Journal of Network and Computer Applications ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks

Abbas Ali Rezaee [a,*], Mohammad Hossein Yaghmaee [b], Amir Masoud Rahmani [a], Amir Hossein Mohajerzadeh [b]

[a] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[b] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 11 May 2012
Received in revised form
17 December 2012
Accepted 23 February 2013

Keywords:
Wireless sensor networks
Congestion control
Healthcare
Routing protocol
Optimization

ABSTRACT

Wireless sensor networks consist of a large number of small, low-power sensors that communicate through wireless links. Wireless sensor networks for healthcare have emerged in recent years as a result of the need to collect data about patients' physical, physiological, and vital signs in the spaces ranging from personal to hospital and availability of the low cost sensors that enables this data collection. One of the major challenges in these networks is to mitigate congestion. In healthcare applications, such as medical emergencies or monitoring vital signs of patients, because of the importance and criticality of transmitted data, it is essential to avoid congestion as much as possible (and in cases when congestion avoidance is not possible, to control the congestion). In this paper, a data centric congestion management protocol using AQM (Active Queue Managements) is proposed for healthcare applications with respect to the inherent characteristics of these applications. This study deals with end to end delay, energy consumption, lifetime and fairness. The proposed protocol which is called HOCA avoids congestion in the first step (routing phase) using multipath and QoS (Quality of Service) aware routing. And in cases where congestion cannot be avoided, it will be mitigated via an optimized congestion control algorithm. The efficiency of HOCA was evaluated using the OPNET simulator. Simulation results indicated that HOCA was able to achieve its goals.

1. Introduction

Wireless Sensor Networks (WSNs) have been widely applied in different areas such as healthcare monitoring (Akyildiz et al., 2002; Armijo et al., 2007; Cao et al., 2005). They have inherent characteristics unlike traditional wireless networks. Sensor nodes have scarce resources for computation, storage, communication bandwidth, and, most importantly, energy supply. So far, extensive studies have been done on different layers of WSNs (Akkaya and Younis, 2005; Sohraby and Wang, 2006). The event-driven nature of WSNs leads to unpredictable network load, especially in healthcare applications. Typically, WSNs carry low traffic load when there are no special events. But the occurrence of important events may cause burst traffics which lead to congestion in the network. Transport protocols control congestion in end to end or cross layer manner.

Nowadays, Healthcare aware Wireless Sensor Networks (HWSNs) have received a great attention due to the properties of WSNs such as reliability, interoperability, efficiency, wearability, low-power consumption and inexpensiveness. One of the applications of WSNs is remote monitoring of patients by doctors and nurses which eliminates the need to be physically present in the patient sites (Cao et al., 2006). Figure 1 shows different sensors attached to patients being capable of sensing patient information which can be sensitive (vital signs, such as the heart rate and breathing condition) or non-sensitive (motional signs, such as legs sensors). The received information can be transmitted to the control center with the help of neighboring nodes. Sensitive information needs low delay and low packet loss while non-sensitive data can tolerate more delay and more packet loss. We restricted ourselves to healthcare applications which require stationary sensor nodes (they do not change their locations for at least a few hours).

In medical emergencies, it is quite likely that the sensors placed in the different patients sense and transmit vital patient information very frequently and simultaneously. This leads to increased likelihood of network congestion in such applications. Congestion in WSNs leads to dropping of packets at the nodes, increased consumption of the limited energy in the nodes and

* Corresponding author. Tel.: +98 9153152739.
E-mail addresses: aa.rezaee@srbiau.ac.ir (A.A. Rezaee), hyaghmae@ferdowsi.um.ac.ir (M.H. Yaghmaee), Rahmani@srbiau.ac.ir (A.M. Rahmani), ah.mohajerzadeh@stu-mail.um.ac.ir (A.H. Mohajerzadeh).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.02.014

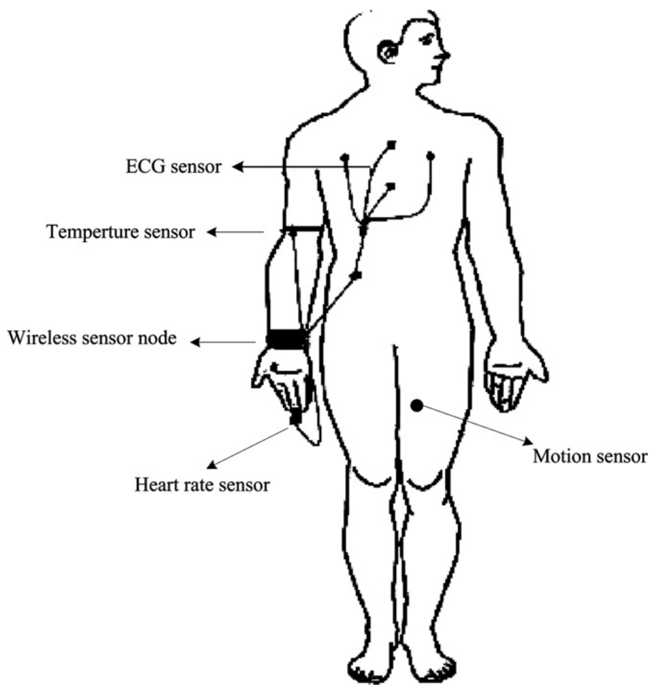

Fig. 1. Different sensors on patient's body.

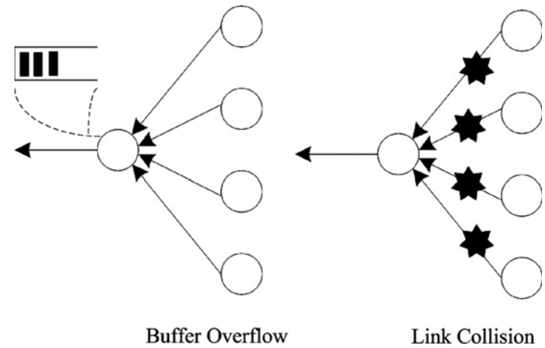

Fig. 2. Causes of congestion in wireless sensor networks.

reduction of the throughput of the network. In life-critical applications involving large numbers of patients, congestion is extremely undesirable and may lead to the death of a patient. However, timely arrival of the packets at their destinations ensures the safety and survival of the patients. Obviously, complete elimination of congestion is unlikely. But, it is possible to significantly reduce the effects of congestion, i.e., significantly decreasing the number of packets that get dropped due to congestion, the large amount of unwanted consumption of the limited energy at the sensors and increasing the number of packets that get successfully delivered with respect to the number of packets which are sent from the different nodes.

We addressed the problem of congestion by proposing a new approach to avoid it. In this approach, congestion will be avoided by distributing packets through multiple routes and if congestion still occurs, we run an optimized congestion control algorithm.

Congestion control algorithms are classified as source based or network based. Source based algorithms are deployed at the end host where the transport protocol is responsible for detecting congestion in the network. Network based algorithms, on the other hand, are implemented in the intermediate network devices, especially routers. Based on the degree of congestion detected in the network, source based algorithms adapt the rate at which the application is sending traffic. This mechanism, more popularly known as end to end congestion control is employed by transport protocols such as the Transmission Control Protocol (TCP). In network based algorithms, the intermediate network equipments are responsible for detecting oncoming as well as subsisting congestion and provide feedback to the sender for indicating the situation. Source based algorithms work well for traffic that is responsive to congestion e.g. TCP traffic. However non-sensitive traffic e.g. User Datagram Protocol (UDP) traffic may still cause congestion due to its greedy behavior. Thus, the need arises for network based congestion avoidance and control mechanisms.

There are different factors involved in the design of transport protocols for sensor networks: congestion control and reliable data delivery. Since most data move from sensor nodes to the sink, congestion is likely to occur around the sink. In order to increase the speed of the connection process, improve efficiency and decrease transmission delay, sensor network transport protocols should facilitate the process of the initial connection or use protocols without connection. Most applications in wireless networks are passive, meaning that the network is monitored inactively and waits for an event before sending data. When an event occurs, these applications may have quantitative packets to send.

Transport control protocols should treat different types of sensor network nodes fairly. If possible, inter-layer optimization should be considered in the design of the transport protocol. For example, if a routing algorithm informs the transport protocol about route failure, the protocol can infer that the packet loss is not caused by congestion, but due to route failure. In such a condition, the sender can maintain its present rate.

Basically, two factor causes congestion in sensor networks (see Fig. 2). The first is when the packet arrival rate is higher than packet service rate which occurs mostly in nodes closer to the sink. The second is the performance at the link level including competition, collision and bit error. This type of congestion occurs on the link.

Congestion control is important in traditional TCP networks as well as wireless sensor networks. QTCP (active Queue management support TCP) (Farzaneh et al., 2011) is one of the latest work for controlling congestion in traditional TCP networks with the help of AQM (active queue management). Because of the inherent nature and the main goal of WSN to transmit data in an energy efficient way traditional congestion control protocols cannot directly used in wireless sensor network (Abbas et al., 2008). So there was a need to design transport protocols for wireless sensor network that could deal with three mechanisms in case of congestion: congestion detection, congestion notification, and rate adjustment.

