Optimized Congestion Management Protocol for Healthcare Wireless Sensor Networks

Abbas Ali Rezaee · Mohammad Hossein Yaghmaee · Amir Masoud Rahmani

Published online: 1 August 2013 © Springer Science+Business Media New York 2013

Abstract The application of Wireless Sensor Networks (WSNs) in healthcare is dominant and fast growing. In healthcare WSN applications (HWSNs) such as medical emergencies, the network may encounter an unpredictable load which leads to congestion. Congestion problem which is common in any data network including WSN, leads to packet loss, increasing endto-end delay and excessive energy consumption due to retransmission. In modern wireless biomedical sensor networks, increasing these two parameters for the packets that carry EKG signals may even result in the death of the patient. Furthermore, when congestion occurs, because of the packet loss, packet retransmission increases accordingly. The retransmission directly affects the lifetime of the nodes. In this paper, an Optimized Congestion management protocol is proposed for HWSNs when the patients are stationary. This protocol consists of two stages. In the first stage, a novel Active Queue Management (AQM) scheme is proposed to avoid congestion and provide quality of service (QoS). This scheme uses separate virtual queues on a single physical queue to store the input packets from each child node based on importance and priority of the source's traffic. If the incoming packet is accepted, in the second stage, three mechanisms are used to control congestion. The proposed protocol detects congestion by a three-state machine and virtual queue status; it adjusts the child's sending rate by an optimization function. We compare our proposed protocol with CCF, PCCP and backpressure algorithms using the OPNET simulator. Simulation results show that the proposed protocol is more efficient than CCF, PCCP and backpressure algorithms in terms of packet loss, energy efficiency, end-to-end delay and fairness.

A. A. Rezaee (🖂) · A. M. Rahmani

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran e-mail: aa.rezaee@srbiau.ac.ir

A. M. Rahmani e-mail: Rahmani@srbiau.ac.ir

M. H. Yaghmaee Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran e-mail: hyaghmae@ferdowsi.um.ac.ir **Keywords** Wireless sensor networks · Congestion control · Healthcare application · Optimization · Quality of service (QoS)

1 Introduction

Application of Wireless Sensor Networks (WSNs) is dominant and growing fast [1]. Designing intelligent sensors with low power, low weight, and wireless connections make it possible to use sensors in healthcare. Sensors, attached to a patient's body, can sense, process, and generate data (such as the pulse rate of a critically sick patient), which are eventually delivered to the control center for analysis by medical personnel, but we face some challenges such as congestion control and energy consuming which are the typical issues in a HWSNs [2,3].

WSN may be used in a variety of healthcare applications. For example, in hospitals, WSN can be applied in monitoring of the heart disease patient's ECG. It is possible to use WSN for early detection of problems like heart attacks. In homes, asthma and stress monitoring, and also monitoring of old or disabled people is possible using WSN [2,4–7]. WSN may also be used in medical emergencies. In majority of emergency cases such as diabetes, the reaction time is a critical issue. The long reaction time in these cases may cause irreparable damages or even may leads to death of the patient. Monitoring via sensors will allow remote (non-hospital) monitoring of the patient. It also increases efficiency of primary medical care in hospital by reducing caregiver's reaction time. Here, the sensors may monitor a patient's physical situation and send vital signs to the hospital or doctors immediately. Patients can be prioritized based on the severity of their condition. Accurate and quick delivery of patient vital signs is crucial for making efficient and real time decisions [8].Applications of sensor networks should have an adequate quality of services to guard against delay, and packet loss; and also being able to enhance the lifetime of the networks.

In many HWSNs (for example when sensors are installed inside the patient's body) [4,9], the nodes are mandated to work continuously and uninterruptedly for months, and sometimes years without having the possibility of replacing/recharging the energy source. Therefore, optimizing the energy consumption of the sensor node is always a critical issue. In medical emergencies, in a short period of time, different type of sensors may sense a large number of events and produce large amounts of data and try to transmit their data to the sink (Medical Center) with the help of intermediate nodes. This leads to a burst data on the network which in turn may cause congestion at the intermediate nodes. Congestion has a direct impact on increasing the end-to-end delay and the packet loss. In modern wireless biomedical sensor networks, increasing these two parameters for the packets that carry EKG signals may even result in the death of the patient [10]. Furthermore, when congestion occurs, because of the packet loss, packet retransmission increases accordingly. The retransmission directly affects the lifetime of the nodes with limited battery, which in turn reduces the network lifetime. Due to mentioned issues, it can be concluded that congestion in HWSNs, is more challenging compared to other applications.

In WSNs, two types of congestion may occur [11,12]: node-level congestion and linklevel congestion. (See Fig. 1). The first type of congestion is called node-level congestion (Fig. 1a) and that is common in conventional networks. It happens when the packet arrival rate is higher than the packet service rate and it causes buffer overflow in the node. This type of congestion can result in packet loss and increased queuing delay. Packet loss will lead to retransmission and will consume additional energy. The second type of congestion is link-level congestion (Fig. 1b). Wireless channels are shared by several nodes using CSMA protocols (Carrier Sense Multiple Access). Collisions may occur when multiple active sensor



Fig. 1 Congestion conditions in wireless sensor networks. a Node level congestion buffer overflow, b linklevel congestion link collision

nodes try to seize the channel at the same time. Link level congestion increases packet service time and decreases link utilization, (which leads to decreasing the overall throughput), and waste energy at the sensor nodes. Both types of congestion have direct impact on energy efficiency and QoS.

Recently, congestion control in WSNs, received a considerable attention [13,14]. It is composed of three mechanisms: congestion detection, congestion notification, and rate adjustment. Depending on the congestion control protocol, the following objectives may be achieved and efficiency of the protocol may be improved. (1) The energy-efficiency requires to be improved in order to extend the system's lifetime. Therefore, in congestion control protocols, packet loss must be avoided or reduced to a least possible value in order to save energy (2) Support traditional QoS metrics such as packet loss, packet delay, and throughput. For example, in multimedia applications, end-to-end delay also needed to be guaranteed [15]. (3) fairness should be guaranteed so that each node can achieve fair bandwidth throughput.

The existing congestion control protocols for WSNs cannot be directly applied in HWSNs. The reason is that: firstly, these protocols do not consider the special QoS requirements of the data carried in these environments. Secondly, these protocols treat the congestion problem after it occurs. In HWSN application, it is also necessary to avoid the congestion in advance before it occurs and then face the congestion itself.

In this paper an Optimized Congestion management protocol is proposed. We propose an Active Queue Management scheme AQM) to avoid congestion according to a packet drop probability. We also proposed three state automata to calculate the congestion level in each node and an optimization function for calculating the new sending rate for child nodes.

The rest of this paper is organized as follows: In Sect. 2, a brief review of prior related works on congestion control is presented. Then the details of the proposed model are explained in Sect. 3. In Sect. 4 performance of the proposed model is evaluated by using computer simulation. And Sect. 5 concludes the paper.

