IJICTR International Journal of Information & Communication Technology Research

Volume 3- Number 4- September 2011 (45-58)

Priority Based Congestion Control and Partial Reliability Guaranty Protocol for Wireless Multimedia Sensor Networks

Mohammad Hossein Yaghmaee Moghaddam
Department of Computer Engineering
Center of Excellence on Soft Computing and Intelligent Information Processing,
Ferdowsi University of Mashhad
Mashhad, Iran
yaghmaee@ieee.org

Donald Adjeroh
Lane Department of Computer Science and Electrical Engineering
West Virginia University
Morgantown, USA
don@csee.wvu.edu

Received: August 1, 2011- Accepted: September 15, 2011

Abstract— **Congestion is an essential problem in Wireless Sensor Networks (WSNs). It not only wastes the scarce energy due to a large number of retransmissions and packet drops, but also hampers the event detection reliability. Thus, to meet the Quality of Service (QoS) requirements for network applications, a reliable and fair transport protocol is mandatory. For some applications however, such as in wireless multimedia sensor networks, having a 100% packet delivery ratio is not necessary. In this paper, we present PCC-PRG, a Priority Based Congestion Control and Partial Reliability Guaranty protocol for wireless multimedia sensor networks. The proposed PCC-PRG protocol can be used for partial order services in WMSNs. PCC-PRG uses the maximum queue length of intermediate nodes to detect and notify congestion in the network. PCC-PRG protocol uses a rate control mechanism to adjust the transmission rate of each traffic flow based on the congestion degree in the network and also the flow priority. All transmitted packets are classified into two different priority groups: high and low. Only lost high priority packets are retransmitted. To use a node's energy efficiently, we use a hop-by-hop NACK-based reliability guaranty model. Simulation results confirm the superiority of the proposed model.**

Keywords: Wireless Multimedia Sensor Networks; Congestion Control; Reliability; Transport Protocols; Rate based control

I. INTRODUCTION

Recent advances in micro electro-mechanical systems (MEMS) technology have made the manufacture of small and low cost sensor nodes easily feasible. A large number of these disposable sensors can be networked in many applications that require unattended operations. Wireless Sensor Networks (WSNs) [1] are a set of communication networks consisting of different independent sensors that cooperatively monitor some physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The sensors measure ambient conditions related to the environment surrounding the sensor node and transform them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the vicinity of the sensor. These sensors have the ability to communicate either among each other or directly to a base-station, also called sink node. A greater number of sensors allows for sensing over larger geographical regions with greater accuracy. Wireless sensor networks have many applications in different areas of technology. Each node in a WSN is typically equipped with one or more sensors, a wireless communications device, a processor, and an energy source, usually a battery.

Wireless Multimedia Sensor Network (WMSN) [2] 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 and monitoring data. One of the most important requirements of applications in WMSNs is low delay bounds. Furthermore, some applications of WMSNs need relative resilience to losses. Streaming multimedia content is generated over longer time periods and requires sustained information delivery. WMSNs can support different types of traffic classes. An increasingly important traffic class in WMSNs is the delay-tolerant, loss-tolerant class. This type of traffic class does not need strict delay bounds. It needs high bandwidth demand for multimedia streams and moderate bandwidth for data applications. Similar to wireless sensor networks, 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 flows mainly from a large number of sensor nodes to the 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 major concern, and must be considered.

To achieve multimedia communication in WMSNs having a reliable transport mechanism is important. In traditional communication networks, the 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. 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. Reliable data delivery services are critical for applications such as file transfers, database services, transaction processing, and other mission-critical applications in where it is guaranteed that every packet will be delivered. Reliable data delivery services are designed to provide guaranteed and accurate delivery of data over unreliable or best-effort networks. This is done by implementing additional protocols that track packet deliveries and retransmit lost packets. Other services include monitoring the network for congestion and throttling back senders that are contributing to congestion. Any reliable transport protocol such as TCP offers different services including flow-control, reliable delivery and congestion control. When congestion occurs in the network, end systems and the network must work together to minimize the congestion. In contrast, flow control is used between end systems. A receiver uses flow control to signal to the sender that it is overloaded. The sender then throttles back or stops its transmission.