There are different congestion detection methods that are employed in wireless sensor networks. One common mechanism is the use of queue length (Wan et al., 2003; Balakrishnan et al., 2004; Adjeroh and Yaghmaee, 2008), packet service time (Bajcsy and Ee, 2004) or the ratio between service time and the time between packets in an intermediate node (Daneshmand et al., 2007). For sensor networks using MAC layer protocols such as CSMA, channel load can also be used as a tool for congestion detection (Wan et al., 2003). When congestion is detected, transport protocols transfer congestion information from the congested nodes to other nodes on the route to the sink or the source nodes that have had a part in detecting congestion. Congestion information can be as small as a binary CN (Congestion Notification) bit (Wan et al., 2003; Balakrishnan et al., 2004; Akyildiz et al., 2003) or contain more information such as permitted data rate (Bajcsy and Ee, 2004) or congestion degree as in Daneshmand et al. (2007). Sensor nodes can adjust their sending rate after receiving congestion notification. If a bit CN is received, the AIMD (Additive Increase and Multiplicative Decrease) method or other types of it are usually applied. However, if more comprehensive congestion information is available, rate adjustment can be done more accurately (Bajcsy and Ee, 2004; Daneshmand et al., 2007).

Today, different algorithms presented in the field of WSN, are more suitable for some applications with regard to their output parameters and diagrams. In most of congestion control methods, the rate of packet sending is reduced immediately after congestion occurs and the lost sensitive packets are tried to be retrieved. This needs to an extra buffer in the previous nodes in order to keep the packets in it until receiving acknowledgment for them making these methods costly. Also, this makes sensitive traffic streams to reduce their sending rate. However, it is favorable that the sending rate for sensitive traffic streams isn't greatly reduced. So, in the proposed protocol, we have developed a congestion avoidance phase in which several paths are made primarily and the nearest one is allocated to sensitive traffic. Having multiple paths makes the traffic streams be distributed among them fairly based on their sensitiveness. This leads to less probability of packet loss especially for sensitive packets and hence less probability of sensitive packet rate reduction. The mentioned problem is more critical for the sensitive applications like health care in which the rate of sensitive data packets hasn't to be decreased. Also, we try to allocate an appropriate bandwidth for sensitive traffic along with using (PQ) priority queue approach in every node's output while servicing. This means that the more sensitive a packet is the sooner it is serviced. Totally, all the aforementioned policies being used in our proposed protocol, results in reduction of packet loss rate for sensitive data packets and consequently reduced delay for them until reaching destination. In this protocol, through a congestion control problem, some QoS parameters like delay, packet loss and network lifetime are compared with respect to the other methods. In the applications like healthcare, these parameters are very considerable because dropping of a sensitive packet may leads in a patient death. Also, in such application, delay in packet arrival will cause later decision and hence harming the patients. This is more noticeable in patients bedridden in ICU. If the battery of a sensor attached to a bedridden patient in ICU is discharged earlier than normal just when there is a fluctuation in his vital signs like blood pressure or hear beat, the patient's life can face a serious risk. The proposed protocol is composed of two main parts, routing and congestion control. Proposed routing protocol is a data centric protocol which composed of 4 different phase. The phases are discussed in Section 3 in details. We have evaluated the requirements of the healthcare applications, and consider them in designing proposed protocol. Forth phase of proposed routing protocol is data transmission. Similar to other networks, congestion may occur in network nodes. We have also proposed a congestion control mechanism which is discussed in Section 3.4.1. As its main job, congestion control mechanism adjusts nodes sending rate (especially source nodes) in order to manage congestion in intermediate nodes. In Section 4, simulation results have been presented. And finally in Section 5 we conclude the paper.

2. Related works

Different protocols have been proposed for congestion control. These protocols are different in terms of congestion detection, congestion notification, and rate adjustment mechanisms. In Fusion (Balakrishnan et al., 2004) congestion is detected by using queue length and controlled in a stop-and-start non smooth manner so that when congestion is detected and notified, neighboring nodes stop forwarding packets to the congested node immediately. In CODA (Congestion Detection and Avoidance) (Wan et al., 2003) protocol congestion is detected also by using queue length at the intermediate nodes. CODA protocol uses a combination of the present and past channel load and the level of buffer load in order to detect congestion in each receiver accurately and at low costs. It uses the selective backpressure method for congestion notification and the multi-source regulation for rate adjustment. CODA also controls the rate of flow of packets based on the AIMD algorithm.

CCF (Congestion control and fairness) protocol (Bajcsy and Ee, 2004) detects congestion based on packet service time. The CCF method carries out upstream congestion control using a scalable and distributed algorithm that ensures the fair delivery of the packets to the central station as well as removing congestion. CCF formulates congestion control and determines the number of downstream nodes, the average sending rate of the packets and the production rate in each sensor. PCCP (Priority-based Congestion Control protocol) (Daneshmand et al., 2007) is a priority based upstream congestion control protocol and measures a congestion degree as the ratio between packet arrivals and packet service time. PCCP also uses rate adjustment algorithm unlike that of the AIMD technique. It supports fairness in weighting sensor nodes. PCCP uses different degrees of priority indexes, so a sensor node with a higher priority index uses more bandwidth and injects more traffic. PCCP allows the application layer to cancel the priority index in a special area in each senor node. This aspect can be useful for a large number of sensor network applications. There are limitations for PCCP which include the lack of packet recovery. QCCP-PS (Queue based Congestion Control Protocol with Priority Support) (Adjeroh and Yaghmaee, 2008) is a queue based Congestion Control Protocol with Priority Support which uses the queue length as a congestion degree indicator. It controls the congestion with the packet priority based on the node priority for a WSN. QCCP-PS also improves the PCCP by controlling the queue more finely but it does not have any mechanism for handling prioritized heterogeneous traffic in the network. The sending rate of each traffic source in the QCCP-PS is increased or decreased based on its congestion degree and its priority index. The rate adjustment for each traffic source is based on its priority index as well as its current congestion degree.

ECODA (Enhanced congestion detection and avoidance) (Tao and Yu, 2010) uses dual buffer thresholds and weighted buffer difference for congestion detection. This method is different from traditional single buffer threshold methods (Wan et al., 2003; Bajcsy and Ee, 2004; Daneshmand et al., 2007). It can differentiate congestion level and dealt with them correspondingly. ECODA is composed of three mechanisms: 1) Using dual buffer thresholds and weighted buffer difference for congestion detection; 2) Flexible Queue Scheduler based on packet priority; 3) A bottleneck-node-based source sending rate control scheme in case of persistent congestion. ECODA also adopts hop-by-hop congestion control scheme for transient congestion. FACC (A Fairness-Aware Congestion Control scheme) (Huang et al., 2009) is a rate-based fairness aware congestion control protocol that divides all intermediate sensor nodes into near-source and near-sink nodes. This protocol detects congestion according to packet loss rate at the sink node. Every time a packet is lost in near-sink nodes, they send a WM (Warning Message) to the near-source nodes. When the near-source nodes receive WM, they send a CM (Control Message) to the source nodes to notify it of the updated sending rate. These messages cause overhead in the network and if any one of them gets lost because of path break, it may leads to a problem in congestion notification as well as rate adjustment

LACAS (Misra et al., 2009) is a Learning Automata-Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks which is more effective in dealing with congestion

Please cite this article as: Rezaee AA, et al. HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks. Journal of Network and Computer Applications (2013), http://dx.doi.org/10.1016/j.jnca.2013.02.014

problems in healthcare WSNs. The process of learning in this work is a learning loop consisting of the RE (Random Environment), and the LA (Learning Automata). The LA tries to learn the optimal action (send rate) offered by the RE. An important feature of LACAS is that it intelligently "learns" from the past and improves its performance significantly as time progresses. One of the limitations of this work is that its environment offers only binary responses for any action selected by the automaton.

Table 1 presents the characteristics of some of popular congestion control protocols.

In ESRT (Event-to-Sink Reliable Transport) (Akyildiz et al., 2003), the node monitors the congestion notification bit which is located in the packet header and obtains a common rate for all sensors so that no packet is lost. This method supports fairness but all sensors cannot adapt to the worst rate in the congestion situation. In this method, a new congestion control pattern capable of fair allocation of bandwidth is proposed. Of course, we expect that each flow should have a fair share of the bandwidth based on its production rate. In sensor networks, both the number of active flows and the accessible bandwidth change with time. Therefore, we cannot consider a fixed rate for each flow. In order to achieve a fair sharing of the bandwidth, the following method has been proposed.

3. The proposed protocol

The proposed protocol has been designed for congestion management in wireless sensor networks for healthcare applications. The main objective of the proposed protocol is to avoid, or if not possible, control congestion in wireless sensor networks. Similar to other data centric protocols such as REEP (Misra et al., 2008), Directed diffusion (Jang, 2007) and TPGF (Two-Phase geographic Greedy Forwarding) (Shu et al., 2010) HOCA has been developed in different phases. All these protocols use different phases to perform different crucial tasks. TPGF (Shu et al., 2010) also uses multipath transmission. And they are all developed for wireless sensor networks. HOCA considers two main parameters, energy and delay (besides lifetime and fairness). In all routing protocols which are developed for WSN, energy should be considered as a goal parameter. Delay is the main goal parameter for healthcare applications. HOCA considers two types of traffics: sensitive and non-sensitive. Sensitive traffics are designed to transfer high priority data (they need low delay) and non-sensitive traffic is designed to transfer normal traffic.