2 Related Works

In recent years congestion control protocols have been studied extensively in WSNs [11, 12, 15–24]. Congestion is an important problem in wireless sensor networks. The reason is that it leads to packet loss and retransmissions which wastes the sensors limited battery energy.

Congestion Detection and Avoidance (CODA) [16], uses open-loop and closed-loop rate adjustment. In open-loop hop-by-hop backpressure, when a node experiences congestion, it broadcasts back-pressure messages upstream towards the source nodes, to reduce their sending rate. In closed-loop multi-source regulation, the sink control congestion of multiple sources. It detects congestion based on buffer occupancy as well as the wireless channel load. The rate adjustment algorithm works like an additive increase multiplicative decrease (AIMD).

Congestion Control and Fairness (CCF) [12] is a protocol that uses packet service time for detecting congestion in each intermediate sensor node and for deducing the available service rate. This protocol controls congestion in a hop-by-hop manner, with exact rate adjustments based upon its available service rate and the number of each node's childs. CCF guarantees simple fairness by the same throughput for each node. It also adjusts the rates that rely only on packet service time, which could have low utilization due to sensor nodes not generating enough traffic or if there is a significant Packet Error Rate (PER).

In Priority for Congestion Control Protocol (PCCP) [11], a hop-by-hop upstream congestion control protocol for WSNs is proposed. The congestion degree in PCCP is measured as the ratio of packet inter-arrival time to the packet service time. This protocol utilizes a cross-layer optimization and imposes a hop-by-hop approach to control congestion based on the introduced congestion degree and the node priority index. It has also been demonstrated that PCCP achieves energy efficiency and flexible weighted fairness for both single-path and multipath routing.

Learning Automata-Based Congestion Avoidance Scheme (LACAS) [18] is appropriate for HWSN application usage. The advantage of LACAS is that it is intelligent, learns from the past, and improves its performance in time. Optimal action is selected at the moment, based on the knowledge of previous levels of congestion, by the automata stationed in the different sensor nodes. The results show that the proposed algorithm is avoiding congestion successfully in typical HWSNs, which is needed for a reliable congestion control mechanism.

In Priority-based rate control [15], a model is presented which is used for congestion control and service differentiation in WMSNs. The proposed congestion control protocol adjusts the traffic source rates based on current congestion in the upstream nodes and the priority of each traffic source. Results show that it can achieve low packet loss probability; it is possible to provide low queuing delay, and it guarantees bandwidth for high priority real time traffics.

Fairness-Aware Congestion Control protocol (FACC) [17] has provided a new mechanism to control congestion and achieve reasonably fair bandwidth allocation in resourceconstrained WSNs. In this protocol, intermediate sensor nodes are categorized into nearsource nodes and near-sink nodes. A per-flow state is maintained by the near-source nodes and an approximately fair rate is allocated to each passing flow. Near-sink nodes do not need to maintain a per-flow state. The nodes use a lightweight probabilistic dropping algorithm based on queue occupancy and hit frequency. FACC provides better performance than previous approaches in terms of throughput, packet loss, energy efficiency, and fairness.

Enhanced Congestion Detection and Avoidance (ECODA) [19] proposes a novel energy efficient congestion control scheme for sensor networks. It uses a flexible queue scheduler and packets are scheduled according to the priority. Transient congestion and persistent congestion efficiently are considered by ECODA. It adopts hop-by-hop congestion control scheme for transient congestion and adjusts the bottleneck node, based on source data sending rate control for persistent congestion. ECODA achieves efficient congestion control and flexible weighted fairness for different class of traffics. It leads to lower packet loss, higher energy efficiency, and better QoS in terms of throughput, fairness, and delay.

Fuzzy congestion controller for wireless sensor networks (FCC) [20] develops a fuzzy rule base as well as fellowship functions. It uses channel load and queue size of intermediate nodes as the indication of congestion to organize the inputs. The output is branched from the fuzzy rule base and the fuzzy reference engine conjuncts and determines new source rates. This algorithm reduces packet loss comparing with non-fuzzy methods. It increases throughput and energy consumption.

A Prioritization Based Congestion Control Protocol for Healthcare Monitoring Application in Wireless Sensor Networks is proposed in [23] provides a congestion control and service prioritization protocol for real time monitoring of patients' vital signs using wireless biomedical sensor networks. This protocol assigns different priorities for physiological signals and provides a better quality of service for transmitting highly important vital signs. Congestion control is performed by considering both the congestion situation in the parent node and the priority of the child nodes in assigning network bandwidth to signals from different patients. In [25] a class based congestion control protocol with service differentiation is proposed to reduce queuing delay and provide reliable data transmission for them. It distinguishes between different kinds of traffic like delay sensitive, drop sensitive and control traffic. Delay sensitive traffic comes from emergency conditions, where the emergency data should be guaranteed to be delivered with a reasonable delay. Drop sensitive traffic comes from patients' normal conditions where the normal data can tolerate some delay but they need low packet drop. Control traffic are generate for network maintenance and doctors periodical or on demand report.

The [26] designs a Queue management and energy efficient congestion control protocol in the wireless body sensor network. The proposed protocol focuses on efficient management of queue to provide reliability and reduce packet loss. In [24] a distributed congestion control algorithm is proposed for tree based communications in wireless sensor networks. This protocol allocates a fair and efficient transmission rate to each node. The congestion control algorithm apportions the total change in a control interval in aggregate traffic into individual flows to achieve fairness.

3 Proposed Congestion Management Protocol

In this section we describe our proposed congestion management protocol for HWSNs. This protocol is motivated by the limitations of existing popular congestion control protocols such as PCCP and CCF. These protocols do not entertain congestion avoidance mechanisms. Furthermore, The CCF protocol cannot effectively allocate the remaining capacity and use work-conservation scheduling algorithm which leads to a lower throughput. The PCCP protocol performs poorly in providing relative priority in case of different congestion situations. In case of low congestion, PCCP increases the scheduling and data rate of all traffic sources without paying attention to their priority index. In case of high congestion, PCCP decreases the sending rate of all traffic sources based on their priority index. Another problem of PCCP is that, it considers only geographical priority and this protocol does not support multi-sensor with dynamic priority situations (patients overall health status) which is necessary in HWSNs. Another issue is that these two protocols do not consider the importance of a node (e.g. The heart rate of a patient with heart diseases compared to a heart rate of a patient with a broken leg).

The proposed protocol solves the problem by a proper adjustment of the send rate of each child node using an optimization function. The proposed optimization function adjusts the sending rate of each child node based on its output available bandwidth and dynamic priority level of each child node.



Fig. 2 Architecture of the proposed congestion management protocol

The proposed protocol architecture is shown in Fig. 2. It consists of two stages. The first stage tries to avoid congestion by using an AQM. It decides either to accept or to drop an incoming packet, according to a packet drop probability. If the incoming packet is accepted, then in the second stage, three mechanisms are used to control possible congestion, namely, automata based Congestion Detection (ACD), implicit Congestion Notification (ICN), and optimized Rate Adjustment (ORA).

