s	f	s	f	s
Normal	Sitting/Standing	32	4	2	1	64	2
Urgent-medium	Walking	64	8	4	2	512	16
Urgent-high	Running	128	16	8	4	1024	32
Overload regime	*Manual trigger*	256^{*}	32	32^{*}	32	2048^{*}	64

Throughput (TP) is the number of information bits forwarded successfully from the source to the targeted destination per unit of time, i.e., $TP = (C_{\mathrm{P}}^{*} \cdot D)/t$.

Energy-per-bit (EPB) is the ratio of the communication power consumption P to the throughput TP, i.e., $EPB = P/TP$, which measures the *energy efficiency* of the transmission and is measured by joule/bit. The communication power consumption P is defined as the averaged power consumption in mW of the sensor node across four power states: transmission (TX) state, listening (RX) state, CPU active state, and low-power mode (LPM) state [8], [26]. The power consumption is given by

$$P = \frac{V(t_{\mathrm{TX}} I_{\mathrm{TX}} + t_{\mathrm{RX}} I_{\mathrm{RX}} + t_{\mathrm{CPU}} I_{\mathrm{CPU}} + t_{\mathrm{LPM}} I_{\mathrm{LPM}})}{t_{\mathrm{CPU}} + t_{\mathrm{LPM}}}, \quad (10)$$

where $I_{\mathrm{TX}} = 24.0\,\mathrm{mA}$, $I_{\mathrm{RX}} = 20.0\,\mathrm{mA}$, $I_{\mathrm{CPU}} = 7.0\,\mathrm{mA}$, and $I_{\mathrm{LPM}} = 0.04\,\mathrm{mA}$ are the average rated currents of the respective states which are obtained from data sheet of SoC Texas Instrument CC2538[5]. The average voltage $V = 3\,\mathrm{V}$. Also, t_{TX}, t_{RX}, t_{CPU}, and t_{LPM} are the elapsed time of these appropriate power states. In order to measure the elapsed time of different power state, we use the Contiki Powertrace [26] built-in power profile.

3) Comparison of MAC Scheduling Schemes: In order to evaluate the network performance, we implement the following three scheduling schemes:

Orchestra scheme [19] allocates a bidirectional link between a sensor and the border-router. For example, as shown in Fig. 2, the downlink and uplink between sensor 1 and the border-router are allocated at TS_1 and TS_6, respectively. In order to reduce transmission collision between the associated sensors, we deliberately choose the sensors with different last byte of MAC addresses, which creates uplinks from sensors to the gateway at different timeslots. The Orchestra scheme is the baseline of the following two schemes.

Static scheme uniformly allocates all timeslots in a slotframe to all associated sensors in dedicated-timeslot fashion at initialization of the BSN; and all the allocated timeslots are fixed after the initial allocation. Specifically, each sensor is first allocated with $\left\lfloor \frac{|\mathrm{SF}|-1}{N} \right\rfloor$ timeslots. The remaining timeslots are randomly allocated to some sensors in the BSN.

Proposed scheme is a dynamic timeslot allocation scheme in which each sensor adds/removes an appropriate number of extra dedicated timeslots based on the required bandwidth when a high-risk activity occurs/finishes, which is described in Section IV. To build the proposed scheme, we inherit

[5]http://www.ti.com/lit/ds/symlink/cc2538.pdf

the implementation of TSCH, and the Orchestra scheduling scheme of the Contiki OS, and build a thin layer on top of the TSCH (refer to Fig. 1b) to add, remove extra timeslots based on instructions from the border-router (which gets from the application layer at the gateway).

In our experiments, we chose a common slotframe length of all mentioned schemes for a fair comparison purpose. According to the Orchestra scheduling scheme [19], the slotframe length is a prime number. Based on the sending rates given in Table I, we select the largest prime number in the range of slotframe length defined in eq. (2), i.e., $|\text{SF}| = 23$ timeslots.

4) User Behavior Classifier: In order to build a classifier for detecting user behaviors (i.e., HAR), we perform the following steps:

Sensor Data Collection: The triaxial *accelerometer* sensor is worn on the left wrist of the user via a strapped band, and samples at a frequency of 32 Hz which is similar to [12]. Data was collected for each of the four activities (i.e., sitting, standing, walking, and running) on two male adults of average build. Each activity was performed for 5 minutes with precisely recorded timestamps, which are used for labeling. The dataset contains 76800 samples with each sample having 3 dimensions.