The proposed protocol works in the following phase: 1) request dissemination which is performed by the sink, 2) event occurrence report which is performed using packets that are forwarded from sensors located on patients body to the sink, 3) route establishment, 4) data forwarding and rate adjustment in case of congestion occurrence. In the design of HOCA, congestion control as the main objective affects other objectives. Routing has been considered as a part of the general objective. In this protocol, data are sent with different priorities. Therefore it can be used for healthcare remote monitoring applications whose networks contain data with different levels of importance and different priorities for different patients.

The proposed protocol acts as a cross layer. As mentioned before, in HOCA the duties of transport layers and the network are carried out simultaneously. First, the sink (the telemedicine center) sends its requirements (required information) to network nodes (sensors connected to the patient's body). In the meantime, any network node observing the event specified by the sink, will inform the sink with an event report (patient's condition) using the phase 2 procedure. In the second phase, the initial routing tables are formed. These tables are then used in the third phase where different routes are chosen in the final routing tables. The final tables are produced in the third phase depending on the priority of the transferred data. The fourth phase is the data forwarding phase in which the data recorded from the events observed by nodes are given to the sink. A large volume of data is moved in this phase; therefore a procedure for congestion control is needed. In HOCA, an adaptive procedure has been proposed for controlling source sending rates. This procedure is also carried out in the fourth phase in case of congestion.

Generally Fig. 3a–c shows the proposed protocol structure.

3.1. Request dissemination phase

This is the first phase in carrying out the routing protocol. In this phase, information required by the sink node (medical center) such as patients' vital signs should be sent to all network nodes. In other words, sink requirements are requested and distributed throughout the network based on different algorithms presented for distributing data in wireless sensor networks. However, the type of data is very important. In some situations, parameters may include highly sensitive information such as heartbeat or blood sugar level (for some patients such as those with diabetes).

This phase is started by the sink and the packets that are used for the implementation of this phase are the same structure. The proposed protocol uses the MLAF (Multimedia location aided Flooding) (Mohajerzadeh et al., 2010) algorithm in this phase. MLAF algorithm uses new methods to optimize energy consumption. Also, this algorithm supports distribution of data with different priorities. In applications where data distribution is carried out through the whole network, this method is not very effective. But, the option of data distribution with different priorities is very important for medical monitoring applications in which data distribution depends on the position of the target nodes (patients).

The following considerations should be taken into account in the structure of the packets to be transferred.

- The priority: in wireless sensor networks for medical monitoring applications, we may have different types of traffic with different

Table 1
Congestion control protocols for wireless sensor networks.

Protocol	Congestion detection	Congestion notification	Rate adjustment
Fusion (Balakrishnan et al., 2004)	Queue length	Implicit	Hop by hop rate adjustment
CODA (Wan et al., 2003)	Queue length and channel state	Explicit	Rate adjustment similar to AIMD
CCF (Bajcsy and Ee, 2004)	Packet service time	Implicit	Hop by hop rate adjustment
PCCP (Daneshmand et al., 2007)	Packet interval time and packet service time	Implicit	Hop by hop rate adjustment
QCCP-PS (Adjeroh and Yaghmaee, 2008)	Queue length	Implicit	Hop by hop rate adjustment
ECODA (Tao and Yu, 2010)	Dual buffer thresholds and weighted buffer difference	Implicit	Delay dependent
FACC (Huang et al., 2009)	Packet Drop At the Sink node	Explicit	Hop by hop rate adjustment

Please cite this article as: Rezaee AA, et al. HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks. Journal of Network and Computer Applications (2013), http://dx.doi.org/10.1016/j.jnca.2013.02.014

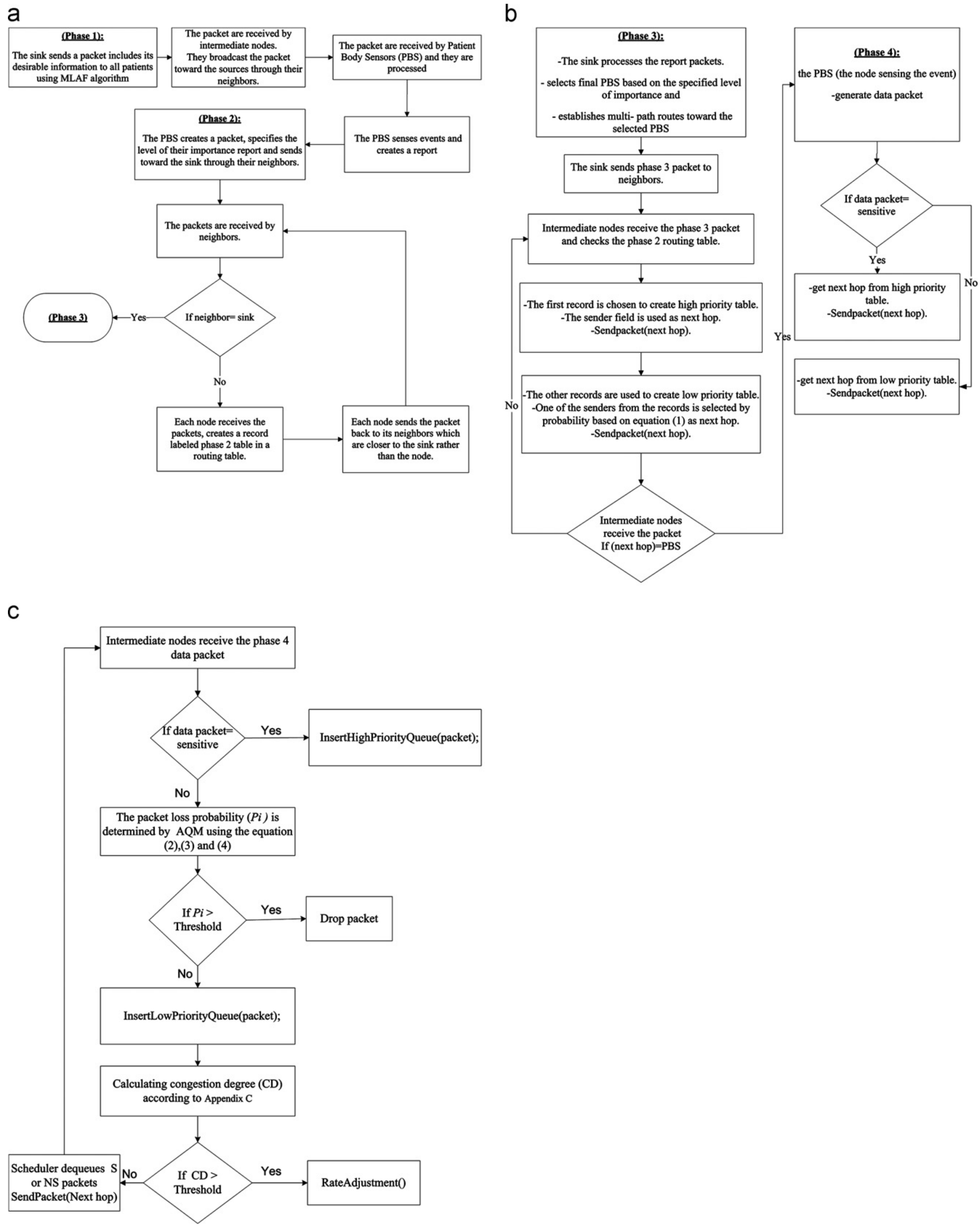

Fig. 3. (a) Flowchart of Phase 1 and Phase 2 in congestion avoidance part. (b) Flowchart of Phase 3 and Phase 4 in congestion avoidance part. (c) Flowchart of Phase 4 congestion control part.

characteristics. Therefore, while transferring a request, the type of patient and its priority should be specified.

- Time: we may have several requests for a certain patient in different points in time. The time of each request should be specified so that the nodes can determine the order of transmission for different requests. However, requests for some patients have an expiry period. After the end of the period the request is not considered anymore.

- Characteristics of the request: each request should contain the duties of the sensors connected to the patient's body. The specification of how these requests are answered should also exist in the nodes.

The above conditions are first met and then the packets are distributed in the network. For example, in a medical monitoring application the above conditions for a request (patient status report request) are defined in the following manner, respectively:

- Usually, in medical monitoring applications data related to vital signs have high priority; therefore this type of request is assigned a high priority. Also, we can consider other types of traffics for data related to patient movements which have a lower priority.
- Request dissemination time is set as the present time.
- It is specified in the request that each sensor on the patient's body should report vital signs out of the normal range to the sink with high sensitivity. The normal range is determined by the expert.

3.2. Event report phase

After the request dissemination phase, if a sensor senses an event based on its duty, it will report the sign to the sink according to the specifications. The report must have the required characteristics so that the sink can show the proper reaction.

In this phase, the information related to the occurring event is sent to the sink, however basic data related to the event are sent in the data forwarding phase. Moreover, the preliminaries of packet routing are also determined in this phase. For this purpose, the patient node creates a packet containing the information related to the sensed event and sends it to all its neighbors. Since nodes (patients) are aware of their own positions the packets are sent to the neighbors that are closer to the sink than the sender. The routing tables required for the routing of node data in the route from the packet to the sink will be provided. And the final routing will be carried out in the route forming phase.