ACD detects congestion by a three-state machine and virtual queue status. The output of ACD is the congestion level ($0 \le CL \le 1$).

If *CL* is more than a specific threshold, optimized rate adjustment (ORA) procedure will be called. This procedure will resolve an optimization problem and will obtain new share rates for child nodes. Since in the applications like healthcare, each node (patient) can be installed with multiple sensors having different priorities, we use priority as an important parameter in our model. We use child priority (*CHP*) and Available Bandwidth (*AB*) as the input parameters for the optimization function (AB = 1 - CL). ORA procedure calculates the new share rates for each child node as well as local source traffic.

The proposed protocol uses the implicit Congestion Notification (ICN) which piggybacks congestion information in the header part of data packets and avoids sending additional control messages which consequently, causes an improvement in energy-efficiency.

Before describing the proposed model in detail, the following provide some definitions related to the priority:

Definition 1 Static Priority at node i (STP^{i}) is defined as:

$$STP^{i} = PaI^{i} \cdot \sum_{k} SP_{k}^{i}$$
⁽¹⁾

where, Patient's importance $(0 \le PaI^i \le 1)$ at node i is the importance degree of the patient. For example a heart disease patient and a diabetic patient have different importance. The parameter $0 \le SP_k^i \le 1$ denotes the *k*th source data priority for a sensor type in node *i*. Its value can be set manually for service differentiation. *k* is the sensor type on a patient's body.

Definition 2 Global Priority at node $i(GP^i)$ is defined as:

$$GP^i = TP^i + STP^i, (2)$$

where, TP^i is Transit data Priority at node *i*. It shows the priority of transit data traffic forwarded through node *i*.

The Global Priority Ratio GPR_j^i for each virtual queue *j* in node *i* is defined as follows. The structure of the node's queue is defined in the next section in detail.

$$GPR_1^i = \frac{STP^i}{GP^i}, \quad GPR_j^i = \frac{GP^j}{GP^i}, \quad j = 2, 3, \dots, N_i + 1$$
 (3)

where N_i is the number of child nodes of the node *i*.

Definition 3 Child Priority *j* in node $i(CP_i^i)$ is defined as:

$$CP_{j}^{i} = \frac{GPR_{j}^{i} \cdot (1 - CL_{j}^{i}) \cdot \overline{TW}_{j}^{i} \cdot \overline{delay}_{j}^{i}}{\sum_{j=1}^{N_{i}+1} GPR_{j}^{i} \cdot (1 - CL_{j}^{i}) \cdot \overline{TW}_{j}^{i} \cdot \overline{delay}_{j}^{i}}$$
(4)

where average traffic weight (\overline{TW}_{j}^{i}) and average delay $(\overline{delay}_{j}^{i})$ of each virtual queue *j* in node *i* are obtained as follows:

$$\overline{TW}_{j}^{i} = \frac{\sum_{k} m_{k}^{j}(i).SP_{k}}{\sum_{k} SP_{k}}, \quad \overline{delay}_{j}^{i} = \frac{\sum_{k} \overline{delay}_{k}^{j}(i).SP_{k}}{\sum_{k} SP_{k}}$$
(5)

Here, $m_k^j(i)$ is the number of packets with source priority k in virtual queue $j \cdot delay$ is the time elapsed from packet generation up to current time. $\overline{delay}_k^j(i)$ is the average packet delay of type k in virtual queue j at node i. $0 \leq CP_j^i \leq 1$ is the normalized child priority. Note that *CP* (child priority) for each node i is independent of the other nodes.

3.1 Congestion Avoidance Algorithm

Active Queue Management (AQM) schemes are mechanisms to avoid congestion and provide quality of service (QoS) by pro-actively dropping packets [27–31]. With these schemes, congestion can be controlled, and network performance such as network delay, link utilization, packet loss rate, and system fairness are improved. The proposed algorithm uses a flexible procedure for queue management. This procedure uses separate virtual queues to store the input packets from each child node. It means that a physical queue (QL) is shared virtually between N child node's traffic and one local source's traffic based on their importance and priority (Fig. 3). If a child node has more priority, it can use more space for its queue. However, the boundaries between these virtual queues are not constant; when one of the child nodes has a free space in its own virtual queue, others can use the free space if necessary.

When a packet is received from a child node *j* by a parent node, packet drop probability $Pdrop_j$ is computed for the packet. At first the initial value of packet drop probability for $j Pdrop_j^{pri}$, is calculated as follows:

$$Pdrop_{j}^{pri} = \begin{cases} q_{j} < \alpha_{1} \cdot QL \cdot CP_{j} & 0\\ \alpha_{1} \cdot QL \cdot CP_{j} < q_{j} < \alpha_{2} \cdot QL \cdot CP_{j} & \frac{q_{j} - \alpha_{1} \cdot QL \cdot CP_{j}}{\alpha_{2} \cdot QL \cdot CP_{j} - \alpha_{1} \cdot QL \cdot CP_{j}} & (6)\\ q_{j} > \alpha_{2} \cdot QL \cdot CP_{j} & 1 \end{cases}$$

Here q_j is the queue size of *j*th virtual queue (which belongs to the *j*th child), and *QL* is the maximum size of the physical queue. α_1 and α_2 are lower and upper thresholds respectively that will be adjusted through simulations. CP_j is the normalized child node priority.



Fig. 3 A physical queue is divided into N + 1 virtual queue

Note that $QL \cdot CP_j$ represents the virtual queue share for child *j* in the physical queue QL. After the initial value of packet drop probability is computed, AQM mechanism calculates packet drop P_j as follows:

$$Pdrop_{j} = \beta_{1}.\delta qv_{i} - \beta_{2}.\left(1 - \left(\sum_{m=1}^{k} q_{m}/QL\right)\right) + Pdrop_{j}^{pri}$$
(7)

The primary value for the drop probability $Pdrop_j^{pri}$, is adjusted based on the network condition. The second expression (7) specifies the total free space in the physical queue. Multiplying this value by β_2 (it is a flexible parameter), will cause $Pdrop_j$ to be reduced. In the other words, the more the free space is, the less packet drop probability will be achieved. However, the effect of this value depends on the β_2 parameter.

 $\delta q v_i$ is the level of variation in the length of the *j*th virtual queue and is calculated as:

$$\delta q v_j = \frac{q_j^{new} - q_j^{old}}{QL \cdot CP_j} \tag{8}$$

The value of $\delta q v_j$ can be positive or negative. $\delta q v_j$ is multiplied by coefficient β_1 , as presented in Eq. (7). Note that the parameters in the Eqs. (6)–(8) are determined periodically. Therefore, the value of q_j^{old} is the *j*th virtual queue size at the start of the last period, and the value of q_j^{new} being the *j*th virtual queue size at the end of the last period and the beginning of the new period. If the differential of *j*th virtual queue size is positive, $\delta q v_j$ will be positive, and will remain positive after multiplying by β_1 causing P_j to increase. It means that if the variation in the virtual queue length is positive (the queue size grows up), then the packet drop probability is increased.

