A New Priority Based Congestion Control Protocol for Wireless Multimedia Sensor Networks

Mohammad Hossein Yaghmaee[*#] and Donald Adjeroh[#]
[*]Department of Computer Engineering, Ferdowsi University, Mashad
[#]Lane Department of Computer Science and Electrical Engineering,
West Virginia University, Morgantown, WV 26506
e-mails:{hyaghmae@ferdowsi.um.ac.ir, don@csee.wvu.edu}

Abstract

New applications made possible by the rapid improvements and miniaturization in hardware has motivated recent developments in Wireless Multimedia Sensor Networks (WMSNs). As multimedia applications produce high volumes of data which require high transmission rates, multimedia traffic is usually high speed. This may cause congestion in the sensor nodes, leading to impairments in the quality of service (QoS) of multimedia applications. Thus, to meet the QoS requirements of multimedia applications, a reliable and fair transport protocol is mandatory. An important function of the transport layer in WMSNs is congestion control. In this paper, we present a new Queue based Congestion Control Protocol with Priority Support (QCCP-PS), using the queue length as an indication of congestion degree. The rate assignment to each traffic source is based on its priority index as well as its current congestion degree. Simulation results show that the proposed QCCP-PS protocol can detect congestion better than previous mechanisms. Furthermore it has a good achieved priority close to the ideal and near-zero packet loss probability, which make it an efficient congestion control protocol for multimedia traffic in WMSNs. As congestion wastes the scarce energy due to a large number of retransmissions and packet drops, the proposed QCCP-PS protocol can save energy at each node, given the reduced number of retransmissions and packet losses.

1. Introduction

Wireless Multimedia Sensor Networks (WMSNs) [1] is a set of sensor nodes, whereby the nodes are equipped with multimedia devices such as cameras, and microphones. Thus a WMSN will have the capability to transmit multimedia data, such as still pictures, stream video, voice, animal sounds, and monitoring data. One important requirement of applications in WMSNs is low delay bounds. Furthermore, some applications of WMSNs need relative resilience to losses. In WMSNs there exist two different types of multimedia content, snapshot and streaming data. Snapshot-type multimedia data contain event triggered observations obtained in a short time period. Streaming multimedia content is generated over longer time periods and requires sustained information delivery. As described in [1], WMSNs can support different types of multimedia traffic classes. Similar to Wireless Sensor Networks (WSNs)[2], applications of WMSNs share different characteristics such as: resource constraints, unbalanced mixture traffic, data redundancy, network dynamics and energy balance. There are many different resource constraints in WMSNs involving energy, bandwidth, memory, buffer size and processing capability. Given the physically small nature of the sensors, and that multimedia applications typically produce huge volumes of data requiring high transmission rates and extensive processing, a fundamental concern in WMSNs is the issue of power consumption. Thus, developing protocols, algorithms and architectures to maximize the network lifetime while satisfying the quality of service requirements of the applications represents a critical problem. In most WSN and WMSN applications, traffic mainly flows from a large number of sensor nodes to a base station (sink) node. Therefore, to meet the quality of service requirements and to use the network resources in a fair and efficient manner, this characteristic of WMSNs becomes a concern, and must be considered. Furthermore, given the relatively high redundancy in the sensor data, techniques such as data compression, data fusion and aggregation becomes

978-1-4244-2100-8/08/$25.00 ©2008 IEEE

important in maintaining robustness, while decreasing the amount of data. Another important characteristic of WSNs and WMSNs is the dynamic changes in topology and the unreliable nature of wireless networks. This is primarily due to changes in node mobility and/or wireless channel failure. To meet quality of service requirements, these natural characteristics of this kind of networks must be considered in designing the required protocols.

To support multimedia communication in WMSNs having a reliable transport mechanism is important. In traditional communication networks, transport layer is responsible for bridging the application and network layers using multiplexing and demultiplexing. It is also charged with providing end-to-end reliable data delivery and with performing congestion control by regulating the amount of traffic injected into the network. WMSNs must support different applications which may require different levels of reliability as well as different congestion control approaches. In addition to the challenges for reliable data transport in WSN, there exist additional challenges due to the unique requirements of multimedia transport such as bounded delay and delay variation, minimum bandwidth demand, smooth traffic variation for multimedia streaming, and error control according to the specific requirements of the multimedia application. As argued in [3], the traditional TCP/UDP transport protocols cannot be directly implemented for WSN and WMSN. Therefore, it is important to develop a reliable transport protocol for WMSNs to ensure that the often differing QoS requirements of various applications can be met.