Feature Extraction: Data samples of the same dimension are grouped into a two-second window [2], [13] that is not too long but also not too short to capture enough feature data for human activities. Two consecutive windows overlap by 50%. Then, we carry out the feature extraction with a set of basic statical features [12], [13]: minimum, maximum, mean, variance, skewness and kurtosis, which are computed across all samples in each window.

Training Machine Learning Models: The following traditional machine learning algorithms[6] are utilized: support vector machine (SVM), decision tree, Gaussian naive Bayes (GNB). We follow the standard practice of dividing the entire data set into a training set (70%) and a test set (30%) via random sampling. We use the library scikit-learn [27] in Python to implement these machine learning models. The accuracy of the decision tree model is 99.50% while that of the SVM and GNB models are 98.63% and 98.10%, respectively. Based on the empirical results, in subsequent experiments we use the *decision tree* model for classification.

In order to make reproducible experimental results, we replay the collected data (3 minutes for each activity) with three consecutive activities, i.e., standing, walking, and running, to trigger the change of user behavior. Note that the machine learning model was trained with the normal sampling rate; however, the system running in the urgent behaviors get the higher-resolution input data. In this work, we simply down sample input data to the normal sampling rate before feeding it to the classifier. The higher-resolution data is stored in the database and analyzed offline for further studying.

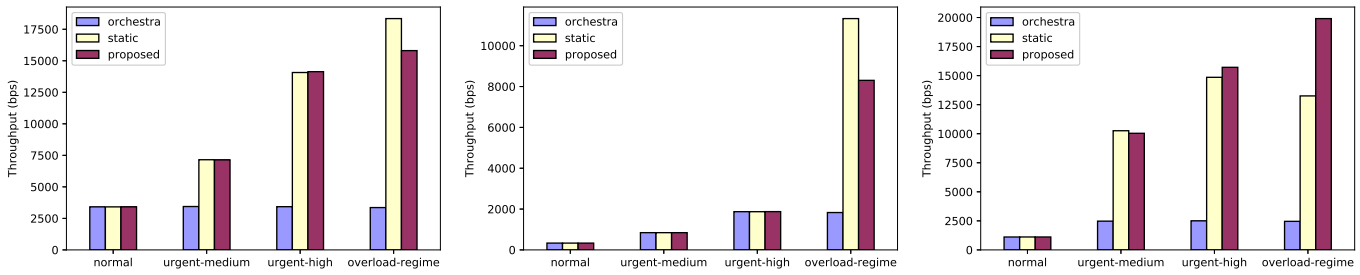
B. Experimental Results

We evaluate the health-monitoring BSN under different user behaviors in term of throughput, PDR, and EBP. The measurement results of these 3 metrics are obtained and shown in Figs. 4, 5, and 6, respectively. Note also that all measurements for accelerometer, temperature, and ECG sensors are grouped together in columns (a), (b), and (c), respectively; and all are plotted against different behavior profiles (normal, urgent-medium, etc.).

We first look at the results for **throughput**. At first glance, Fig. 4 shows that throughput generally increases as behavior shifts from normal to urgent-medium, urgent-high, and towards the request overload regime. An exception is the Orchestra scheme, whose throughput level gets saturated despite changing sending rates due to its non-adaptive scheduling. As such, it cannot support high-throughput BSN. Recall that the static scheme is a greedy method and uses up all timeslots, so it allocates way more timeslots than necessary in the low traffic load (e.g., the normal, urgent-medium behaviors). Therefore, in contrast to the Orchestra scheme, the static scheme is expected to achieve the highest level of throughput (and PDR as well), but at the cost of higher energy consumption. This explains why the static scheme outperforms the Orchestra scheme in terms of throughput as shown in Figs. 4a, 4c (and PDR as well, as shown later on). Nevertheless, in *normal, urgent-medium*, and *urgent-high* behaviors, our proposed scheme can achieve throughput as high as, or even better than the static scheme. For instance, in Fig. 4c for ECG sensor under urgent-high behavior, the proposed scheme's throughput (15.7 kbps) is 5.7% higher than that of the static scheme (14.8 kbps); and in terms of total throughput, the proposed scheme (31.7 kbps) is also ahead of the static scheme (30.7 kbps). Moreover, our scheme can deliver it with better energy efficiency, as shown later on. Next, under the *request overload regime*, there is an observable difference between proposed and static schemes in the way they allocate throughput to different sensors, i.e., the proposed scheme allocates lower throughput for accelerometer and temperature sensors with sending rates 32 packets/s than the static scheme, but higher throughput for ECG sensor with a higher sending rate 64 packets/s. This is partly explained by the fact that the proposed scheme tries to satisfy sending rate requirements and fairness, which will be reflected by a higher fairness index as presented later on. Furthermore, the proposed scheme's total throughput in this regime is approximately 15% more than the static scheme's.