Finally, the packet will be queued or dropped, according to the drop probability P_j value. If the packet doesn't drop and is queued in its virtual queue in the parent node *i*, then our protocol checks congestion status in the parent node and decides whether to control congestion or not.



Fig. 4 The proposed automaton states

3.2 Congestion Control Algorithm

(1) Three state Automata: We use automata to determine congestion levels in each node. We assume all the intermediate nodes have automata stationed. The concept underlying the theory of automata is that automata is a self-operating machine, or a mechanism that responses to a sequence of instructions in a certain way to reach a certain goal. We map our proposed three state automata to the 5-tuple {Q, A, B, F(.,.), G(.)}, in [32]:

Q = {*Normal, Congestion Avoidance, Congested*} is the set of finite states.

 $A = \{0 < congestion \ level < 1\}$ is the output.

 $B = \{q_1, q_2, ..., q_k\}$ are the inputs to the automaton, which represent a queue portion for each child in the physical queue at any time t.

 $F(.,.): Q \times B \rightarrow Q$ is the transition function [i.e. a function that determines the state of the automaton at any subsequent time instant (t + 1)] mapping in terms of the state and input at the instant t, such that, q (t + 1) = F(Q, B).

G(.): $G : Q \rightarrow A$ is the output function mapping, that determines the output of the automaton (*congestion level*).

Our proposed automaton is presented in the Fig. 4. Finite states are illustrated by circles and transitions represented by arrows. When the automaton meets a symbol of the condition according to Eq. (9), it makes a transition (or jump) to another state based on its transition function (which takes the current state and each child virtual queue portion as its inputs). When the packet is accepted and queued in a virtual queue in the parent node, three finite states of machine calculate *congestion level* for the child node traffic.

The automaton transitions are as follows:

$$I: \text{ if } \sum_{m=1}^{k} q_m > QL \cdot \alpha_1 \text{ or } \exists j | q_j > QL \cdot CP_j$$

$$II: \text{ if } \sum_{m=1}^{k} q_m > QL \cdot \alpha_1 \text{ or } \forall j \in S, L_j > 0 \cdot 95 \cdot QL \cdot CP_j$$

$$III: \text{ if } \sum_{m=1}^{k} q_m < QL \cdot \alpha_2$$

Deringer

$$IV: \text{ if } \sum_{m=1}^{k} q_m < QL \cdot \alpha_1$$
$$0 \le \alpha_1 \le \alpha_2 \le 1$$
(9)

I. At first all nodes are in Normal state. In this state, the arrival rate from the neighborhood of the node, is less than or equal to the service rate of the relative queue. So congestion doesn't occur at any node and the average of virtual queue length is lower than their portions. When the total length of virtual queues becomes more than $QL.\alpha_1$, or at least one of the virtual queues exceeds from its own portion, the neighborhood transmission rate is increased and the node will change from Normal state into Congestion Avoidance state.

II. In Congestion Avoidance state, when the arrival rate increases, the algorithm prevents congestion as much as possible. More rigor is applied for preventing congestion compared with the Normal state. If the total length of virtual queues is more than $QL.\alpha_2$, or for each neighbor part of *s* (*s* is half of neighborhood number k/2), the related size of the virtual queue is more than 95 percent of its portion (depending on the repeated experiments), the node state will change into the Congested state.

III. The node has congestion in the Congested state. If the total length of queues becomes lower than $QL \cdot \alpha_2$, the node comes back to the Congestion Avoidance state. This will help the node to adapt itself if congestion is reduced and it aids in the congestion level measuring.

IV. Also, if in the Congestion Avoidance state, the total length of queues becomes lower than $QL \cdot \alpha_2$, the node comes back to the Normal state.

 α_1 and α_2 is the threshold of transferring from Normal to Congestion Avoidance state and from Congestion Avoidance to Congestion state respectively.

(2) Automata based Congestion Detection (ACD): The node congestion level ncl_i reflects the current congestion level at *i* sensor node. In each *i*th node with *k* child nodes (Fig. 3), the congestion level of *j*th child (cl_j) , is calculated according to its current virtual queue and automaton status. We suppose that each child node which is sending traffic, buffers it in a separate virtual queue. Initially all queues are in the Normal state. So if a child node's traffic is queued in a virtual queue with Normal state, and its size q_i is more than its share, the child node congestion level (cl_j) is calculated as:

$$\forall j, if \ \frac{q_j}{QL.CP_j} \ge 1 \ then \ \ cl_j = \frac{q_j}{QL \cdot CP_j} - 1, \ else \ \ cl_j = 0 \tag{10}$$

where $QL \cdot CP_j$ is the *j*th child node's portion of physical queue QL in node *i*. If sent traffic from a child node is queued in a virtual queue with Congestion Avoidance state, the child node congestion level is obtained as:

$$\forall j, if \begin{cases} \frac{q_j}{QL \cdot CP_j} < 1/3 \rightarrow cl_j = 0\\ 1/3 < \frac{q_j}{QL \cdot CP_j} < 2/3 \rightarrow cl_j = \frac{q_j}{QL \cdot CP_j} \cdot \frac{L_j}{QL \cdot CP_j}\\ \frac{q_j}{QL \cdot CP_j} > 2/3 \rightarrow cl_j = \frac{L_j}{QL \cdot CP_j} \end{cases}$$
(11)

where L_j is the average length of *j*th virtual queue, with input rate μ , and service rate λ in a server system computed as [33]:

$$L_{j} = \frac{\rho}{1-\rho} - \frac{(k+1)\cdot\rho^{k+1}}{1-\rho^{k+1}}, \quad k = QL \cdot CP_{j}$$

$$\rho = \frac{\lambda \cdot (1-P_{k})}{\mu}$$

$$P_{k} = \frac{(1-\rho), \rho^{k}}{1-\rho^{k+1}}$$
(12)

🖉 Springer

K is the total queue size and $P = (\lambda/\mu)$. If the sent traffic from a child node is queued in a virtual queue with Congestion state, the child node congestion level is calculated as:

$$\forall j, if \begin{cases} \frac{q_j}{QL \cdot CP_j} < 1/3 \rightarrow cl_j = \frac{q_j}{QL \cdot CP_j} \cdot \frac{L_j}{QL \cdot CP_j} \\ \frac{q_j}{QL \cdot CP_j} > 1/3 \rightarrow cl_j = \frac{q_j}{QL \cdot CP_j} \end{cases}$$
(13)

 cl_j is utilized for rate adjustment. Using the congestion level equation (13), each *i* sensor node may calculate the node congestion level ncl_i as:

$$ncl_i = \sum_{j=1}^k CP_j.cl_j \tag{14}$$

The *node congestion level* (*ncl*) indicates the available bandwidth in the node (AB = 1-ncl). When the packet arrival rate is larger than packet service rate (*ncl* > 0), the node experiences congestion.