Congestion control is another important issue that should be considered in transport protocols. Congestion is an essential problem in wireless sensor networks. It not only wastes the scarce energy due to a large number of retransmissions and packet drops, but also hampers the event detection reliability. Congestion in WSNs and WMSNs has a direct impact on energy efficiency and application QoS. Two types of congestion could occur in sensor networks [4]. The first type is node-level congestion that is caused by buffer overflow in the node and can result in packet loss, and increased queuing delay. Not only can packet loss degrade reliability and application QoS, but it can also waste the limited node energy and degrade link utilization. In each sensor node, when the packet-arrival rate exceeds the packet-service rate, buffer overflow may occur. This is more likely to occur at sensor nodes close to the sink, as they usually carry more combined upstream traffic. The second type is link-level congestion that is related to the wireless channels which are shared by several nodes using protocols, such as CSMA/CD (carrier sense, multiple access with collision detection). In this case, collisions could occur when multiple active sensor nodes try to seize the channel at the same time. To avoid the negative aspects of congestion in WMSNs, congestion must be effectively controlled. Each congestion control solution consists of three important parts: congestion detection, congestion notification, and rate adjustment. In traditional TCP protocol, congestion is detected at the end nodes based on a timeout or redundant acknowledgments. In general, link-by-link congestion detection in sensor networks has better performance than traditional end-to-end congestion detection using time out or duplicate acknowledgment. Thus, in sensor networks, proactive methods are used, based on some form of congestion indicator. Different congestion indicators have been proposed, such as, queue length [5, 6], packet service time [4], or the ratio of packet service time to packet inter-arrival time at the intermediate nodes [7]. After detecting congestion, to prevent the negative aspects of congestion in the networks, the transport protocol needs to propagate congestion information from the congested node to the upstream sensor nodes or the source nodes that contribute to congestion. This can be done explicitly by sending a special control message to the other sensors, or implicitly using piggybacking technique in data packets. When a node receives a congestion notification message, it should adjust its transmission rate using a rate control techniques such as Additive Increase Multiplicative Decrease (AIMD).

The reminder of this paper is organized as follows. In section 2, we present a brief survey on the previous studies in congestion control in sensor networks. In section 3, we explain the proposed QCCP-PS (Queue based Congestion Control Protocol with Priority Support). Section 4 gives a numerical example. In section 5, using computer simulation, we evaluate the performance of the proposed congestion controller. Section 6 concludes the paper.

2. Related works

Different congestion control techniques have been proposed for wireless sensor networks. The congestion control mechanisms all have the same basic objective: they all try to detect congestion, notify the other nodes of the congestion status, and reduce the congestion and/or its impact using rate adjustment algorithms. In [8], *CODA,* an energy efficient congestion control scheme for sensor networks was proposed. *CODA*

(COngestion Detection and Avoidance) comprises three mechanisms: (i) receiver-based congestion detection; (ii) open-loop hop-by-hop backpressure; and (iii) closed-loop multi-source regulation. CODA detects congestion based on queue length as well as wireless channel load at intermediate nodes. Furthermore it uses explicit congestion notification approach and also an AIMD rate adjustment technique.

Congestion Control and Fairness (CCF) was proposed in [4] as a distributed and scalable algorithm that eliminates congestion within a sensor network and ensures the fair delivery of packets to a sink node. CCF exists in the transport layer and is designed to work with any MAC protocol in the data-link layer. In the CCF algorithm, each node measures the average rate *r* at which packets can be sent from the node, divide the rate *r* among the number of children nodes, adjust the rate if queues are overflowing or about to overflow and propagate the rate downstream. CCF uses packet service time to deduce the available service rate. Congestion information is implicitly reported. It controls congestion in a hop-by-hop manner and each node uses exact rate adjustment based on its available service rate and child node number. It can be shown that CCF guarantees simple fairness. As shown in [7], CCF has two major problems. The rate adjustment in CCF relies only on packet service time which could lead to low utilization when some sensor nodes do not have enough traffic or there is a significant packet error rate. Furthermore, it cannot effectively allocate the remaining capacity and as it uses work-conservation scheduling algorithm, it has a low throughput in the case that some nodes do not have any packet to send.