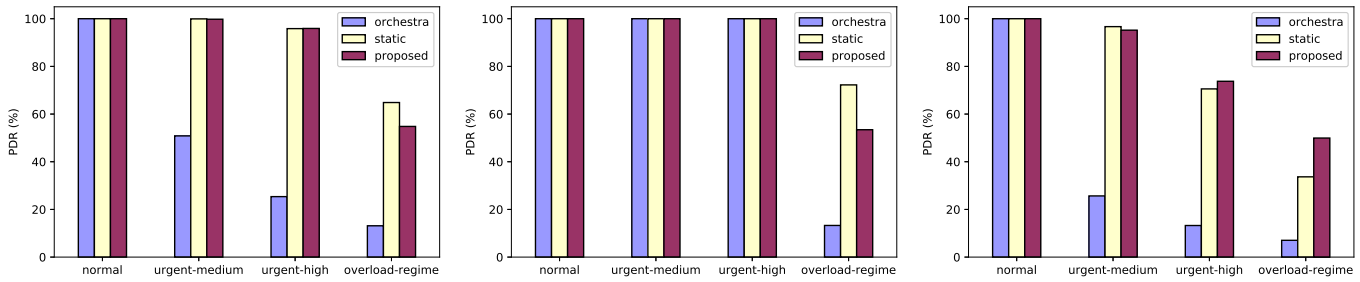
Fig. 5 shows **PDRs** of the three evaluated schemes under different user behaviors. As opposed to throughput, we observe a decreasing trend in PDR across different behaviors corresponding to increasing sending rates, although the trend is not the same for different sensors and schemes. For Orchestra scheme without high data rate support, its PDR is significantly worsened especially for sensors with high sending rates (i.e., for accelerometer and ECG, its PDR drops from 100% in normal behavior to below 15% in urgent-high behavior). Temperature sensor with much lower sending rate, on the other hand, can maintain good PDR (nearly 100%) for all evaluated schemes. Between the static and proposed schemes, measurements suggest no significant difference in the obtained PDR. Specifically, for accelerometer sensor, the static scheme's PDRs are 100%($\pm$0%), 99.92%($\pm$0.33%), and

[6]It is worth to mention that which machine learning model and which feature selection we use are not the novel parts of this paper.



(a) Accelerometer sensor (b) Temperature sensor (c) ECG sensor

Fig. 4. Throughput of (a) accelerometer, (b) temperature, and (c) ECG sensors of the evaluated schemes under different user behaviors.



(a) Accelerometer sensor (b) Temperature sensor (c) ECG sensor

Fig. 5. PDR of (a) accelerometer, (b) temperature, and (c) ECG sensors of the evaluated schemes under different user behaviors.

95.84%($\pm$4.5%)[7] under normal, urgent-medium, and urgent-high behaviors, respectively. For the proposed schemes, the corresponding numbers are 100%($\pm$0%), 99.78%($\pm$0.55%), and 95.92%($\pm$2.88%), respectively. Similar numbers are obtained for ECG sensor, which shows no major differences in PDR. Nevertheless, to the proposed scheme's advantage, it allows for adding more sensors after deployment without causing transmission collisions and PDR degradation, while the static scheme is unable to support this because of its fixed timeslot allocation. Lastly, under the *request overload regime*, the proposed scheme is seen to achieve approximately the same PDRs for all three sensors as a result of fairness allocation in this regime.