After creating the packet (which we call phase 2 packet), if the nodes are aware of their positions this will lead to lower energy consumption for the protocol. However since we need to locate all the nodes it cannot be applied everywhere. It is worth noting that in applications where the request should only be sent to part of the network, nodes are aware of their positions.

After receiving the packet from phase 2, each node creates a record labeled phase 2 table in a routing table. The priority of the packet (compared to the priority of the traffic and the event in question), the source node, the sender, the length of the covered route and the number of covered hops are kept in this record. In the proposed protocol, each node has an ID that is placed in all outgoing packets. The length of the covered route is obtained from the length of the route from the source of the packet to the current node. After creating the record, the node sends the packet back to its neighbors. This procedure is repeated until the packet reaches the sink.

Note that from any source, there could be more than one record in each node's phase 2 table. The reason for this is that phase 2 packets may arrive at a node from different routes. Only packets with identical fields are ignored.

At the end of phase 2, each node has a routing table called phase 2, table which is used for final routing in phase 3. Records in phase 2 routing table determine the possible routes between the desired node and the source node sensing the event. Appendix A presents the pseudo code related to the phase 2 mechanism.

3.3. Route establishment phase

After the arrival of phase 2 packets at the sink, a type 3 confirmation packet is sent to the source node by the sink which notifies the source node to send its data to the sink for processing. Then, sensors from one or more patient(s) may send messages. In this stage, the sink chooses one or several nodes for the final transfer of data based on the information sent from source nodes. In phase 2 packets, each node specifies the level of its importance. For example, the heart beat sensor or the kinesthetic sensor connected to the patient's foot sends a message to the center and specifies the level of importance. The sink chooses the source node for the patient's report based on the specified level of importance.

Following the selection of the source, phase 3 packets are sent. As the phase 3 packet moves along the route, it creates a phase 3 routing table. Phase 3 routing table is the final routing table for routing the data sent from the source. The transfer confirmation depends on the priority of the sensed event. Two types of confirmations are considered, high priority confirmation (sensitive traffic) and low priority confirmation (non-sensitive traffic).

The sink checks the phase 2 routing table in order to send a high priority confirmation. The first record is chosen for sending confirmation. Phase 2 packets are then arranged chronologically in the phase 2 routing. Upon receiving a type 2 packet, the nodes place it in the first record. In fact, the number assigned to the packet record in the phase 2 routing table determines their time sequence. Since time is very important insensitive applications, the first record in the phase 2 routing table which is chronologically the first created record is chosen. However, in choosing records, the source node in the record is always considered. Moreover, only records in which the source node is the one chosen by the sink will be considered.

Each node forms two tables in phase 3: Phase 3 routing table with high priority and phase 3 routing table with low priority. During this phase, two tables are completed. Routing table of each node maintains the best routes to the sink through its neighbors which are closer to the sink. Considering the maximum number of neighbors for each node in WSN, the routing table will be practical and small.

When a node receives a phase 3 packet with high priority, it creates a high priority record for the packet in the phase 3 routing table. This table consists of the following components: sender (the source node of the receiving phase 3 packet with high priority), receiver (the destination node for the phase 3 packet with high priority), source node (the node sensing the event which is the final destination of the phase 3 packet) and type of application (this component will be used in networks designed for multiple applications). Based on what has been mentioned so far, each node chooses the first record from the phase 2 routing table as the next hop for the high priority phase 3 packets. This procedure will continue until the packet reaches the source. In fact, at the end of phase 3, a record is placed in the sensitive phase 3 routing table for each source.

What has so far been mentioned in Section 3.3 is related to high priority traffic. We will go on to explain the creation of low priority phase 3 routing table. From among the records in the phase 2 routing table, the sink considers the records chosen in relation to the source. For each of these records, the probability RSP_i is computed using the following equation:

$$RSP_i = \frac{TD_i/HC_i}{\sum_{\text{for all } j \in \text{Selected Records}}(TD_i/HC_i)} \tag{1}$$

where TD_i is the route length and HC_i is the number of hops for the ith record route. RSP_i is the route select probability of choosing the record as the next hop for the low priority phase 3 packets.

After determining RSP_is for all the records with the intended source, two records are chosen based on probability. Then, the low priority phase 3 packet is sent to these records. Different routes are chosen so that fairness is observed in energy consumption of the network nodes.

Each node receives a phase 3 packet with low priority and records it in its routing table. Then, through a procedure similar to that of the sink, the next two neighboring hop neighbors are chosen and the phase 3 packet is sent to them. All the characteristics are recorded in non-sensitive phase 3 routing records. Appendix B presents the pseudo code related to the phase 3 mechanism.

3.4. Data forwarding phase

Towards the end of phase 3, sensitive and non-sensitive phase 3 routing tables are created. Each node will contain asensitive phase 3 routing table and anon-sensitive phase 3 routing table. This provides multipath routing for our proposed protocol and can distribute packets through more than one path.

Depending on the type of the sensed event, the source node (the node sensing the event) can send its data to the sink after receiving sensitive traffic from phase 3. As mentioned before, all nodes including the source node have two types of routing table. Sensitive phase 3 routing table is used for sending sensitive data and non-sensitive phase 3 routing table is used for sending non-sensitive data.

In the sensitive phase 3 routing table, there is only one record toward the sink for each source. Each node receives sensitive traffic from the node in question and uses the traffic to send the record to the next hop. However, in each non-sensitive phase 3 routing table, there will be more than one record for each source in the table. Each record has a probability RSP_i based on which the next hop is chosen. The greater the RSP_i in the record, the more likely it will be chosen. Finally, a record will be chosen as the next hop and data are sent to this record.

Appendix C presents the pseudo-code related to the phase 4 mechanism.

3.4.1. Congestion control mechanism in intermediate nodes

Our goal is to provide routing and congestion management in WSN's for healthcare applications. Congestion management comprises two phases: Congestion avoidance and congestion control. Congestion avoidance is implemented by distributed routing algorithm (Section 3).

AQM schemes are one of the important mechanisms that provide quality of service and prevent congestion in IP networks that perform special operations in our protocol to achieve better performance for end flows. With these mechanisms, congestion is controlled and network degradation is avoided (Borden and Firoiu, 2000). Figure 4 depicts the queuing model on an intermediate node. In this figure a classifier has been provisioned in network layer. The purpose of a classifier is to classify different types of data and route them in their corresponding queues.

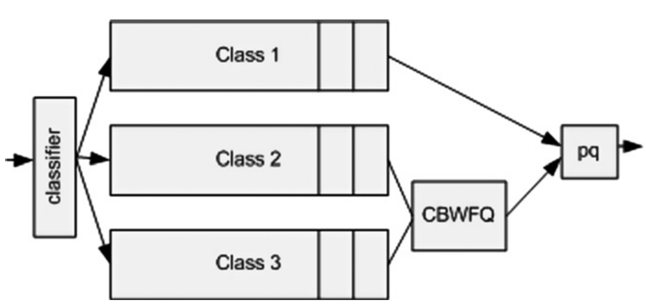

Fig. 4. The structure of an intermediate sensor node.

The type of data is located in the packet header. We define three type of traffic; high priority, low priority and control packets. Sensitive traffics are sent to class 1, non-sensitive traffic sent to class 2 and control packets are sent to class 3.

In our proposed protocol the CBWFQ (Class Based Weighted Fair Queue) scheduler (Fischer et al., 2008) is used with the addition of a PQ (Priority Queue). The use of PQ ensures low latency and more reliability for sensitive traffics. PQ allows sensitive traffics to be dequeued and sent first. While there is a class 1 packet in the queue, the scheduler sends class 1 packets out of queue. In order to provide fairness between class 1 and other classes, only 20 percent of network bandwidth is assigned to class 1 traffics, so using PQ scheduler does not cause unfairness. Indeed fewer queuing delays for class 1 traffics results from PQ scheduler which use for sensitive traffics.

3.4.1.1. Proposed AQM. AQM (Active queue management) has been proposed as a solution of preventing packet loss due to buffer overflow. In a network, if the queue length isn't suitably managed, the senders will continue sending the packets at the initial rate leading to packet loss. Therefore, using AQM methods, the proposed protocol reduces sending rate of packet senders before exceeding a determined threshold in order to prevent the packets from being dropped. In a sensitive application like health care, packet loss is a very important problem which has to be reduced as much as possible.

P_i is the packet loss probability which is determined by active queue management mechanism. HOCA uses a flexible procedure for queue management. The proposed procedure shares the queue in each node for the flows passing the node. However, the boundaries between queues are not fixed; meaning that if one of the active flows has free space in its queue, other flows facing a lack of space can use this free space on certain conditions. In other words, queues in Fig. 3 are separated virtually with flexible boundaries.