To provide reliability guaranty at the transport layer, ARQ (Automatic Repeat reQuest) mechanism can be used [4]. In this protocol, every packet has a unique sequence number. At any time the sender may have a number of frames outstanding and awaiting acknowledgements. A sufficiently-large sender window permits continuous transmission of new packets. A smaller link sender window causes the sender to pause transmission of new packets until a timeout or a control packet, such as an acknowledgement (ACK), is received. When packets are lost, a negative acknowledgement (NACK) is sent to the source and then the source retransmits the packet.

In traditional TCP protocol, congestion is detected at the end nodes based on a timeout or redundant acknowledgments. Both congestion and reliability are coupled with the receipt of an ACK from the receiver. TCP assumes the non-receipt of an ACK as a congestion problem and it slows down its transmission rate along with retransmitting the packet for reliability. TCP protocol has good performance in wired networks where the channels are mostly reliable. However in wireless networks, this is a huge problem as error rates are usually high in wireless media. ACK/NACK based protocol can also be used in WSNs. This approach can easily detect errors, but a huge number of status report transmissions are required. On the other hand, though NACK based protocol spends less network resources, error detection is much harder than in ACK/NACK based protocols and requires a high computational complexity.

In WSNs, the nodes use a radio channel to transmit their data toward the sink node. Because of this, congestion is a very realistic concern in sensor networks. As the power consumption is an important issue in these networks, the cost of retransmission of a lost frame is very high. This makes the congestion control problem in WSN a more urgent concern. Congestion in WSNs has a direct impact on energy efficiency and application QoS: 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 happen at the sensor nodes close to the sink, as they usually carry more combined upstream traffic. Congestion control mechanisms typically consist of three important components: congestion detection, congestion notification, and rate adjustment.

The reminder of this paper is organized as follow. Section 2 reviews related work. In Section 3, we explain the congestion control part of PCC-PRG protocol. In Section 4, the PCC-PRG loss recovery protocol is described. Simulation results which confirm the superiority of the PCC-PRG protocol are given in Section 5. In this section the performance of PCC-PRG protocol is evaluated and compared with that of the RCRT protocol. Section 6 concludes the paper.

II. Related Work

The past few years, different congestion control protocols have been proposed for WSNs. STCP [5], Fusion [6], CODA [7], PCCP [8], CCF [9] are the most well known congestion control protocols in WSNs. Recently we proposed QCCP-PS [10], a queue based congestion control protocol with priority support for wireless multimedia sensor networks. QCCP_PS showed better performance than the PCCP and CCF protocols. Different methods have been proposed to detect congestion in wireless sensor networks. Some protocols such as SCTP, Fusion and CODA use the queue length as a congestion indicator. CCF uses the packet service time to detect any possible congestion in the network. PCCP proposes the ratio of packet inter-arrival time and packet service time as a congestion indicator.

As the links in WSNs are not reliable, the end-to-end reliability model is not suitable in this type of network. To save energy at a node and to minimize overall energy consumption, most transfer protocols in WSNs use a hop-by-hop reliability model. In the hop-by-hop reliability model, intermediate nodes are supposed to participate in data transport by caching and retransmitting data packets, generating or changing the contents of control packets. To minimize energy consumption, retransmission should be reduced. Retransmission can be reduced by using hop-by-hop error recovery schemes. Different congestion control and reliability guaranty models have been proposed for WSNs. Popular examples include PSFQ [11], RMST [12], DTC [13], DTSN [14], ESRT [15], and STCP [16]. PSFQ consists of three different states includes: pump, fetch, and report. When the sink is in the pump state, it periodically broadcasts data packets to its neighbors. Upon detecting any gap in the received data packets, the sensor node goes to the fetch state. It also sends an NACK in reverse path to recover the missing fragment. In the report state, using a simple and scalable hop-by-hop reporting mechanism, the sink sends information on data delivery to the sensor nodes. ESRT protocol guarantees event reliability through end to-end source rate adjustment while RMST provides packet reliability through hop-by-hop loss recovery. In the ESRT protocol, when the achieved reliability exceeds a stated value, the source rates are decreased in a multiplicative manner. Otherwise, the source rates are additively increased.