(3) Implicit Congestion Notification (ICN): There are two methods to inform congestion information: Explicit Congestion Notification (ECN) and Implicit Congestion Notification (ICN). The explicit congestion notification uses specific control messages and causes additional overhead. So, we use the advantage of the broadcast nature of wireless channels (we assume that Omnidirectional antenna is used), so that the child nodes can obtain congestion and acknowledgment information from the packet's header which are then forwarded towards the sink by parent nodes. In this way transmission of additional control messages is avoided in the implicit congestion notification.

(4) **Optimized Rate Adjustment (ORA):** Suppose that node *i* has *k* child nodes which they send packets to their parents to forward them toward the sink, with their initial rate. The maximum rate for each node is r_{max} . Congestion will occur in node *i* if the total average rate of child nodes is more than service rate. In this situation, an optimization problem will be used to compute the new sending rate for child nodes (Fig. 4). Following, we propose an optimization problem, to calculate a new sending rate for child nodes.

$$MinF_i = \sum_{j=1}^{N+1} CP_j \cdot \frac{AB_i - S_j}{\varepsilon + S_j}$$
(15)

S.t.

$$\sum_{j=1}^{N+1} S_j \le \alpha \cdot AB_i, \quad S_j \ge 0, 0 \le \alpha \le 1$$
(15-1)

where (15-2)

$$0 \le CP_j \le 1$$
, $AB_i = 1 - ncl_i$

Figure 5 clarifies Eq. (15). Equations (15-1) and (15-2) present the conditions of optimization problem. The aim of optimization is to minimize the cost function of Eq. (15). In this function, N is the number of child nodes of the node i and CP_j is the *j*th child node's normalized priority. This function calculates N new share rates for the child nodes' data traffic and one for the local source node's data traffic. AB_i is the available bandwidth in node i. The overall congestion level at node i is denoted by ncl_i . If the node congestion level increases, its available bandwidth will be decreased. For node i with N child nodes, the total share of all child nodes is obtained as Eq. (15-1), where s_j is the portion of *j*th child node

21



Fig. 5 Rate adjustment model used in intermediate nodes



Fig. 6 A run of optimization function

for sending data determined by node *i*. This share is proportional to the maximum allowed bandwidth of the child nodes. Therefore the new sending rate of each child node is calculated as in Fig. 5 ($r_{new} = r_{max} \times s_i$). In Eq. (15-1), α is a constant which controls the maximum rate of the total share in relation with available bandwidth.

One of the fundamental characteristics of the proposed algorithm is that not only priority of the nodes is considered but also, the network status is taken to account and depending on the transmission portion is adjusted.

In Fig. 6 a run of optimization function for a node with two child node is depicted for $CP_1 = 0.6$, $CP_2 = 0.4$, $\alpha = 0.9$, AB = 0.8.

3.3 Output Scheduler

In the output of each node, there is a scheduler to service the packets in virtual queues, based on the priority of source or child's traffic. We use a weighted fair queuing (WFQ) algorithm which is able to service each virtual queue practically and can guarantee fairness between source and child traffic. The priority of source traffic and child's transit traffic as in relation (4) is used as the weight, for source traffic and child's transit traffic virtual queues.



Fig. 7 The single path topology used in simulation

| Table 1 | Simulation | parameters |
|---------|------------|------------|
|---------|------------|------------|

| Transmission range | 40 m |
|----------------------------|-------------------------------|
| Initial node energy | 50 J |
| Initial service rate | 700 packets/s |
| Number of traffic types | 4 |
| Network area | $200 \times 200 \mathrm{m}^2$ |
| Packet sent energy | 12 MJ |
| Packet receive energy | 10 MJ |
| Congestion detection epoch | Each 50 packet |

| PT | FP | SndAd | Sqn | CT | NH |
|----|----|-------|-----|----|----|
| | | | | | |

Fig. 8 The structure of the data packet header. *PT* Packet type, *FP* flow priority, *SndAd* sender address, *Sqn* sequence number, *CT* created time, *NH* next hop

4 Performance Evaluation of the Proposed Protocol

We evaluate the performance of the proposed protocol using MATLAB and OPNET [34]. We run the optimization function in Eq. (15) in MATLAB and the simulation process was carried out using OPNET. Since both software have been programmed based on C++, we have the option of creating links between them. Therefore, the proposed protocol was simulated by linking OPNET and MATLAB software which use C++ compiler. In this simulation, we run both MATLAB and OPNET software concurrently. The optimization function equation (15) along with the other related functions are implemented in MATLAB software. Then OPNET calls MATLAB functions when is needed. The network topology used in the simulation is a simple single path network as in Fig. 7. Table 1 also presents parameters used in simulations.

The structure of the proposed data packet header is shown in Fig. 8.

The *packet type* distinguishes data packets from control packets (rate adjustment packet). The *flow priority* indicates the source sensor priority. The *Sender address* specifies the child number for subsequent rate share determination. The *Sequence number* field is a positive integer that avoids duplicated packets. The *created time* field is used to calculate data packet delay. The *next hop* determines the next node to which a data packet should be delivered.

The structure of rate adjustment control packet in the proposed protocol is shown in Fig.9.

| PT | AB | S | SRV |
|----|----|---|-----|

Fig. 9 The structure of the rate adjustment control packet. *PT* packet type, *AB* available bandwidth, *S* share rate, *SRV* service rate, *NH* next hop



Fig. 10 System total normalized throughput (fixed service time)

The *Available Bandwidth* indicates the free bandwidth of parent node. The *S* field is share rate of the child node. The *Service rate* is the parent node maximum service rate.

In these simulations we implemented backpressure (25%) and (75%), CCF and PCCP algorithms for comparison purposes in our simulations. As we all know, backpressure refers to the CODA's hop-by-hop congestion-control mechanism [16]. If a sensor node is congested (based on channel utilization and buffer occupancy), it periodically sends backpressure messages to its child nodes, which will reduce their forwarding rates by a certain percentage (i.e. (25%) or (75%) reduction percentages in simulations) in response to a backpressure message. If the child nodes are data source node, it reduces the new data generating rate by the same percentage.

4.1 Throughput

We first start the simulation on normal situation in which all nodes are generating packets and have the same priority and importance. This can be the case in the hospital Intensive Care Unit (ICU) or other sections in which all patients are in bed and have the same priority and importance. As shown in Fig. 10, similar to PCCP, we consider a fixed service time for all sensor nodes. Figure 10 shows the sum of the throughput for all nodes. We define network total throughput as the packet receiving rate of the sink over the total bandwidth of the sink. For example if the packet receiving rate of the sink is 900 packets/s and the total bandwidth of the sink is 1000 packets/s, then the network total throughput becomes 0.9.