The probability of the drop (P_i) of a packet in ith queue is determined using the following equation:

$$P_i = \beta_1 . \delta qv_i - \beta_2 \cdot (1 - (\sum_{j=1}^{n} q_j / QL)) + P_i^{pri} \tag{2}$$

When a packet is received by the node, drop probability P_i is computed for the packet. Packet will be queued or dropped, based on P_i value. In fact, higher probabilities of loss for a flow show that the corresponding queue is in critical status with respect to the congestion. Therefore, the weight of P_i has been used directly in determining the sending rate and the degree of congestion in each node. P_i^{pri} is an initial value for P_i which is determined using Eq. (3). q_j presents the number of packets stored in jth virtual queue. δqv_i shows the level of variation in the length of the ith virtual queue. The value of δqv_i can be positive or negative. δqv_i is multiplied by coefficient β_1 as presented in Eq. (4). If δqv_i is positive, it will remain positive after multiplying by β_1 and will finally cause an increase in P_i. It means that if the variation in the flow queue length is positive (the queue size is prolonged) the packet loss probability and the probability of congestion are increased. β_2 specifies the flexibility of the flow queues. The expression $\sum_{j=1}^{n} q_j$ specifies the total used space in the node queue. Dividing the total by QL (total space in the node queue) gives us the percentage of used space in the node queue. Multiplying this value by β_2 will result in a number which reduces the value of P_i. In other words, the greater the free space in the queue the lesser the packet loss probability of the flows. However, the effect of this value depends on the β_2 parameter. β_1 and β_2 are determined based on node priority by the user:

$$P_i^{pri} = \begin{cases} 0 & \text{if} : iq_i \prec 2 \cdot QL/3 \cdot n \\ n^2.(q_i/QL) - 2 & \text{if} : q_i \geq 2 \cdot QL/3 \cdot n \end{cases} \tag{3}$$

$$\delta qv_i = \frac{q_i^{new} - q_i^{old}}{QL/n} \tag{4}$$

The parameters in Eqs. (2)–(4) are determined in a periodical manner. Therefore, in Eq. (4) the value of q_i^{old} is the queue length in the ith flow in the preceding calculation and the value of q_i^{new} is the queue length in the ith flow in the present calculation. Generally, in all the equations q_i shows the queue length in the ith flow. Parameter n is the number of node's neighbors.

3.4.1.2. Proposed rate adjustment. Congestion control, as mentioned in Section 1, consists of two parts: (1) Congestion notification and (2) rate adjustment. These procedures are done interestedly in a hop by hop manner, from the congested node to the source node with rate adjusting packets including children rate portions. As discussed in Section 3.4.1.1, AQM considers arrival rate ($q_i^{new} - q_i^{old}$) and queue length (q) in order to determine P_i. We use P_i as congestion indicator. Following using proposed Optimization problem (Eq. (5)) the upstream neighbor's rate adjustment is performed.

Since data are transferred in the data forwarding phase, it is likely to have network congestion in this phase. HOCA controls congestion by controlling the sender's data sending rate. However, congestion will also be prevented as far as possible, using multiple routing. The mechanism of congestion control comprises two parts: active queue mechanism in intermediate nodes and sender rate control mechanism. Active queue mechanism manages queues as well as detecting the level of congestion.

The following equations show the optimization problem which is used in order to control the forwarding rate.

$$\text{Min } F = \alpha \left[\sum_{i=1}^{n} \left(\frac{1-\theta_i}{1+\theta_i} \right) \cdot P_i \right] + (1-\alpha)\theta_c \tag{5.1}$$

$$\begin{aligned} &\theta_1 + \theta_2 + \cdots + \theta_n + \theta_c = 1 \\ &\forall\, i, 0 \le \theta_i \le 1 \\ &0 \le \theta_c \le 1 \\ &0 \le \alpha \le 1 \end{aligned} \tag{5.2}$$

$$\sum_{i=1}^{n} \theta_i (q_i.n/QL) \prec \theta_c^{NH} \tag{5.3}$$

In Eq. (5–1), n is the number of upstream neighbors and P_i is the drop probability computed by Eq. (2). The aim of optimization is to minimize the function of Eq. (5.1). Figure 5 clarifies the variables in Eq. (5). Eqs. (5.2) and (5.3)) present the optimization problem conditions. The importance of congestion control is determined by α parameter by the user.

The network has been considered identical in the design of the HOCA protocol. Therefore all links in the network are identical and have the same bandwidth. θ_1, θ_2,… θ_n are the shares of the first, second, … and nth sender, respectively. Each sender can determine its sending rate by multiplying θ by link bandwidth (which is the same in the entire network). θ_c is used as the congestion parameter. In fact, θ_c is part of the node's incoming bandwidth which cannot be used because of congestion. θ_c^{NH} is the current node's share for sending data which is determined by the next hop node (parent). For example, θ_1 is given to the preceding child node by the present node, and it is known as θ_c^{NH} in that node.

The optimization function (Eq. (5.1)) determines the congestion degree in the present node as well as the sending rate in the preceding child nodes. However, the maximum sending rate for the node (equal to the volume of arriving traffic plus the volume

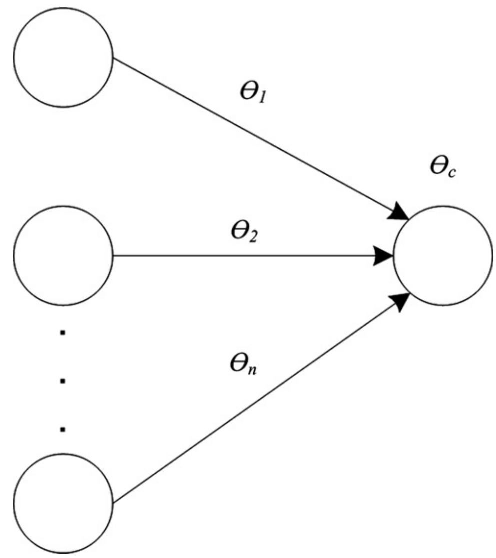

Fig. 5. The model used in intermediate nodes.

of produced traffic) corresponds to the rate determined by the next hop node. Eq. (3) is a statement of the mentioned condition. n is the number of upstream neighbors (preceding child nodes), q_i the number of packets in the queue related to ith traffic and QL/n is the maximum queue length in ith traffic.

Each node after receiving a set of packets runs Eq. (5.1) function and in case of detecting congestion or an increase in the sending rate of one of the senders, determines the sending rate of the preceding node(s) and provides this rate to the nodes. All θ parameters are in the range (0 and 1); 1 meaning that the entire bandwidth can be used and 0 meaning that no data can be sent.

Parameter α determines the importance of congestion in the network. The greater this parameter, the greater the importance of congestion control in the network. For example, if α is set as 1, the factor of θ_c becomes zero and the value of θ_c is practically 1. In this case, according to Eq. (5.2), the rate of all the senders will be zero.

4. Performance evaluation of the proposed protocol

MATLAB and OPNET (⟨www.OPNET.com⟩), are the two software used in investigating the performance of the proposed protocol. Eq. (2) optimization function along with other required functions were run in MATLAB. The simulation phase was carried out using OPNET. Since both software have been programmed based on C++ we have the option of creating links between the two. Therefore, the proposed protocol was simulated by linking the two software using C++ compiler.

In order to implement the proposed protocol, both MATLAB and OPNET software were used concurrently. The optimized function (2) with the other related functions are implemented in MATLAB software. Then OPNET calls MATLAB functions when needed. Figure 6 illustrates the used topology. Table 2 presents parameters used in simulations. Our optimization algorithm is very simple and consequently doesn't impose a heavy calculating load on the protocol. Each source node generates data units according to a Poisson process and the service rate is constant.

In addition to backpressure methods as factors of evaluating the proposed protocol performance, the REEP (Jang, 2007) protocol was also used. The algorithm proposed in (Yaghmaee and Adjeroh, 2009), considers shortcomings of CCF and PCCP and tries to solve them effectively. This algorithm is useful more for multimedia applications. It hasn't used routing phase and just

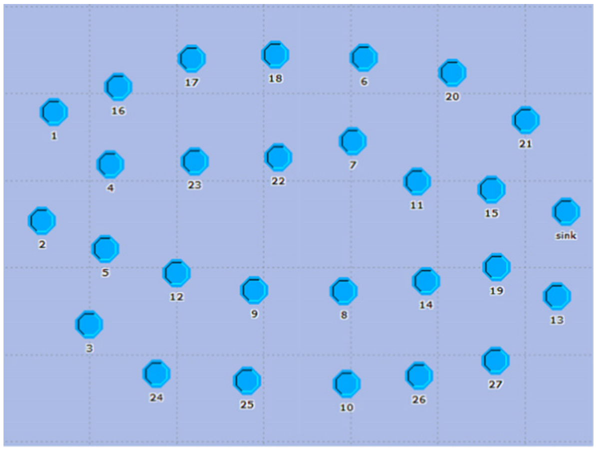

Fig. 6. The topology which is used in simulation.

Table 2
Simulation parameters.

Transmission range	40 m
Initial node energy	50 J
Type traffic	Sensitive and non-sensitive
Network area	200×200 m^2
Packet sent energy	12 mJ
Packet receive energy	10 mJ
Congestion detection epoch	Each 50 packet

reduces the rate of sending nodes in occurrence of congestion. But, in sensitive applications like medical health care, it isn't favorable to reduce the sending rate of packets; meaning that through allocating a proper bandwidth, it is tried to reduce the sending rate only for non-sensitive traffic not for sensitive one. Therefore, among routing algorithms, we chose REEP which is a multi-phase algorithm close to health care structure to be compared with our work. The major problem of REEP is ignoring the priority and forwarding all the packets from the same path. Thus in the proposed algorithm, the REEP problem was solved via a multi-path method along with support of packet prioritizing. REEP is a data-centric, energy efficient and reliable routing protocol for WSNs. This protocol follows different phases like other data centric protocols for routing which include: Sense event propagation, Information event propagation and Request event propagation. REEP also uses an energy threshold value in order to make the sensor nodes energy-aware. REEP also has five important elements, i.e. sense event, information event, request event, energy threshold value and request priority queue (RPQ). We use REEP besides back pressure in order to make a reasonable basis to find the proposed protocol efficiency.