In [9], an Adaptive Rate Control (ARC) mechanism was proposed which is most effective in achieving the goal of fairness, while being energy efficient for both low and high duty cycle of network traffic. The ARC does not have any congestion detection or notification mechanisms. Each intermediate node increases its sending rate by a constant α if it overhears successful packet forwarding by its parent node. Otherwise, the intermediate node multiplies its sending rate by a factor β. Priority based Congestion Control Protocol (PCCP) was proposed in[7]. PPCP is an upstream congestion control protocol for WSNs which measures the congestion degree as the ratio of packet inter-arrival time to the packet service time. Based on the introduced congestion degree and node priority index, PCCP utilizes a cross-layer optimization and imposes a hop-by-hop approach to control congestion. It has also been shown that PCCP achieves efficient congestion control and flexible weighted fairness for both single-path and multi-path routing. In [10], a reliable transport protocol suitable for reliable data applications called PSFQ (Pump Slowly, Fetch Quickly) has been proposed. It takes a different approach and supports a simple, robust and scalable transport that is customizable to meet the needs of different reliable data applications. We mention that PSFQ is only a reliability guarantee protocol and not for congestion control. The combination of CODA and PSFQ may achieve both congestion control and reliability. In [11-14], different protocols for congestion control and upstream and downstream reliability in transport protocol were proposed.

3. Proposed congestion control protocol

In this section we describe our proposed congestion control protocol for wireless multimedia sensor networks. The proposed protocol is called QCCP-PS (Queue based Congestion Control Protocol with Priority Support). Our approach is motivated by the apparent limitations of existing popular schemes, such as the PCCP. The simulation results confirm that the PCCP performs very poorly in providing relative priority in the case of random service time. Based on line 6 and 9 of PCCP algorithm given in [7], it can be seen that in the case of low congestion, the PCCP will increase the scheduling rate and source rate of all traffic sources without paying any attention to their priority index. In the case of high congestion, PCCP will decrease the sending rate of all traffic sources based on their priority index. The proposed QCCP-PS protocol solves this problem by a proper adjustment of the rate at each node. In the QCCP-PS, the sending rate of each traffic source is increased or decreased depending on its congestion condition and its priority index. Figure 1, shows the architecture of QCCP-PS. Similar to the other congestion control protocols, QCCP-PS consists of three parts namely, Congestion Detection Unit (CDU), Congestion Notification Unit (CNU), and Rate Adjustment Unit (RAU). The CDU is responsible for detecting any congestion in advance. The CDU uses the queue length as the congestion indicator. The output of CDU is a congestion index, which is a number between 0 and 1. CDU uses a similar congestion indicator strategy used in the RED active queue management algorithm [15]. For this purpose, two different fixed thresholds $\max_{th}$ and $\min_{th}$ are defined. When the queue length (q) is less than $\min_{th}$, congestion index is very low and the

source node could increase its rate. On the other hand, when queue length is greater than $\max_{th}$, congestion index is high and the traffic source should decrease its rate to avoid any packet loss. In the case that queue length is between $\max_{th}$ and $\min_{th}$ the congestion index is related to queue length linearly. In each predefined time interval T, each parent node calculates the sending rate of all its child traffic sources as well as its local traffic source. As each sensor node may have different priorities since sensor nodes might be installed with different kinds of sensors in an environment, the upstream node also considers the priority of each of its child nodes in calculating the rate of the child nodes. Based on the current congestion index and the source traffic priority, the RAU calculates the new rate of each child traffic sources as well as its local traffic source. The new rate is sent to the CNU unit which is responsible for notifying all the child nodes of the new rate. To decrease energy consumption, CNU uses an implicit congestion notification by adding the new rate of each child node to the sending data of each sensor node. When a node receives a congestion notification message from its upstream node, the node is expected to adjust its traffic rate accordingly.

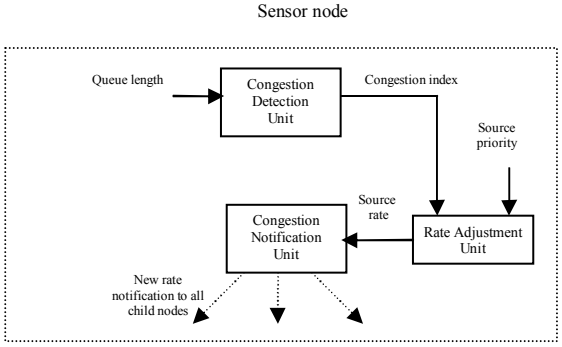

Figure 1. The structure of proposed protocol

In the proposed QCCP-PS protocol, in each sensor node we use a separate queue to store input packets from each child node. The sent traffic from each child node is buffered in a separate queue. Furthermore, as each sensor node may have some local source traffic to be sent to the sink node, an additional queue is also considered for local traffic of the sensor node. So, if an intermediate node i has N_i child nodes, then it needs $N_i + 1$ queues to store packets. Figure 2, shows the network model used in each sensor node. All queues are implemented as First-In First-Out (FIFO) queues.