Comparison between CCF, PCCP and our proposed protocol showed in Fig. 10. Among the five compared protocols, the CCF, PCCP and our proposed protocol have the most throughputs which are between 0.9 and 1. As shown in the figure, the average throughput of CCF, PCCP and the proposed protocol are equal to 0.97, 0.96 and 0.99, respectively. According to our simulation results the average throughput of backpressure (25%) and backpressure (75%) are equal to 0.85 and 0.72 respectively. The reason is that backpressure algorithms are hard to adapt to an appropriate level of bandwidth share, while the other three protocols assign a



Fig. 11 Total normalized throughput (node 1, 9 and 13 are off in the interval [40 s, 80 s]

better share of the bandwidth (based on a certain policy). Because of the low throughput of backpressure algorithms, these two algorithms have been ignored in presenting the results.

In Fig. 11, the sensor nodes 1, 9 and 13 turn off in the time interval [40 s,80 s]. This can be the case in which some of the patients do not transmit data in some time intervals, so we need other nodes could scale-up their scheduling rate. As mentioned before, CCF cannot effectively allocate the remaining bandwidth, so it has a lower throughput in the interval [40 s, 80 s] when some nodes do not have sufficient source traffic. Proposed protocol and PCCP can use the remaining bandwidth better and so achieve more throughput than other protocols. The proposed protocol includes the available bandwidth in the calculation of the optimum new share rates, so its throughput is better than the PCCP protocol in the interval [40 s, 80 s].

When nodes 1, 9 and 13 generates small traffic or stop sending data during the interval [40 s, 80 s], our proposed protocol determines that the congestion level gets lower in these nodes, so it calls optimization function and calculates new portions for child nodes (i.e., nodes 10 and 11 for node 13). As a result, the proposed protocol maintains higher throughput during this time interval as shown in Fig. 3.

4.2 End-to-End Delay and Jitter

We consider the end-to-end delay as a fundamental parameter which is very important in critical applications like healthcare. End-to-End delay is the time taken for a packet to be transmitted from a source node to the sink. In Fig. 12 end-to-end delay for different protocols has been shown. This figure shows that, the end-to-end delay of the proposed protocol is less than the other algorithms. Low end-to-end delay is expected for critical patient's packets.

We define jitter as the time between each two end-to-end delays. In a remote healthcare system that monitors patients or elderly people, some video or acoustic sensors may be used. These kinds of sensors are not jitter-tolerant. In these applications, jitter is a critical issue and should be avoided or lowered to a minimum possible value.

As shown in Fig. 12, the end-to-end delay for backpressure algorithms (25%) and (75%) is higher than CCF, PCCP and the proposed protocols. The end-to-end delay for backpressure algorithms (25%) and (75%) is above 1 s while it is under 1 s for the other three protocols.

In the simulated scenario, the jitter is shown for CCF, PCCP and proposed protocols. In this scenario, sensor nodes 1, 9 and 13 were inactive (turned off) during the time interval [40 s, 80 s]. As can be seen from Fig. 13, jitter of our proposed protocol is lower than the



Fig. 12 End to end delay



Fig. 13 Jitter

CCF, and PCCP protocols. The reason is that the proposed protocol applies an optimization function for child nodes which uses network situations and available bandwidth of the parent node in order to assign the optimum sending rate. This decreases the end to end delay and jitter.

4.3 Packet Loss

We define packet loss probability as :

$$P_{loss} = (N_{loss} / N_{loss} + N_{recived})$$
(16)

where N_{loss} is the number of dropped packets in each epoch time (epoch time = 1 s) in the network. This can be because of AQM *Pdrop* in Eq. (7) or queue overflow or packet error rate (PER) in the network. $N_{recived}$ is the number of receiving packets at the epoch time.

The PCCP uses a single queue for all transit traffic and one for local source traffic, but we use one queue for all traffic, and divide it into $(N_i + 1)$ virtual queues; One virtual queue for local source traffic and the others for N_i child nodes, where N_i is equal to the number of Childs.

Figures 14 and 15 present the packet loss probability and cumulated packet loss over time respectively. As Fig. 14 shows the packet loss probability of CCF, PCCP and proposed



Fig. 14 Packet loss probability (buffer size = 150 packets)



Fig. 15 Cumulated packet loss over time (buffer size = 150 packets)

protocols are lower than backpressure algorithms, because in the occurrence of congestion, backpressure algorithms cannot allocate optimized rates for the source and the child nodes and hence, their packet loss rate is higher.

As PCCP maintains two different queues, if the sending rate of a source or child node is more than its share, its packet will be dropped immediately and not allowed to use the other child node's queue share. Also, in CCF method an equal portion is allocated to each child node at every moment. Consequently, packet loss increases in both PCCP and CCF.

The proposed protocol in the first step defines a dynamic weight for the child nodes using Eq. (4). According to this weight, each child node's portion is determined in each time interval. Then, it exploits its new AQM algorithm which allows if any child node's allocated queue is going to be filled and there are empty places in other virtual queues, they are utilized instead of dropping the packets. When a child node uses the other one's portion, in the second phase of congestion control based on automaton, the congestion level of the node is increased. Considering the empty bandwidth along with the weight and importance of child nodes and by means of the optimization function, the new portions will be divided among them so that each node's input rate is equal to its output rate which leads to reduction in packet loss probability.

Therefore, the packet loss in the proposed method is less than CCF and PCCP methods even at the beginning of data transmission (Fig. 14). This is very important especially for



Fig. 16 Mean queue size at the node closest to the sink (buffer size = 150)

sensitive traffic flows of HWSNs in which a patient's status may change suddenly and there is little time for analyzing the situation and making decision about him/her. As can be seen, the other methods have usually more packet loss at the beginning of the transmission.

4.4 Queue Size

The queue size is the number of packets in the queue. Figure 16 illustrates the mean queue size in the node 14, the closest node to sink over time. It means that queue size is a major metric in delay measurement. The more queue size will cause more delay. Our proposed protocol considers the optimum portion for child nodes by the optimization function so that input rates get always lower than the output rates in each node.

Using the optimization function along with considering the available bandwidth in each node's output and the priority of each child node, our proposed protocol allocates the suitable portion to each child node so that the sending rate of any child equals its parent's available bandwidth. In this way, the parent's queue length can be managed so that the overflow doesn't occur.

It is clearly seen in Fig. 5 that the queue length of our proposed protocol is less than PCCP in which the queue length may exceed half of the buffer in some moments leading to more packet loss as well as more delay. In contrast, it is clear that the proposed AQM provides a reasonable stability for mean queue size. The proposed algorithm may stabilize the queue length around a desired level.

4.5 Energy Performance Comparison

In this section, performance of the proposed protocol in terms of energy efficiency is compared with the mentioned protocols. In different simulations, remained energy value and consumed energy value are considered.