The proposed protocol uses MLAF (Mohajerzadeh et al., 2010) algorithm in the first phase. MLAF is specially designed for data dissemination in wireless sensor networks. Data centric Routing protocol REEP uses Flooding to perform the first phase that has a lower efficiency. MLAF algorithm prevents the wasting of energy by considering new method and provides the possibility of data transmission with different priorities.

4.1. Energy performance comparison

Life time, remaining average energy and fairness are three important factors that should be taken into account in evaluating the performance of the proposed protocol. Figures 7 and 8 illustrate lifetime and remaining average energy of the network, respectively. The horizontal axis represents traffic load in kb/s and the vertical

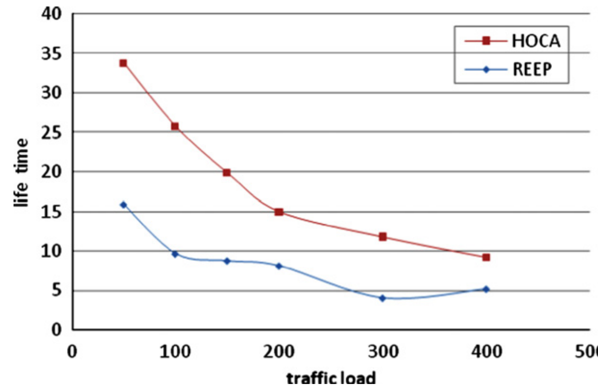

Fig. 7. Life time over traffic load.

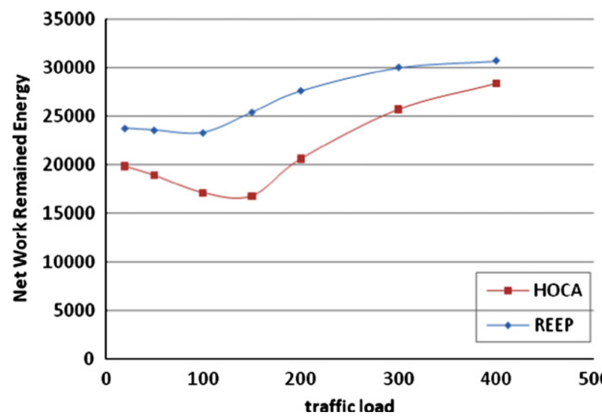

Fig. 8. Mean remained energy over traffic load.

axis represents lifetime per time unit. Network lifetime spans from the time the simulation is run until the first node dies.

As you can see in Fig. 7, the performance of HOCA is obvious in comparison with REEP from the view of traffic load, which are about 400 packets per time unit. For example, at a traffic load of 200 packets per time unit, HOCA increases lifetime in comparison to REEP by about 78 percentages. HOCA uses multiple paths to send data. This method ensures fair distribution of traffic at the destination, which increases network life time while the REEP uses one way traffic transmission. We can see in Fig. 7 that HOCA has a better performance than REEP in terms of network life time.

In Fig. 8, the mean of remaining node energy at the time of death for the first node has been calculated. According to the results in Fig. 8, the mean remaining energy in nodes for the HOCA protocol is less than REEP. The lower remaining energy is a result of higher energy consumption.

If the nodes consume most of their energy until the end of simulations, the protocol is considered more successful. Of course, energy consumption with more attention to increase percentage in life time (Fig. 7) is acceptable.

As we mentioned before, respecting fairness on energy consumption is one of the powerful point of HOCA in energy performance. If we can keep better balance in the energy consumption of nodes the lifetime of the network increases under the same conditions. According to Fig. 9, fairness parameter is more successful in HOCA rather than REEP one. Considered parameter has calculated with Eq. (6). Eq. (6) calculates the variance of normalized remaining energy of network nodes to average remained energy (*Ave*) of total network(In worst case half of the nodes get energy empty and half remain full, so if we normalize the equation we can achieve normalized fairness equation.). In Eq. (6), $Energy_i$ is node i remaining energy when simulations end.

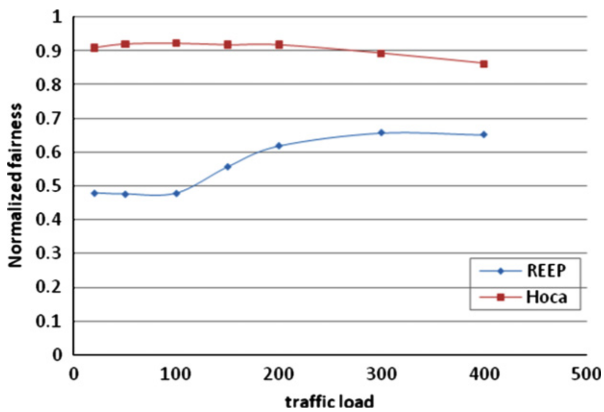

Fig. 9. Fairness over traffic load.

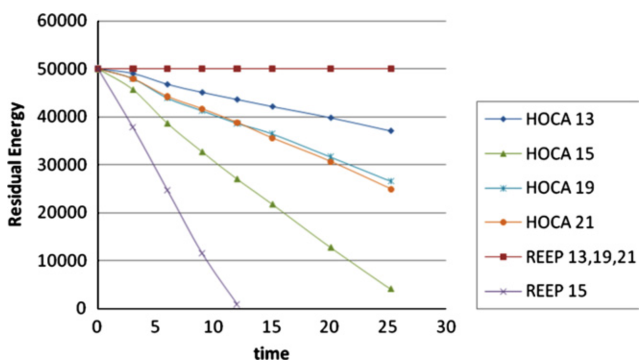

Fig. 10. Near-sink nodes energy over time.

As it is clear in Eq. (6) the more fairness parameter, the protocol is much success that it means remaining energy of nodes are closer with each other. If fairness parameter is equal one, the network has the best and the most fairness case (all nodes have the same remained energy), however when it is equal zero we have the most unfairness case of energy consumption.

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

$$\mathrm{Fairness} = 1 - \mathrm{DEV}/\mathrm{DEV}_{\mathrm{Worst}} \tag{6}$$

In WSNs, when data converge toward the sink, congestion is more likely to happen at sensors near sink which are likely to receive more data than they can forward. Every near-sink sensor node is a hotspot and so, its resources are more valuable. By providing fairness, network lifetime will be prolonged.

As it is clear in Fig. 10 the rate and speed of nodes residual energy for HOCA is closer with each other rather than REEP one. There are four nodes with number such as 13, 15, 19 and 21 on sink's neighborhood. The more increasing life time, the more successful network we have. In order to increase network life time, the traffic near the sink has to distribute among all nodes so that its life time prolongs. In REEP method, all the packets reach to sink by node 15 and other near sink nodes do not participate in traffic pass. So speed of this is less than other nodes and life time of network get worst. But in proposed method by fair traffic distribution between all nodes, the speed of node energy decreases going to be diminishing so that it causes network longer life time and fairness improvement on energy consumption. In Fig. 10, horizontal axis is the time and vertical axis shows nodes residual energy.

At the end, the total results of Figs. 7–10 show that HOCA energy performance is more efficient.

4.2. Packet loss comparison

Figures 11 and 12 show packet loss over traffic load and time, respectively. Figure 11 shows Aggregative number of dropped packets in the networks with respect to traffic load. As mentioned in Section 1 there are two different flows in network: sensitive and non-sensitive traffic. In Fig. 11 there are three flows: Sensitive HOCA, non-sensitive HOCA and one flow for REEP. Be careful REEP has no priority for different traffic, so regardless of packet priority does the same reaction.

As can been seen in Fig. 11 HOCA is more successful in packet delivery. Of course for traffic loads less than 60 due to lack of congestion, packet loss for both protocols are low and close. But in other points HOCA has been able to decrease packet loss in an appropriate level. Another point is the packet loss difference between the sensitive and non-sensitive flows. One of the needs of the sensitive traffic is to minimize packet loss. It can be seen in Fig. 11 that HOCA has achieved its goal.

In Fig. 12 aggregative packet loss rate over time with initial source rate 200 packets per second has been shown. According to Fig. 12 before the time 10 due to existing of control packets, the possibility of controlling source rate is a difficult process. Also hop by hop rate adjustment from congested node to source node will be accompanied with delay. With regard to the mentioned above after time 10 rate adjustment performs efficiently and as a result packet loss rate decreases that can be seen in Fig. 12.

In Fig. 13 HOCA protocol in addition to REEP is compared with 25% and 50% backpressure too. Back pressure refers to the back-pressure algorithms with 25% and 50%reduction percentages, respectively, in a sensor's data rate in response to a backpressure

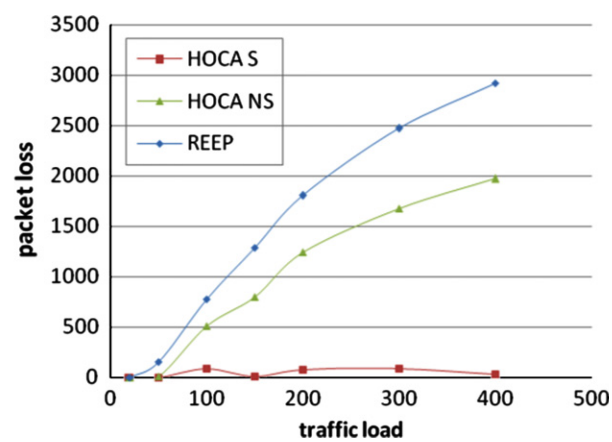

Fig. 11. Packet loss over traffic load (Kbps).