3.1. Computing the node rate

Let $T_s(i)$ denote the service time of the current packet in node i. Using the exponential weighted sum, the average service time $\overline{T}_s(i)$ is calculated as:

$$\overline{T}_s(i) = (1-\alpha)\overline{T}_s(i) + \alpha.T_s(i) \quad (1)$$

where α is a constant coefficient. Note that the average service time $\overline{T}_s(i)$ is the time taken to successfully transmit a data packet over the MAC layer. It is measured starting from the time when the network layer first sends the packet to the MAC layer to the time when the MAC layer notifies the network layer that the packet has been transmitted. After calculating the average service time $\overline{T}_s(i)$, the node rate, r_i, is obtained as follows:

$$r_i = \frac{1}{\overline{T}_s(i)} \quad (2)$$

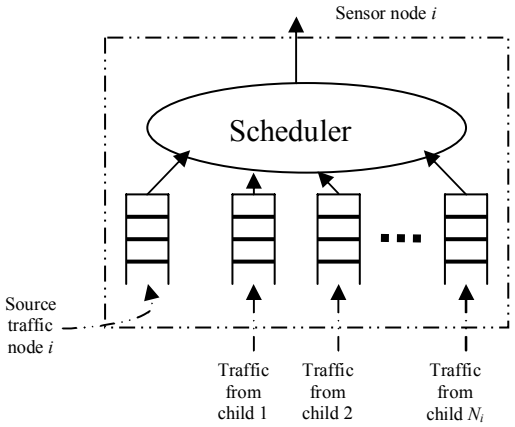

Figure 2. Per child queuing in sensor node *i*

3.2. Congestion index

At each sensor node i, we use current queue size to obtain the congestion index, $I_X(i)$. Let $q^k(i)$ denote the current queue size of the *k-th* queue in node i. We define two thresholds $\min^k{}_{th}(i)$ and $\max^k{}_{th}(i)$ which are used for calculation of the congestion index. The congestion index, $I_X^k(i)$, is calculated using the buffer occupancy as follows:

$$I_X^k(i)=\begin{cases}\varepsilon & \text{if} \quad q^k(i)\le \min^k{}_{th}(i) \\ \dfrac{\max_p(q^k(i)-\min^k{}_{th}(i))}{\max^k{}_{th}(i)-\min^k{}_{th}(i)}+\varepsilon & \text{if} \quad \min^k{}_{th}(i)\le q^k(i)\le \max^k{}_{th}(i) \\ \max_p+\varepsilon & \text{if} \quad q^k(i)\ge \max^k{}_{th}(i)\end{cases} \tag{4}$$

where ε and $\max_p$ are small numbers less than 1. Notice that by using the above definition for the congestion index, we will always have $I_X^k(i)\le \max_p+\varepsilon$.

3.3. Rate assignment

Using the above congestion index, each sensor node *i* would be able to assign the proper rate to each of its childe nodes as well as to its local traffic source. Suppose that sensor node *i* has N_i child nodes. So it has N_i+1 queues (N_i queues for its child nodes and one queue for its local traffic source). For each queue *k* in sensor node *i*, the congestion index $I_X^k(i),\ k=1,2,...,N_i+1$ is calculated using equation (4). Then $\bar{I}_X^k(i)$ is obtained as:

$$\bar{I}_X^k(i)=\frac{\sum_{j=1}^{N_i+1} I_X^j(i)-I_X^k(i)}{N_i\cdot\sum_{j=1}^{N_i+1} I_X^j(i)} \tag{5}$$

Now, suppose each node *i* has a different priority. Let $SP(i)$ denote the source priority at sensor node *i*. We define the total priority, $Total\_P(i)$, as the sum of priorities of all nodes in the subtree rooted at node *i*. Let $C(i)$ be the set of node *i*'s child nodes. Then the total priority, $TP(i)$, is calculated as:

$$TP(i)=\sum_{j\in C(i)} TP(j)+SP(i) \tag{6}$$

If a node doesn't have any child, then its total priority is equal to its source priority. In each queue *k* in node *i*, the weight w_i^k and the input rate r_i^k are calculated as:

$$w_i^k=\frac{p_i^k\bar{I}_X^k(i)}{\sum_{j=1}^{N_i+1} p_i^j\bar{I}_X^j(i)}$$
$$r_i^k=w_i^k.r_i \tag{7}$$

where p_i^k is obtained as:

$$p_i^1=\frac{SP(i)}{TP(i)}\ ,\ p_i^k=\frac{TP(k)}{TP(i)},k=2,3,...,N_i+1 \tag{8}$$

Note that the rate r_i^k is calculated for all active sources. When a sensor node is not active, then its congestion index is set to infinity. In this case the allocated rate to all inactive nodes will be equal to zero. So in each sensor node *i*, the output rate r_i is shared only between active nodes.