According to the results shown in Fig. 17, the normalized fairness of the network in energy consumption for the proposed protocol is more than the other protocols. Equation (17) calculates the variance of normalized remaining energy of the network's nodes to the average remaining energy (Ave) of the total network. In the worst case scenario, half of the nodes have no energy and half of them will remain full. If we normalize the equation, we will achieve the normalized fairness equation. In Eq. (17), energy is the remaining energy of the node at the end of the simulations. The Fairness parameter shows that, if the remaining



Fig. 17 Normalized fairness over energy



Fig. 18 Energy consumption in closest node to sink

energy of the nodes becomes closer to each other, a protocol will be more successful. If fairness is equal to one, the network has the best and the fairest case (when all nodes have the same remained energy). However, if it is equal to zero, we have the most unfairness cases of energy consumption. As shown in Fig. 11, the proposed protocol fairness is more successful rather than the other protocols.

$$DEV = \sum_{i=1}^{n} (Energy_i - Ave)^2$$

Fairness = 1 - DEV/DEV Worst (17)

Because the sensor nodes used in healthcare are of different types and hence they have different lifetimes, we assumed a network of sensor nodes with different initial energies. The initial sending rate of all nodes is 10 packets/s. In such conditions, we can observe that the nodes which use the proposed algorithm show the better fairness in energy consumption.

Since in a sensor network, all the packets are forwarded towards the sink, the nodes closer to the sink pass the most traffic and hence usually uses up their energy before the other nodes. In Fig. 18, we show the energy of the closest node to the sink. As can be seen, the proposed method has increased the network lifetime using an improved traffic management. (Note that we define the network lifetime as the time the first node in the network will die.)

At the end, the total result of Figs. 17 and 18 shows that, the energy performance of the proposed protocol is more efficient.



Fig. 19 a Node 0 source rate, b Node 5 source rate

4.6 Source Data Transfer Rate Adjustment

Figure 18 shows the sending rate of the source node 0 in different protocols. In this diagram, the initial sending rate of node 0 and node 5 has been assumed 100 and 10 packets/s is respectively. The service rate of the node near the sink, namely node 14 is 700 packets/s. Also, all the nodes have the same significance and priority. This can happen for a part of the HWSN environment in which all of the patients have the same initial status.

One of the important problems in HWSN is related to the time when a node suddenly starts its operations in a high traffic rate. In this situation the primary times of sending data have a particular importance so that the most significant decisions for one's health provision are made in these times.

Although CCF and PCCP methods adjust the rate properly, but they do this only based on nodes' priority and do not take into account the traffic conditions of a specific node. As it can be seen in Fig. 19, the proposed method considers the traffic conditions in addition to the importance and priority of nodes. Thus, it doesn't reduce the sending rate immediately and does it slowly with respect to the initial sending rate of the nodes. It will cause transferring the most important data before any reduction of sending rate, in the primary times.

The proposed algorithm achieves the largest total source data send rate due to its efficient rate adjustment method in comparison with CCF and PCCP protocols.

5 Conclusion

In this paper we presented a congestion management protocol for healthcare wireless sensor networks (HWSNs). The best scenario for the proposed protocol can be stationary patients in a special ward of a hospital or a clinic, particularly apoplectic or brain dead patients. This protocol consists of two stages. In the first stage, a novel Active Queue Management (AQM) scheme is proposed to avoid congestion and provide quality of service (QoS). This scheme uses separate virtual queues on a single physical queue to store the input packets from each child node based on importance and priority of the source's traffic. Using this scheme when one of the child nodes has a free space on its own virtual queue, others can use the free space if necessary.

According to the proposed AQM, if the incoming packet is accepted, the second stage starts. In the second stage, three mechanisms are used to control congestion. The proposed protocol detects congestion by a three-state machine and virtual queue status; it adjusts the child's sending rate by an optimization function.

Simulation results confirmed that the proposed protocol is more efficient than CCF, PCCP and backpressure algorithms in terms of packet loss, energy efficiency, end-to-end delay and fairness. As a future direction of this research, efficiency of the proposed protocol for the mobile patients in the hospital or city can be studied.

References

- Vieira, M. A. M., Coelho, C. N., Jr., da Silva, D. C., Jr., & da Mata, J. M. (2003). Survey on wireless sensor network devices. In *Emerging technologies and factory automation*, 2003. Proceedings. ETFA '03. IEEE conference, September 16–19, 2003 (Vol. 1, pp. 537–544, vol. 531). doi:10.1109/etfa.2003.1247753.
- Alemdar, H., & Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15), 2688–2710. doi:10.1016/j.comnet.2010.05.003.
- Boulis, A., Smith, D., Miniutti, D., Libman, L., & Tselishchev, Y. (2012). Challenges in body area networks for healthcare: The MAC. *Communications Magazine*, *IEEE*, 50(5), 100–106. doi:10.1109/mcom.2012. 6194389.
- Baker, C. R., Armijo, K., & Wright, P. K. (2007). Wireless sensor networks for home health care. Paper presented at the proceedings of the 21st international conference on advanced information networking and applications workshops—Volume 02.
- 5. Dishongh, T. J., & McGrath, M. E. (2010). Wireless sensor networks for healthcare applications. London: Artech House.
- Zhao, W., Ning, S., & Liu, L. (2011). Wireless sensor networks for in-home healthcare: Issues, trend and prospect. In *International conference on computer science and network technology (ICCSNT), 2011*, December 24–26, 2011 (Vol. 2, pp. 970–973). doi:10.1109/iccsnt.2011.6182123.
- Darwish, A., & Hassanien, A. E. (2012). Correction: Darwish, A. and Hassanien, A.E. Wearable and implantable wireless sensor network solutions for healthcare monitoring [published erratum]. *Sensors* (*Basel*), 12(9), 12375–12376. doi:10.3390/s120912375.
- Chen, B., & Pompili, D. (2011). Transmission of patient vital signs using wireless body area networks. *Mobile Networks and Applications*, 16(6), 663–682. doi:10.1007/s11036-010-0253-7.
- Lo, B., Atallah, L., Aziz, O., ElHew, M., Darzi, A., & Yang, G.-Z. (2007). Real-time pervasive monitoring for postoperative care. In S. Leonhardt, T. Falck & P. Mähönen (Eds.), 4th international workshop on wearable and implantable body sensor networks (BSN 2007) (Vol. 13, pp. 122–127, IFMBE proceedings). Berlin: Springer.
- Tia, G., Massey, T., Selavo, L., Crawford, D., Bor-rong, C., Lorincz, K., et al. (2007). The advanced health and disaster aid network: A light-weight wireless medical system for triage. *IEEE Transactions on Biomedical Circuits and Systems*, 1(3), 203–216. doi:10.1109/tbcas.2007.910901.
- Wang, C., Li, B., Sohraby, K., Daneshmand, M., & Hu, Y. (2007). Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on Selected Areas in Communications*, 25(4), 786–795. doi:10.1109/jsac.2007.070514.
- Ee, C. T., & Bajcsy, R. (2004). Congestion control and fairness for many-to-one routing in sensor networks. Paper presented at the proceedings of the 2nd international conference on embedded networked sensor systems, Baltimore, MD, USA.