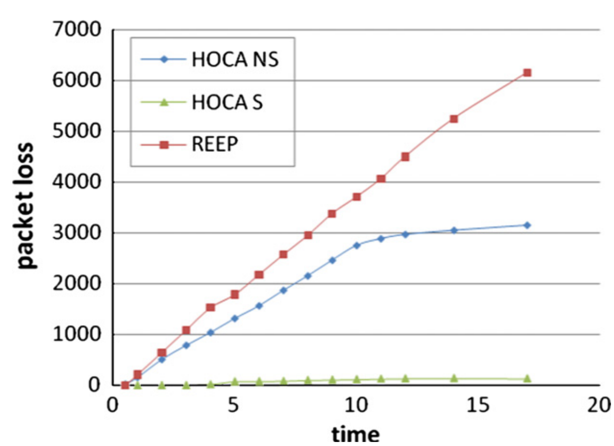

Fig. 12. Packet loss over time.

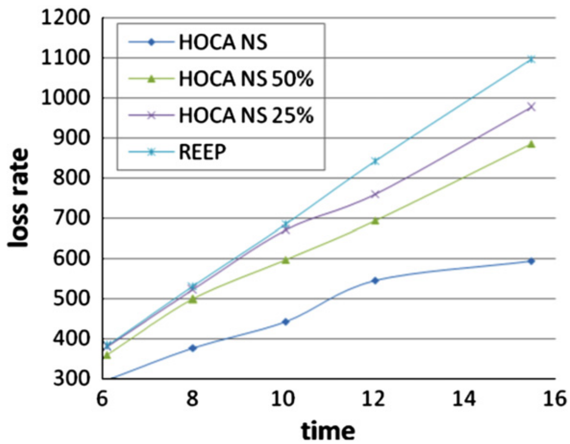

Fig. 13. Aggregative packet loss over time.

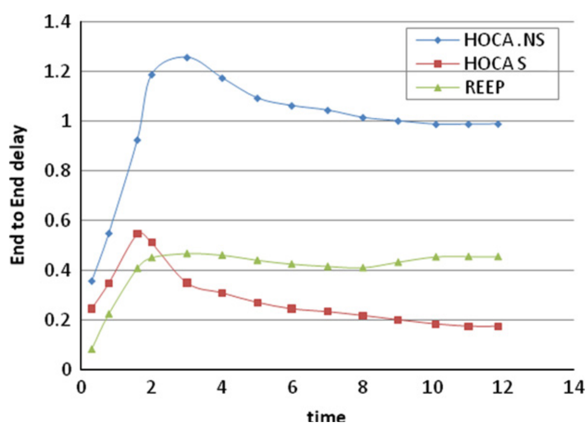

Fig. 14. End to end delay over time.

message. Horizontal axis is the time and vertical axis shows Aggregative packet loss. Initial source rate in the simulations is 100 packets per second.

We implement the backpressure algorithm for comparison purposes in our simulations. We have implemented backpressure here instead of proposed congestion control mechanism in HOCA. For example, "HOCA NS 25%" performs routing process with HOCA the same, but it uses 25% backpressure as congestion control mechanism. If a sensor x is congested, it periodically sends backpressure messages to its neighbors, which will reduce their forwarding rates by a certain percentage (25% or 50% in the simulations). If an upstream node is a data source, it reduces the new data generating rate by the same percentage.

As can be seen in Fig. 13 packet loss rate for HOCA is less than other methods. HOCA, due to using efficient congestion control and proper rate adjustment has less number of packet losses. According to Fig. 13 it can be observed that after HOCA, HOCA 50% and HOCA 25% have least number of packet losses. REEP does not have any congestion control procedure and therefore it has the largest number of packet loss.

4.3. End to end delay comparison

Another fundamental parameter which is considered in HOCA is the end to end delay. Delay is a parameter which is crucially important for the healthcare applications. With regard to the fact that REEP could not have priority for different traffic type, there exists only one priority for it. In Fig. 14 End to End delay for sensitive and non-sensitive traffic in HOCA and for REEP has been shown.

Due to the fact that it is not possible for REEP to prioritize different types of traffic, it supports only one type. Figure 13 presents the end-to-end delay in both sensitive and non-sensitive HOCA as well as REEP. End-to-end delay is the time taken for a packet to be transmitted from source to destination. Figure 14 indicates that the end-to-end delay for sensitive traffics is less than both insensitive and REEP traffics. Low end-to-end delay is expected for sensitive traffics considering the scheduler utilized for them. Simulations reveal that HOCA could achieve its objectives.

The HOCA protocol transmits more control packets in the first and second phases at the beginning of simulation results in more end-to-end delay in comparison with the total average. But with increase in time and end of control packets and beginning of congestion control procedure in companion with rate adjustment, HOCA traffic delay decreases.

Figure 15 illustrates the mean queue size over time. Mean queue size is a major metric in delay measurement. The more queue size, the more delay. AQM procedure is a fundamental factor to compute queue size that is discussed in Section 3.4.1. The reason behind the queue size rate being less in HOCA is utilizing multipath technique.

4.4. Source data transfer rate adjustment

Figure 16 shows the source data transfer rate adjustment which is an essential function for a data transfer protocol. As it is quite clear in Fig. 16, the rate adjustment is done in a hop-by-hop manner, starting from the congested node to the source, causing the source rate to be different in the end.

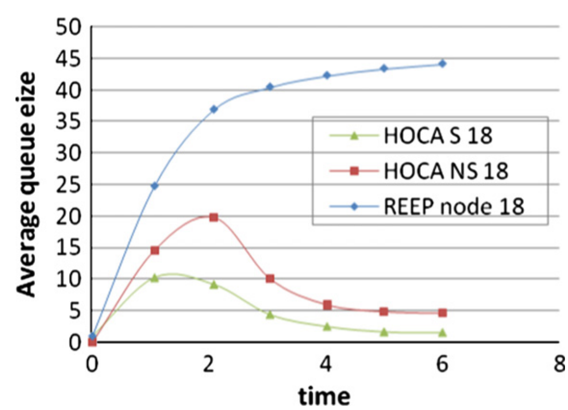

Fig. 15. Mean queue size over time.

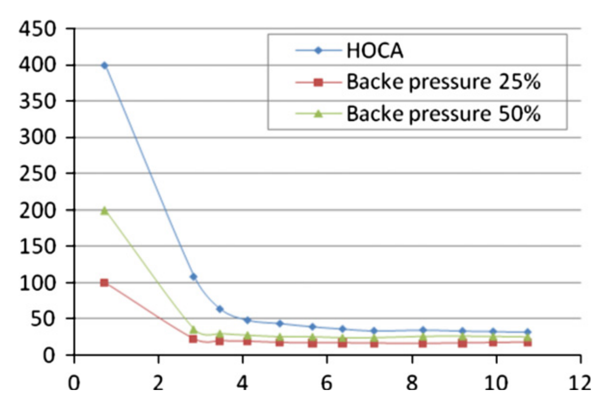

Fig. 16. Source data rate adjustment versus time.

Please cite this article as: Rezaee AA, et al. HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks. Journal of Network and Computer Applications (2013), http://dx.doi.org/10.1016/j.jnca.2013.02.014

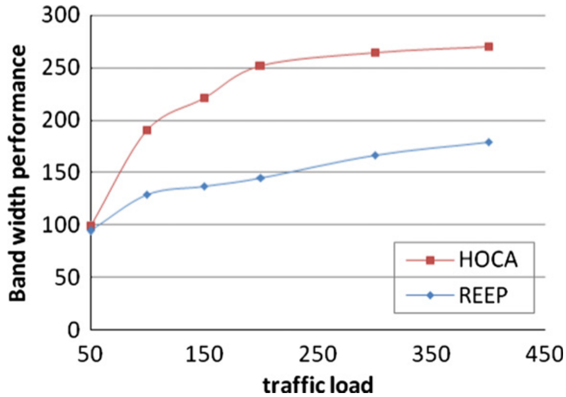

Fig. 17. Bandwidth performance over traffic load.

Since REEP does not have rate adjustment, its service rate remains constant. As mentioned before, REEP is used as a criterion to show the performance of the proposed protocol. HOCA 50% and HOCA 25% are also used for comparison. This figure shows the source sending rates for the algorithms. It is clear in Fig. 15, these algorithms have different sending rates. Figure 16 shows average sending rate versus time. As can be seen in this figure, our congestion control scheme is capable of automatically adapting the sensor's sending rate according to the network conditions and achieving better rate than other schemes. During the period of the congestion control, the total source rate is reduced, but at the time instant of around 5, congestion control comes into play (except for REEP), and the total source rate becomes stable after the mentioned time. Proposed HOCA achieves the largest total source rate due to its capability of allocating the exact bandwidth share to each passing flow in comparison to HOCA 50% and HOCA 25%.