3.4. Operation of scheduler

At each node *i*, there is a scheduler that is responsible for servicing each queue based on its weight. To provide fairness based on the priority of each sensor node, the scheduler should service each queue according to its weight. According to Equations 7 and 8 which represent, w_i^k, the weight of the *k-th* queue in node *i*, it is clear that summation of all weights in each node *i* should be equal to 1, (i.e. $\sum_{k=1}^{N_i+1} w_i^k=1$). We use a WFQ scheduler which is able to service each queue based on its weight.

4. Numerical example

In this section we present a numerical example on the operation of QCCP-PS. Consider a sensor network consisting of ten nodes. These ten nodes are connected to one sink node using the network topology shown in Figure 3. Note that this network is a single path sensor network. Suppose the maximum output rate of node 1 is normalized to 1 ($r_1=1$).

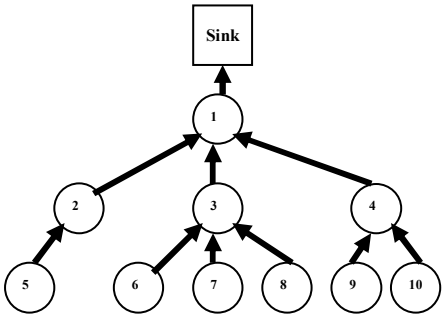

Figure 3. A single path network topology

Suppose also that each sensor node has different priority which is equal to its node number

($SP(i)=i, i=1,2,....,10$). So node 10, has the most priority and node 1 has the least priority. For simplicity, we suppose that at time $t=t_1$, the congestion degree in all nodes is the same. This means that: $\bar{I}_X^k(i)=\frac{1}{N_i+1}$, $i=1,2,..10$, $k=1,..,N_i+1$. In this case, the total priority of each sensor node is obtained as follows:

$$TP(10)=\sum_{j\in C(10)}TP(j)+SP(10)=10$$
$$TP(9)=\sum_{j\in C(9)}TP(j)+SP(9)=9$$
$$TP(8)=\sum_{j\in C(8)}TP(j)+SP(8)=8$$
$$TP(7)=\sum_{j\in C(7)}TP(j)+SP(7)=7$$
$$TP(6)=\sum_{j\in C(6)}TP(j)+SP(6)=6$$
$$TP(5)=\sum_{j\in C(5)}TP(j)+SP(5)=5$$
$$TP(4)=\sum_{j\in C(4)}TP(j)+SP(4)=19+4=23$$
$$TP(3)=\sum_{j\in C(3)}TP(j)+SP(3)=21+3=24$$
$$TP(2)=\sum_{j\in C(2)}TP(j)+SP(2)=5+2=7$$
$$TP(1)=\sum_{j\in C(1)}TP(j)+SP(1)=54+1=55$$

To obtain the local source traffic rate of each node, the following calculation is used:

At node 1:

$$p_1^1=\frac{SP(1)}{TP(1)}=\frac{1}{55}, p_1^2=\frac{TP(2)}{TP(1)}=\frac{7}{55},$$
$$p_1^1=\frac{TP(3)}{TP(1)}=\frac{24}{55}, p_1^4=\frac{TP(4)}{TP(1)}=\frac{23}{55}$$
$$w_1^1=\frac{p_1^{\,1}\bar{I}_X^1(1)}{\sum_{j=1}^{3+1}p_1^{\,j}\bar{I}_X^j(1)}=p_1^1=\frac{1}{55}, w_1^4=\frac{p_1^{\,2}\bar{I}_X^2(1)}{\sum_{j=1}^{3+1}p_1^{\,j}\bar{I}_X^j(1)}=p_1^2=\frac{7}{55},$$
$$w_1^3=\frac{p_1^{\,3}\bar{I}_X^3(1)}{\sum_{j=1}^{3+1}p_1^{\,j}\bar{I}_X^j(1)}=p_1^3=\frac{24}{55}, w_1^4=\frac{p_1^{\,4}\bar{I}_X^4(1)}{\sum_{j=1}^{3+1}p_1^{\,j}\bar{I}_X^j(1)}=p_1^4=\frac{23}{55}$$
$$r_1^1=w_1^1.r_1=\frac{1}{55}, r_1^2=w_1^2.r_1=\frac{7}{55},$$
$$r_1^3=w_1^3.r_1=\frac{24}{55}, r_1^4=w_1^4.r_1=\frac{23}{55}$$