Fig. 6 demonstrates that the **EPBs** of the three sensors decrease when increasing their sending rates. For the *normal, urgent-medium*, and *urgent-high* behaviors, noticeably, the proposed scheme always spends much less energy to send a bit compared to the static scheme, which results in longer battery life if we account for a longer period of usage. Moreover, the proposed scheme can achieve EPBs comparable to Orchestra's, at the same time outperforming Orchestra in throughput and PDR. For instance, the EPBs of accelerometer sensor for the proposed scheme under normal, urgent-medium, and urgent-high behaviors are 0.52 mJ/b, 0.42 mJ/b, and 0.36 mJ/b, respectively, while those of Orchestra are 0.52 mJ/b, 0.36 mJ/b, and 0.29 mJ/b, respectively. Under the *request overload regime*, the proposed scheme also achieves better energy efficiency than the static scheme for all three sensors. That is, the EPBs of accelerometer, temperature, and ECG sensors for the proposed scheme are approximately 21%, 12%, and 35% less than those of the static scheme, respectively.

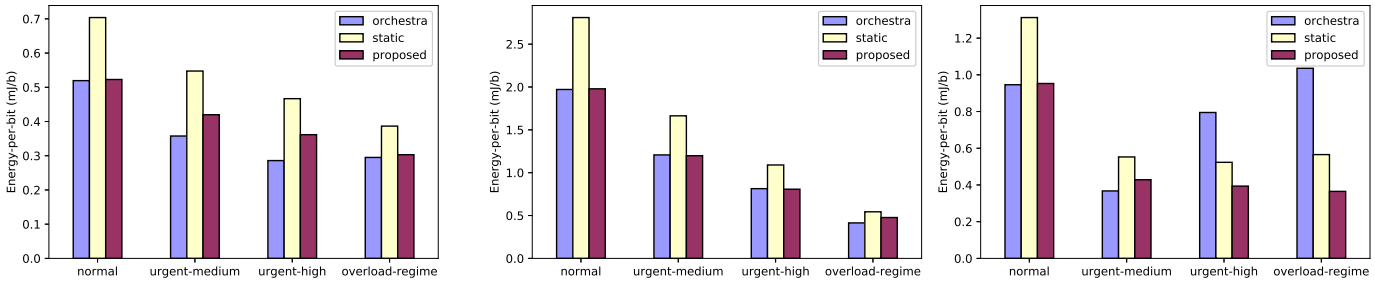
Finally, in the *request overload regime*, since all the requesting sensors cannot satisfy their required sending rates, the **fairness index** provides a better performance indicator as discussed in Section IV-C. According to eq. (4), the required number of extra timeslots for the accelerometer, temperature, and ECG sensors are 6, 6 and 12 timeslots, respectively. However, the total of these numbers exceeds the available extra timeslots given by $N_{\text{free}} = 23 - 4 = 19$ timeslots. Therefore, by applying the fairness allocation mechanism, we obtain $N_{\text{Accelerometer,add}} = 4$, $N_{\text{Temperature,add}} = 4$, and $N_{\text{ECG,add}} = 11$ timeslots. Subsequently, Table II shows the actual throughputs and throughput ratios for the static and proposed schemes under the request overload regime. According to eq. (6), we can calculate the fairness index of the proposed scheme as $F_{\text{proposed}} = 0.997$, which is significantly higher than that of the static scheme $F_{\text{static}} = 0.913$. This result verifies that the proposed scheme achieves close-to-optimal fairness across different sensors under the request overload regime.

In summary, the proposed scheme with adaptive MAC scheduling scheme driven by user behaviors can adapt to the dynamic traffic loads, provides an energy efficiency and reliable communication (i.e., high throughput, and PDR) solution for healthcare applications under the evaluated behaviors, and achieves fairness among the emergency sensors (regarding throughput and PDR) under the request overload regime.

VI. Conclusion

This paper proposed an adaptive user-behavior-driven cross-layer MAC scheduling scheme in order to address the joint

[7]These are mean values with standard deviations in brackets.



(a) Accelerometer sensor (b) Temperature sensor (c) ECG sensor

Fig. 6. Energy-per-bit of (a) accelerometer, (b) temperature, and (c) ECG sensors of the evaluated schemes under different user behaviors.