Most existing transport protocols for wireless sensor networks address congestion control or reliable transport separately. When the congestion control and loss recovery algorithms are separated from each other, applications that need reliability can invoke only a loss recovery algorithm, or invoke a congestion control algorithm. The joint use of congestion control and loss recovery may provide the full functionality required by the transport protocols for wireless sensor networks. Thus, an important direction in WSNs is the design of such transport protocols, which not only provides congestion control but also supports mechanisms to provide reliability guaranty for different applications. To our knowledge, Rate-Controlled Reliable Transport (RCRT) [17] is the only published work that has attempted a joint consideration of congestion control and reliability guarantee in wireless sensor networks.

RCRT is a new transport protocol for wireless sensor networks. RCRT consists of four major components namely: congestion detection, rate adaptation, rate allocation and end-to-end retransmission. In the RCRT protocol a network is uncongested as long as end-to-end losses are repaired quickly which permits a few end-to-end losses caused by transient congestion, or by poor wireless links. RCRT uses the length of retransmission list as the congestion indicator. When there are too many packets in the retransmission list, it means that the congestion density is high. In this case the RCRT tries to adapt the transmission rate of each sensor node, using an Additive Increase Multiplicative Decrease (AIMD) rate control mechanism. While RCRT uses AIMD, it adapts the total aggregate rate of all the flows as observed by the sink, rather than the rate of a single flow. RCRT places its congestion control functionality at the sink, whose perspective into the network enables better aggregate control of traffic, and affords flexibility in rate allocation. The rate allocation component of RCRT essentially assigns rates to each flow in keeping with a rate allocation policy. As RCRT decouples rate adaptation from rate allocation, it is possible to obtain this flexibility. RCRT implements an NACK-based end-to-end loss recovery scheme. The sink detects packet losses and repairs them by requesting end-to-end retransmissions from source nodes.

Although RCRT has taken the first step towards joint congestion control and reliability, the current RCRT protocol still has some major problems. First, RCRT uses an end-to-end retransmissions strategy. All lost packets are detected in the sink node which implies a potentially long delay between packet loss occurrence and loss detection. As discussed earlier, hop-by-hop retransmission strategy provides a much better performance when compared with the end-to-end model. It can be shown that the RCRT protocol does not have good performance in high noisy channels. When the channel error rate is high, most packets are lost in the intermediate links. So the sink node will always detect out-of-order packets and can’t repair losses quickly. Simulation results show that the performance of RCRT degrades quickly when the channel error rate is increased. Further, when there is a high channel error rate in the network, the number of lost packets increases, and the retransmission list keeps growing. When the sink node detects too many lost packets, it decreases the transmission rate of all traffic flows which causes a significant decrease in the network throughput. RCRT protocols always provides

100% packet delivery ratio for all transmitted packets. We believe that providing 100% reliability is not always necessary for some applications of sensor networks especially in multimedia applications. For example in the MPEG-4 standard a video stream is encoded to 3 different frame types -- I, P and B, where the P and B frames depend on the I frame. Loosing an I frame causes a noticeable worsening of the video quality of all the frames in the group. However, losing a B frame may cause significantly less degradation when compared with losing an I frame.

In this paper, we present PCC-PRG, Priority-based Congestion Control and Partial Reliability Guaranty protocol for wireless multimedia sensor networks. The proposed PCC-PRG protocol can be used for partial order services in WMSNs. For some applications such as multimedia services, there exists a genuine ability to tolerate packet losses. Losing one frame per second in a 30 frame per second video or losing a segment of its accompanying audio channel is usually not a problem. The proposed PCC-PRG protocol provides partial reliability for this type of application. In PCC-PRG protocol, each packet is classified into one of two priority classes, namely, high priority packets (for example I frame in MPEG-4) or low priority packets (for examples B and P frames in MPEG-4). These two types of packets are referred to as HIGH and LOW packets, respectively. The main objective of the PCC-PRG protocol is to provide high reliability rate for HIGH packets. To reduce energy consumption, when a LOW packet is lost, the packet is not retransmitted. Rather, the lost packets are recovered (interpolated) at the sink node, using some context based recovery techniques.