- Chonggang, W., Sohraby, K., Bo, L., Daneshmand, M., & Yueming, H. (2006). A survey of transport protocols for wireless sensor networks. *Network, IEEE*, 20(3), 34–40. doi:10.1109/mnet.2006.1637930.
- Rathnayaka, A. J. D., & Potdar, V. M. (2013). Wireless sensor network transport protocol: A critical review. *Journal of Network and Computer Applications*, 36(1), 134–146, doi:10.1016/j.jnca.2011.10. 001.
- Yaghmaee, M. H., & Adjeroh, D. A. (2009). Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks. *Computer Networks*, 53(11), 1798–1811. doi:10.1016/j.comnet.2009.02.011.
- Wan, C.-Y., Eisenman, S. B., & Campbell, A. T. (2003). CODA: Congestion detection and avoidance in sensor networks. *Paper presented at the proceedings of the 1st international conference on embedded networked sensor systems*, Los Angeles, California, USA.
- Xiaoyan, Y., Xingshe, Z., Rongsheng, H., Yuguang, F., & Shining, L. (2009). A fairness-aware congestion control scheme in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 58(9), 5225– 5234. doi:10.1109/tvt.2009.2027022.
- Misra, S., Tiwari, V., & Obaidat, M. (2009). LACAS: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, 27(4), 466–479. doi:10.1109/jsac.2009.090510.
- Tao, L., & Yu, F. (2009). ECODA: Enhanced congestion detection and avoidance for multiple class of traffic in sensor networks. *Paper presented at the proceedings of the 15th Asia-Pacific conference on communications*, Shanghai, China.
- Samimi, M., Rezaee, A., & Yaghmaee, M. H. (2012). Design a new fuzzy congestion controller in wireless sensor networks. *International Journal of Information and Electronics Engineering*, 2(3), 395– 399. doi:10.7763/IJIEE.2012.V2.123.
- Esmailpour, B., Rezaee, A., & Abad, J. (2010). Congestion avoidance and energy efficient routing protocol for WSN healthcare applications. In T.-H. Kim, T. Vasilakos, K. Sakurai, Y. Xiao, G. Zhao, & D. Śløgonezak (Eds.), *Communication and networking* (Vol. 120, pp. 1–10, Communications in Computer and Information Science). Berlin: Springer.
- Yaghmaee, M. H., & Adjeroh, D. (2008). A new priority based congestion control protocol for wireless multimedia sensor networks. In *International symposium on a world of wireless, mobile and multimedia networks, WoWMoM 2008*, June 23–26, 2008, pp. 1–8. doi:10.1109/wowmom.2008.4594816.
- Yaghmaee, M., Bahalgardi, N., & Adjeroh, D. (2013). A prioritization based congestion control protocol for healthcare monitoring application in wireless sensor networks. *Wireless Personal Communications*, 1–27. doi:10.1007/s11277-013-1169-x.
- Brahma, S., Chatterjee, M., Kwiat, K., & Varshney, P. K. (2012). Traffic management in wireless sensor networks: Decoupling congestion control and fairness. *Computer Communications*, 35(6), 670–681. doi:10.1016/j.comcom.2011.09.014.
- Rezaee, A. A., Yaghmaee, M. H., & Rahmani, A. M. (December 2011). Class based congestion control method for healthcare wireless sensor networks. *International Geoinformatics Research and Development Journal*, 2(4).
- Samiullah, M., Abdullah, S. M., Bappi, A. F. M. I. H., & Anwar, S. (2012). Queue management based congestion control in wireless body sensor network. In *International conference on informatics, electronics* and vision (ICIEV), 2012, May 18–19, 2012 (pp. 493–496). doi:10.1109/iciev.2012.6317349.
- Kunniyur, S. S., & Srikant, R. (2004). An adaptive virtual queue (AVQ) algorithm for active queue management. *IEEE/ACM Transactions on Networking*, 12(2), 286–299. doi:10.1109/tnet.2004.826291.
- Firoiu, V., & Borden, M. (2000). A study of active queue management for congestion control. In *INFO-COM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies Proceedings.* IEEE, March 26–30, 2000 (Vol. 3, pp. 1435–1444, vol. 1433). doi:10.1109/infcom.2000. 832541.
- Wang, H., Liao, C., & Tian, Z. (2011). Effective adaptive virtual queue: A stabilising active queue management algorithm for improving responsiveness and robustness. *Communications, IET*, 5(1), 99– 109. doi:10.1049/iet-com.2009.0700.
- Reddy, T. B., & Ahammed, A. (2008). Performance comparison of active queue management techniques. Journal of Computer Science, 4(12), 1020–1023. doi:10.3844/jcssp.2008.1020.1023.
- Chen, W., Fan, X.-L., & Zhang, J. (2012). An adaptive BLUE algorithm for active queue management. Paper presented at the proceedings of the 2012 international conference on electronics, communications and control.
- Narendra, K.S., & Thathachar, M. A. L. (1989). Learning automata—an introduction. Englewood Cliffs, NJ: Prentice Hall.

- 33. Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). *Fundamentals of queueing theory*. London : Wiley-Interscience.
- 34. http://www.opnet.com.

Author Biographies



Abbas Ali Rezaee was born on November 1975 in Mashhad, Iran. He received his BS in computer engineering from Shahid Beheshti University, Tehran, in 1998,the M.S. in computer engineering from Esfahan University, Esfahan, in 2001. He is a Ph.D. candidate in computer engineering at computer engineering department, Islamic Azad University (IAU), Science and Research Branch, Tehran, Iran. His research interests are in QoS in distributed systems, networks, ad hoc and wireless sensor networks, scheduling algorithms.



Mohammad Hossein Yaghmaee was born on July 1971 in Mashad, Iran. He received his B.S.degree in Communication Engineering from Sharif University of Technology, Tehran, Iranin 1993, and M.S. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 1995. He received his Ph.D. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 2000. He has been a computer network engineer with several networking projects in Iran Telecommunication Research Center (ITRC) since 1992. From November 1998 to July 1999, he was with Network Technology Group (NTG), C&C Media research labs., NEC Corporation, Tokyo, Japan, as visiting research scholar. From September 2007 to August 2008, he was with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA as a visiting associate professor. He is the author of 3 books all in Farsi language. He has published more than 85 international conference and journal papers. His research inter-

ests are in Wireless Sensor Networks (WSNs), traffic and congestion control, high speed networks including ATM and MPLS, Quality of Services (QoS) and fuzzy logic control.



Amir Masoud Rahmani received his B.S. in computer engineering from Amir Kabir University, Tehran, in 1996, the M.S. in computer engineering from Sharif University of technology, Tehran, in 1998 and the Ph.D. degree in computer engineering from IAU University, Tehran, in 2005. He is assistant professor in the Department of Computer and Mechatronics Engineering at the IAU University. He is the author/coauthor of more than 80 publications in technical journals and conferences. He served on the program committees of several national and international conferences. His research interests are in the areas of distributed systems, ad hoc and sensor wireless networks, scheduling algorithms and evolutionary computing