TABLE II
THROUGHPUT (BPS) AND THROUGHPUT RATIO (r) OF THE THREE SENSORS UNDER THE REQUEST OVERLOAD REGIME

	Accelerometer	**Temperature**	**ECG**
Ideal	29440.00	16128.00	42496.00
Static	24432.05	11573.40	15780.49
Proposed	18271.92	10240.26	23730.14
r_{static}	0.623146	0.702398	0.311892
r_{proposed}	0.536906	0.514845	0.468485

challenge of transmission reliability and energy efficiency in BSN, and built a comprehensive health-monitoring BSN testbed to evaluate its performance. We determined the TSCH slotframe length, the numbers of extra timeslots for applications with dynamic sampling and sending rates, then proposed the equally spaced timeslot allocation algorithm as part of our adaptive MAC scheduling scheme. Our experimental results showed that the proposed scheme can jointly achieve reliable transmission and low energy-efficiency in compared to the Orchestra and static schemes for the BSN with dynamic traffic loads. When there is an overload of requests, the proposed scheme obtains better fairness among the requesting sensors, and better PDR and EPB than other schemes. In the future, we would consider scheduling for multiple BSNs co-located in the vicinity of each other, which introduces significant co-channel interference. In such a scenario, frequency diversity could be leveraged by taking into account both timeslot indices and channel offset to provide reliable communication.

REFERENCES

[1] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, "Enabling effective programming and flexible management of efficient body sensor network applications," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 1, pp. 115–133, Jan. 2013.

[2] M. V. Ngo, Q. D. La, D. Leong, and T. Q. S. Quek, "User behavior driven MAC scheduling for body sensor networks," in *Proc. IEEE HealthCom*, Dalian, China, Oct. 2017.

[3] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M. Leung, "Body area networks: A survey," *Mobile Netw. App.*, vol. 16, no. 2, pp. 171–193, 2011.

[4] R. Cavallari, F. Martelli, R. Rosini, C. Buratti, and R. Verdone, "A survey on wireless body area networks: Technologies and design challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1635–1657, Third Quarter 2014.

[5] B. Shrestha, E. Hossain, and S. Camorlinga, "IEEE 802.15.4 MAC with GTS transmission for heterogeneous devices with application to wheelchair body-area sensor networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 5, pp. 767–777, Sep. 2011.

[6] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015*, vol. 1, pp. 57–150, 2015.

[7] Y. Huang, A. Pang, and H. Hung, "An adaptive GTS allocation scheme for IEEE 802.15.4," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 641–651, May 2008.

[8] Q. D. La, D. Nguyen-Nam, M. V. Ngo, and T. Q. S. Quek, "Coexistence evaluation of densly deployed BLE-based body area networks," in *Proc. IEEE Globecom*, Singapore, Dec. 2017.

[9] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, and T. Engel, "On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks," *IEEE Sensors J.*, vol. 16, no. 2, pp. 550–560, Jan. 2016.

[10] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial iot applications using IEEE802.15.4e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.

[11] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, "Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things," in *Proc. IEEE WoWMoM*, Jun. 2013, pp. 1–6.

[12] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "AdaSense: Adapting sampling rates for activity recognition in body sensor networks," in *Proc. RTAS*, Apr. 2013, pp. 163–172.

[13] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, Third Quarter 2013.

[14] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors J.*, vol. 17, no. 2, pp. 386–403, Jan. 2017.

[15] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Communications and Networks*, vol. 5, no. 1, pp. 3 – 9, 2019.

[16] S. Galzarano, A. Liotta, and G. Fortino, "QL-MAC: A q-learning based MAC for wireless sensor networks," in *Algorithms and Architectures for Parallel Processing*, ser. Lecture Notes in Computer Science. Springer International Publishing, pp. 267–275.

[17] S. Galzarano, G. Fortino, and A. Liotta, "A learning-based MAC for energy efficient wireless sensor networks," in *Internet and Distributed Computing Systems*, ser. Lecture Notes in Computer Science. Springer International Publishing, pp. 396–406.

[18] A. Morell, X. Vilajosana, J. L. Vicario, and T. Watteyne, "Label switching over IEEE 802.15.4e networks," *Emerging Telecommunications Technologies*, vol. 24, no. 5, pp. 458–475, Aug. 2013.

[19] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. ACM SenSys*, 2015, pp. 337–350.

[20] P. Du and G. Roussos, "Adaptive time slotted channel hopping for wireless sensor networks," in *Proc. 2012 4th Computer Science and Electronic Engineering Conference (CEEC)*, Sep. 2012, pp. 29–34.