4.5. Bandwidth performance

Bandwidth performance is one of the most important parameters in congestion management methods. In Fig. 16 we demonstrate the relation between bandwidth utilization and the total produced traffic. Here the bandwidth performance parameter is calculated based on the number of packets arrived at the sink node in a certain time unit. As is shown in Fig. 17, HOCA has much better bandwidth performance compared to REEP. This is mainly because of the large amount of lost packets in REEP. Also HOCA uses different paths to be able to send great amounts of traffic (multipath).

5. Conclusion

In this paper a congestion management protocol for the use in healthcare applications in wireless sensor networks has been presented. The best scenario for the proposed protocol can be patients bedridden in a special ward of a hospital or private clinic, particularly apoplectic or brain dead patients who are unable to move and so the sensor nodes are stationary in these applications. The proposed data driven congestion management algorithm consists of congestion avoidance and control components. The first phase of HOCA is designed to disseminate the demands of the sink while the other phases are respectively patients sign report (event report), the route establishment, data forwarding and congestion control. In data forwarding phase, HOCA encounters three kinds of traffic, namely, sensitive, non-sensitive, and control packet. Data packets are sent through multipath to avoid congestion. If congestion occurs, congestion control mechanism adjusts traffic source rates hop by hop. The proposed protocols takes into account parameters like end to end delay, energy consumption, lifetime of the network and fairness in energy consumption. Finally, using performed simulations, the performance of HOCA has been investigated. Simulation results show that the proposed protocol is more efficient than the backpressure and REEP schemes in terms of packet loss, energy efficiency, end to end delay and fairness.

Appendix A. Phase 2 (Event report)

Source nodes (patients):
DataSensitive = SenseSensitivePatientsSigns();
DataNonSensitive = SenseNonSensitivePatientsSigns();
Alarm_sink(DataSensitive, DataNonSensitive);
Intermeidate nodes:
ReceivePacket ();
If(SelfDistanceToSink < SenderDistanceToSink)
CreateRoutingTable();//fill up routing table records in intermediate nodes according to packets arrival time

Appendix B. Phase 3 (Route establishment phase)

Sink node:
//Send control packets from sink to source for S and NS data to make distinct routes
SendPacketSensitive();
SendPacketNonSensitive();
Intermediate nodes:
//choose first record in routing table for S making routing table and other records for NS routing table.
ReceivePacket();
If(PacketType = = Sensitive)
CreateSRoutingTable();//choose first record in Routing table for sensitive and create *S* routing table
If(PacketType = = non-Sensitive)
CreateNSRoutingTable();//choose other records in Routing table for non-sensitiveaccording to Eq. (1)
SendPacket();//send a packet for S and a packet for NS.

Appendix C. Phase 4 (Data forwarding phase)

//sensitive traffic = 20%, non-sensitive traffic = 80%, Congestion detection epoch time = 200 ms.
//CD: Congestion Degree
Variables:
CD: node congestion degree.
Threshold: congestion degree threshold
ql: max queue size.
q_i: virtual data queue for child I.
n: number of Child nodes.
Intermediate nodes:
ReceivePacket();
If (PacketType = sensitive)
InsertHighPriorityQueue();
SendSensitivePacket();//firstly send S packet in priority queue to next hop

Please cite this article as: Rezaee AA, et al. HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks. Journal of Network and Computer Applications (2013), http://dx.doi.org/10.1016/j.jnca.2013.02.014

```
CongestionDetection();
If (CD > Threshold )//congestion detected
        {θ^NH =  1;
        RateAdjusment (θ^NH);
        //Sending new Rates to child nodes ();
        SendRates();
      If (P_i is high )//calculate packet loss P_i with Eqs. (4)
  and (5)
        DeletePacket();
      else {
      //for records two and three other than first one
        x=Distance/HopCount;//record two
        y=Distance/HopCount;//record three
        P_n1=x/x+y;//probability of selecting record two for
  next hop
        P_n2= y/x+y;//probability of selecting record three for
  next hop
        SendNonSensitivePacket();
        }
}
```

Congestion detection

```
 If (PacketType=non-sensitive)
min=2/3 × ql/n;
if (q_i > min)
      CD=(3 × n × q_i − 2 × ql)/ql;
Else
      CD=0.0;
```

Receiving rate share packets from parent to child nodes

```
CongestionDetection();
θ^NH_new = α × θ^NH_old + (1 − α) × θ^NH));//α  < 1
RateAdjustment( Min(θ^NH,CD)//with formulas (2)
Node_service_rate=MaxRate*p1*θ1+MaxRate*p2* θ2;
Sending_ratei= Node_service_rate
θ^NH_old = θ^NH_new   //for the next time
If (CD > Threshold)
If (θ^NH >  (1 − CD))
         θ^NH = 1 − CD;
```

References

Abbas C, González R, et al. A proposal of a wireless sensor network routing protocol. Telecommunication Systems 2008;38:61–8.

Adjeroh, D, Yaghmaee, H.. A new priority based congestion control protocol for wireless multimedia sensor networks. In: International symposium on a world of wireless, mobile and multimedia networks, Newport beach, CA, USA, June 23–26, 2008; 2008. p. 1–8.

Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. Ad Hoc Networks 2005;3:325–49.

Akyildiz IF, Su Y, Sankarasubramaniam W, Çayirci E. Wireless sensor networks: a survey. Computer Networks Journal 2002;38:393–422.

Akyildiz, IF, Akan, OB, Sankarasubramaniam, Y.. ESRT: Event-to-sink reliable transport in wireless sensor networks. In: Proceedings of ACMMobihoc'03, June 1–3, 2003, Annapolis, Maryland; 2003.

Armijo, CRK, Belka, S, Benhabib, M, Leland, ES, Pering, T, Waterbury, A, et al.. Wireless sensor networks for home health care. In: Proceedings 21st international conference on advanced information networking applications workshops; 2007.

Bajcsy, R, Ee, C-T.. CCF: Congestion control and fairness for many-to-one routing in sensor networks. In: Proceedings of ACM Sensys'04, November 3–5, 2004, Baltimore, Maryland, USA; 2004.

Balakrishnan, H, Hull, B, Jamieson, K.. Mitigating congestion in wireless sensor networks. In: Proceedings of the ACM Sensys '04, Baltimore, MD, November 2004; 2004.

Borden, M, Firoiu, V.. A study of active queue management for congestion control. IEEE INFOCOM; 2000.

Cao, Q, Doan, T, Fang, L, He, Z., Kiran Lin, SR, Stankovic, JA, et al.. Wireless sensor networks for in-home healthcare: Potential and challenges. In *Proceedings of the high confidence medical device software systems (HCMDSS) workshop*, June 2005; 2005.

Cao, Q, Doan, T, Fang, L, He, Z, Lin, S, Selavo, L, et al.. Alarm-net: An assisted living centered and testbed-oriented information system based on residential wireless sensor network. In: Proceedings of the transdisciplinary conference distributed diagnosis home healthcare; 2006.

Daneshmand M, Hu Y, Li B, Sohraby K, Wang C. Upstream congestion control in wireless sensor networks through cross-layer optimization. In IEEE Journal on Selected Areas in Communications 2007;25(4):786–95.

Farzaneh N, Monsefi R, et al. QTCP: a novel congestion control protocol with AQM support for IP-based networks. Telecommunication Systems 2011:1–16.

Fischer, M, Masi, D, Shortle, J. Approximating low latency queuing buffer latency. In: Fourth advanced international conference on telecommunications; 2008.

Huang R, Fang Y, Li S, Yin X, Zhou X. A fairness-aware congestion control scheme in wireless sensor networks. In IEEE Transactions on Vehicular Technology 2009;58(9 (November)).

Jang J. A study on a sequenced directed diffusion algorithm for sensor networks. In: Proceedings of 9th ICACT, Gangwon-Do, Korea, February 2007; 2007. p. 679–683.

Misra S, Rashvand HF, Zabin F, Woungang I. REEP: data-centric, energy-efficient and reliable routing protocol for wireless sensor networks. In IET Communications 2008 2008;2(8):995–1008.

Misra S. Obaidat, M.S. Fellow, Tiwari V. LACAS: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. IEEE Journal on Selected Areas in Communications 2009;27(4(May)).

Mohajerzadeh, AH, Monsefi, R, Yaghmaee, MH. (. MLAF: A QoS based data dissemination protocol for wireless multimedia sensor networks. *IEEE*; 2010.

Shu L, Zhang Y, et al. TPGF: geographic routing in wireless multimedia sensor networks. Telecommunication Systems 2010;44(1):79–95.

Sohraby, K, Wang, C.. A survey of transport protocols for wireless sensor networks. In: IEEE network magazine, May/June 2006; 2006. p. 34–40.

Tao LQ, Yu FQ. ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks. Transactions on Consumer Electronics 2010;56(3).

Wan, C–Y, Eisenman, SB, Campbell, AT.. CODA: Congestion detection and avoidance in sensor networks. In: Proceedings of ACMSensys'03, November 5–7, 2003, Los Angeles, California, USA; 2003.

〈www.OPNET.com〉.

Yaghmaee MH, Adjeroh DA. Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks. Computer Networks 2009, . http://dx.doi.org/10.1016/j.comnet.2009.02.011.