At node 2:

$$p_2^1=\frac{SP(2)}{TP(2)}=\frac{2}{7},$$
$$p_2^2=\frac{TP(5)}{TP(2)}=\frac{5}{7}$$
$$w_2^1=p_2^1=\frac{2}{7}, w_2^2=p_2^2=\frac{5}{7}$$
$$r_2^1=w_2^1.r_2=\frac{2}{7}.\frac{7}{55}=\frac{2}{55}, r_2^2=w_2^2.r_2=\frac{5}{7}.\frac{7}{55}=\frac{5}{55}$$

At node 3:

$$p_3^1=\frac{SP(3)}{TP(3)}=\frac{3}{24}, p_3^2=\frac{TP(6)}{TP(3)}=\frac{6}{24},$$
$$p_3^3=\frac{TP(7)}{TP(3)}=\frac{7}{24}, p_3^4=\frac{TP(8)}{TP(3)}=\frac{8}{24}$$
$$w_3^1=p_3^1=\frac{3}{24}, w_3^2=p_3^2=\frac{6}{24}, w_3^3=p_3^3=\frac{7}{24}, w_3^4=p_3^4=\frac{8}{24}$$
$$r_3^1=w_3^1.r_3=\frac{3}{24}.\frac{24}{55}=\frac{3}{55}, r_3^2=w_3^2.r_3=\frac{6}{24}.\frac{24}{55}=\frac{6}{55},$$
$$r_3^3=w_3^3.r_3=\frac{7}{24}.\frac{24}{55}=\frac{7}{55}, r_3^4=w_3^4.r_3=\frac{8}{24}.\frac{24}{55}=\frac{8}{55}$$

At node 4:

$$p_4^1=\frac{SP(4)}{TP(4)}=\frac{4}{23}, p_4^2=\frac{TP(9)}{TP(4)}=\frac{9}{23},$$
$$p_4^3=\frac{TP(10)}{TP(4)}=\frac{10}{23}$$
$$w_4^1=p_4^1=\frac{4}{23}, w_4^2=p_4^2=\frac{9}{23}, w_4^3=p_4^3=\frac{10}{23}$$
$$r_4^1=w_4^1.r_4=\frac{4}{23}.\frac{23}{55}=\frac{4}{55}, r_4^2=w_4^2.r_4=\frac{9}{23}.\frac{23}{55}=\frac{9}{55},$$
$$r_4^3=w_4^3.r_4=\frac{10}{23}.\frac{23}{55}=\frac{10}{55}$$

Thus, the source traffic rate of all nodes ($r_i^s, i=1,2,...,10$) is obtained as below:

$$r_{10}^s=\frac{10}{55}, r_9^s=\frac{9}{55}, r_8^s=\frac{8}{55}, r_7^s=\frac{7}{55}, r_6^s=\frac{6}{55}, r_5^s=\frac{5}{55}, r_4^s=\frac{4}{55},$$
$$r_3^s=\frac{3}{55}, r_2^s=\frac{2}{55}, r_1^s=\frac{1}{55}$$

In this scenario, the congestion index in all nodes are the same, the source traffic rate of each node is relative to its source priority. So node 10 which has highest priority, has the largest rate ($\frac{10}{55}$), and node 1 which has lowest priority has the least rate ($\frac{1}{55}$).

5. Simulation results

In this section, we evaluate the performance of the QCCP-PS protocol and compare it with the CCF and PCCP algorithms. For this purpose we developed a discrete event simulation software using C++ language in a UNIX operating system environment. The network topology used in the simulation is a simple single path WSN network similar to Figure 3. We evaluate the performance of all algorithms under different

scenarios. The simulation time is set to100 sec. Given space constraints, we show results for only a few scenarios.

In the first simulation trial, similar to PCCP, we consider a fixed service time for all sensor nodes. All nodes have the same priority. Figure 4 shows the total sum of normalized throughput. It can be seen that all algorithms have the same throughput. The average throughput of CCF, PCCP and QCCP-PS are equal to 0.99, 0.96 and 0.98, respectively.

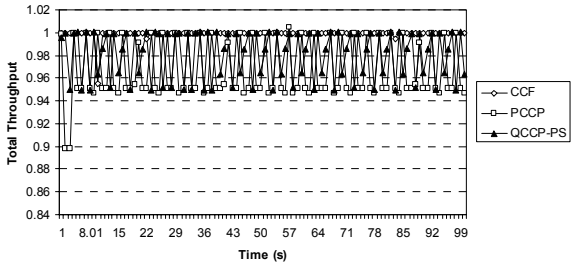

Figure 4. Total throughput (fixed service time)

In the second trial, the sensor node 2, is off in time interval [20 sec,70 sec]. As shown in Figure 5, since CCF cannot effectively allocate the remaining capacity and use a work-conservation scheduling algorithm, it has a lower throughput in the interval [20 sec, 70 sec]. Both PCCP and QCCP-PS protocols use the network capacity better and thus maintain high throughput during the interval. The average throughput for CCF, PCCP and QCCP-PS were 0.95, 0.97 and 0.98, respectively.