[21] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct 2013.

[22] K.-H. Choi and S.-H. Chung, *A New Centralized Link Scheduling for 6TiSCH Wireless Industrial Networks*. Cham: Springer International Publishing, 2016, pp. 360–371.
[23] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," in *Proc. WCNC*, Apr. 2016, pp. 1–6.
[24] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proc. IEEE Local Comput. Netw.*, Nov. 2004, pp. 455–462.
[25] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research, Tech. Rep., 1984.
[26] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," Swedish Institute of Computer Science, Tech. Rep., Mar. 2011.
[27] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

Mao V. Ngo (S'17) received the B.Eng. degrees (with first-class honours) in telecommunications engineering from the National Technical University of Ukraine, Kyiv Polytechnic Institute (KPI), in 2011, and the M.Eng. degree in electronics engineering from Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 2015. He is currently pursuing the Ph.D. degree with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design (SUTD) under the Singapore International Graduate Award (SINGA) A*STAR scholarship. His current research interests include edge computing, machine learning, wireless sensor networks, and scheduling for internet-of-things networks.

Quang Duy La (S'09-M'14) received the B.Eng. degree (with first-class honours) in 2008, and the Ph.D. degree under the Nanyang Presidents Research Scholarship in 2013, both in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore. He was a Postdoctoral Research Fellow at Temasek Laboratories, Singapore University of Technology and Design (SUTD) from 2015 to 2018. He has participated in research projects on radio resource allocation, smart grid, body area networks and network security. His research interests lie in game-theoretic and distributed algorithms for next generation communications systems, network security, mobile edge computing, as well as machine learning and big data analytics for the Internet of Things.

Dr. La is a co-author of the monograph "Potential Game Theory: Applications in Radio Resource Allocation" published by Springer in 2016.

Derek Leong received his B.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2005, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2008 and 2013, respectively. From 2012 to 2018, he was a scientist with the Institute for Infocomm Research (I^2R), A*STAR, Singapore. His research and development interests include distributed systems and networking.

Tony Q.S. Quek (S'98-M'08-SM'12-F'18) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. Currently, he is a tenured Associate Professor with the Singapore University of Technology and Design (SUTD). He also serves as the Acting Head of ISTD Pillar, Sector Lead of the SUTD AI Program, and the Deputy Director of the SUTD-ZJU IDEA. His current research topics include wireless communications and networking, network intelligence, internet-of-things, URLLC, and big data processing.

Dr. Quek has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is currently serving as the Chair of IEEE VTS Technical Committee on Deep Learning for Wireless Communications as well as an elected member of the IEEE Signal Processing Society SPCOM Technical Committee. He was an Executive Editorial Committee Member for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, and an Editor for the IEEE WIRELESS COMMUNICATIONS LETTERS.

Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards – Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, and the 2016-2018 Clarivate Analytics Highly Cited Researcher. He is a Distinguished Lecturer of the IEEE Communications Society and a Fellow of IEEE.

Hyundong Shin (S'01-M'04-SM'11) received the B.S. degree in electronics engineering from Kyung Hee University (KHU), Yongin-si, Korea, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2001 and 2004, respectively. During his postdoctoral research at the Massachusetts Institute of Technology (MIT) from 2004 to 2006, he was with the Wireless Communication and Network Sciences Laboratory within the Laboratory for Information Decision Systems (LIDS).

In 2006, Dr. Shin joined the KHU, where he is now a Professor at the Department of Electronic Engineering. His research interests include quantum information science, wireless communication, and nanonetworks.

Dr. Shin was honored with the Knowledge Creation Award in the field of Computer Science from Korean Ministry of Education, Science and Technology (2010). He received the IEEE Communications Societys Guglielmo Marconi Prize Paper Award (2008) and William R. Bennett Prize Paper Award (2012). He served as a Publicity co-chair for the IEEE PIMRC (2018) and a Technical Program co-chair for the IEEE WCNC (PHY Track 2009) and the IEEE Globecom (Communication Theory Symposium 2012, Cognitive Radio and Networks Symposium 2016). He was an Editor for IEEE Transactions on Wireless Communications (2007-2012) and IEEE Communications Letters (2013-2015).