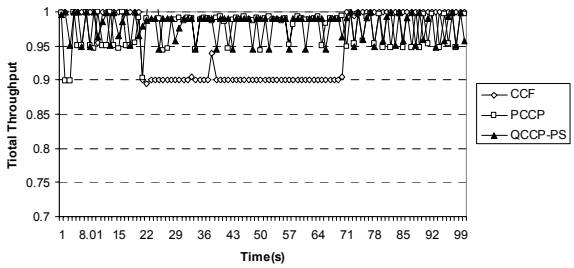

Figure 5. Total throughput (node 7 is off in the interval [20 sec, 70 sec])

To evaluate the performance of PCCP and QCCP-PS protocols in real conditions, we consider a random service time. In this case sensor nodes have an exponential service time with mean of 0.005 sec. Figures 6 and 7 show results for different buffer sizes, the total throughput, achieved priority index and probability of packet loss, for the case of different priorities. Note that as the PCCP uses a single buffer for all transient traffic, we set the buffer size of PCCP N_i times that of the QCCP-PS protocol, where N_i is the number of child nodes in sensor node *i*.

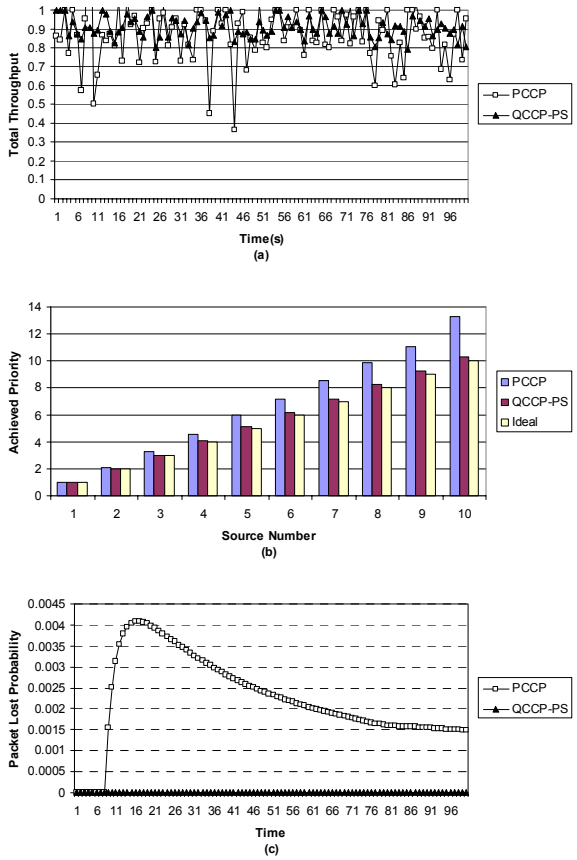

Figure 6. (a) Total throughput; (b) achieved priority; (c) packet loss probability (buffer size = 100 packets)

Figures 6 and 7 show that QCCP-PS has better performance than PCCP. Figures 6(a), 7(a), confirm that the total throughput of the QCCP-PS protocol is better than PCCP. When buffer size is set to 100 packets, the average throughput of PCCP and QCCP-PS protocols is equal to 0.87 and 0.91, respectively. In the second case, when buffer size is equal to 10 packets, the average throughput of PCCP and QCCP-PS protocols is equal to 0.84 and 0.91, respectively. Based on the results in Figures 6(b) and 7(b), it is clear that the QCCP-PS protocol provides a better priority index than the PCCP. For example, when buffer size is equal to 100 packets (Figure 6(b)), for source node 10, the achieved priority of PCCP and the QCCP-PS are equal to 13 and 10.1, respectively. So it is clear that QCCP-PS can provide a better priority index in comparison to PCCP. QCCP-PS adjusts the traffic rate of each node based on its degree of congestion, and thus can avoid unnecessary packet loss. This results in

a high throughput and better achieved priority index. From Figures 6(c) and 7(c), when buffer size is equal to 100 packets and 10 packets respectively, the packet loss probability of PCCP in the steady state is equal to 0.0015 and 0.015, respectively. For QCCP-PS protocol, we have a near-zero packet loss. Thus, its operation is good for multimedia traffic.

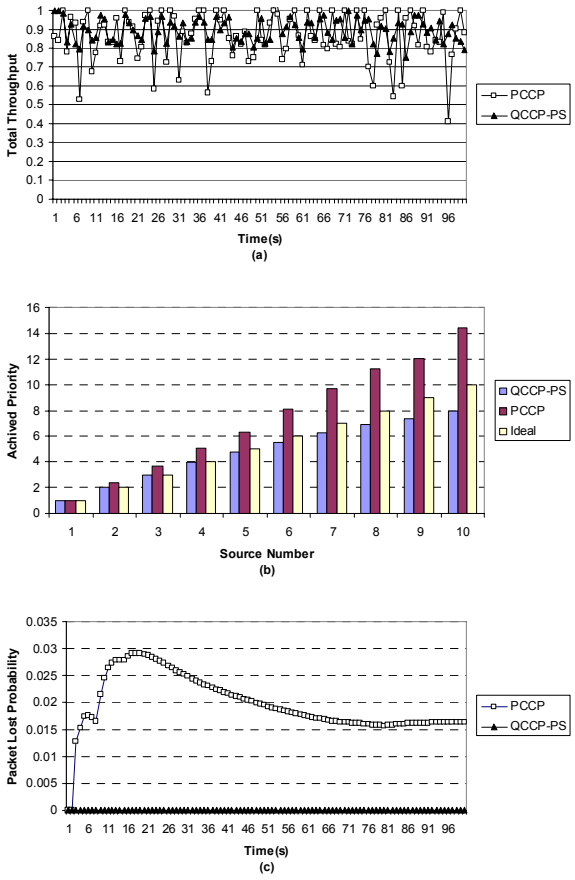

Figure 7. (a) Total throughput; (b) achieved priority; (c) packet loss probability (buffer size = 10 packets)

6. Conclusion

In this paper, we developed a congestion control protocol named QCCP-PS. QCCP-PS uses a queue based congestion indicator and can adjust the sources traffic rate based on current congestion in the upstream nodes and the priority of each traffic source. We used the buffer occupancy as a congestion indicator, and considered varying priorities for each sensor node. The proposed congestion protocol can adjust the source rate based on the current congestion status in its parent node. The performance of proposed protocol was evaluated using computer simulation. The results show that QCCP-PS can achieve low packet loss probability.

7. References

[1] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, "A survey on wireless multimedia sensor networks", *Computer Networks* 51, 921–960, 2007,.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks* 38 (4), 393–422, 2002.

[3] C-G Wang, K. Sohraby, "A survey of transport protocols for wireless sensor networks",in *IEEE Network Magazine*, 34-40, May/June 2006.

[4] C.-T. Ee and R. Bajcsy, "Congestion control and fairness for many-to one routing in sensor networks," in *Proc. ACM Sensys*, Nov. 2004.

[5]Y. G. Iyer, S. Gandham, and S. Venkatesan, "STCP: A generic transport layer protocol for wireless sensor networks," *Proc. IEEE ICCCN* 2005, San Diego, CA, Oct. 17–19, 2005.

[6] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," *Proc. ACM Sensys '04*, Baltimore, MD, Nov. 3–5, 2004

[7] C. Wang, Member, K. Sohraby, M. Daneshmand, Y. Hu," Upstream congestion control in wireless sensor networks through cross-layer optimization", *IEEE Journal On Selected Areas In Communications*, 25(4), 2007, 786-795.

[8] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," *Proc. ACM Sensys* '03, Los Angeles, CA, Nov. 5–7, 2003.

[9] A. Woo and D. C. Culler, "A transmission control scheme for media access in sensor networks," *Proc. ACM Mobicom* '01, Rome, Italy, July 16–21, 2004.

[10] C.-Y. Wan and A. T. Campbell, "PSFQ: A reliable transport protocol for wireless sensor networks," *Proc. ACM WSNA* '02, Atlanta, GA, Sept. 28, 2002.

[11] C.-Y. Wan et al., "Siphon: Overload traffic management using multi-radio virtual sinks in sensor networks," *Proc. ACM SenSys* '05, San Diego, CA, Nov. 2–4, 2005.

[12] P. Levis et al., "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," *Proc. 1st Symp. Networked Sys. Design and Implementation*, San Francisco, CA, Mar. 29–31.

[13] F. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," *Proc. IEEE SNPA* '03, Anchorage, AK, May 11, 2003.

[14] H. Zhang et al., "Reliable bursty convergecast in wireless sensor networks," *Proc. ACM Mobihoc* '05, Urbana-Champain, IL, May 25–28, 2005.

[15] Floyd, S., and Jacobson, V., "Random early detection gateways for congestion avoidance**"** *IEEE/ACM Trans. Networking*, 1(4), 397-413, August 1993.