The PCC-PRG protocol uses a non-binary feedback mechanism for congestion control. At the header of each packet, a 3-bit field is reserved for intermediate nodes to announce their maximum buffer occupancy. Each intermediate node compares its current buffer occupancy with the value of this field. When current buffer occupancy of the sensor node is more than the value of this field, the node writes its buffer occupancy in this field. When packets arrive to the sink node, using a simple threshold mechanism the sink node is able to detect any congestion in the intermediate nodes. The sink node uses an AIMD technique to tune the transmission rate of all flows based on current congestion degree in the network. When there is no congestion in the network, the sink node increases the transmission rate of all active flows. We use an end-to-end priority based rate adaptation policy to adjust the transmission rate of all sensor nodes efficiently. In the proposed model, the rate allocation to each traffic flow is based on its priority. When congestion is detected, the sink node calculates and informs the new transmission rate of each traffic flow. Using the Average Loss Interval (ALI) method described in [18], the packet loss rate of each traffic flow is estimated. PCC-PRG uses a rate decrease factor which is related to the maximum packet loss ratio of all traffic flows.

III. PCC-PRG CONGESTION CONTROL

In this section we explain congestion control component of the proposed PCC-PRG protocol in detail. Similar to the other transport protocols, the PCC-PRG performs three major functions including: congestion detection and notification, rate adjustment and reliability guaranty.

A. Congestion Detection and Notification

Each sensor node in the PCC-PRG protocol has two different buffers for saving incoming packets. All in-order packets are placed in a single receive buffer. The out-of-order packets are placed in an out-of-order buffer. The congestion detection unit of PCC-PRG protocol uses the queue length indicator to detect any congestion in the network. Each node monitors its entire receive buffer queue. At the source nodes, data packets are classified to two different types: HIGH packets and LOW packets. HIGH packets are high priority packets which should be delivered to the sink node in order and correctly. LOW packets are low priority packets which have more tolerance for data loss than HIGH packets. The packet's type is specified using a single bit in the packet's header. The structure of the packet's header is shown in Fig. 1.

1 bit	1 bit	3 bits
Type	Last Type	Congestion Index (CI)

Fig. 1. The packet header structure

The Type bit represents the current packet's type. If the current packet is a HIGH packet then this bit is set to 1, otherwise it is set to 0. The Last Type bit represents the value of Type field in the last submitted packet. This bit is used by the reliability guaranty model which will be explained in the next subsection. The Congestion Index (CI) field is used to indicate the maximum congestion degree in all sensor nodes among the path from source node to the sink. We use the buffer occupancy as a congestion indicator. The initial value of this field is set to zero. Every intermediate node compares its current queue length with the value of this field. If the current queue length is more than the value of this field, the sensor node writes its current queue length in this field; otherwise the value of this field is not changed. At the sink node, the value of CI field which represents the maximum queue length of all sensor nodes along the path is used to detect possible congestion in the network.

Since energy consumption is a major constraint in wireless sensor networks, we reduce energy consumption by allocating only 3 bits to the CI field. At each intermediate sensor node, the current value of queue length is converted to a 3 bit binary number. For this purpose, the queue size is divided into 8 portions. Based on the current value of queue length, a proper number (between 0-7) is assigned to the queue length. Then it is compared with the value of CI field. If the node queue length is more than the value of CI field, the node replaces the CI field with its current queue length. The replacement is done if and only if the queue length is greater than the value of this field.

Simulation results confirm that using only 3 bits produces an acceptable accuracy, without an intolerable amount in energy consumption.

Suppose there are n sensor nodes (including sink node) among the path between source and sink. Let node n be the source node and node 1 is the sink node. If at time t a packet arrives at the sink node, the current value of the CI field, $CI(t)$ then represents the maximum queue length of all sensor nodes along the path from source to sink. $CI(t)$ can be expressed as follows:

$$CI(t) = \max\{q_n(t-d_n), q_{n-1}(t-d_{n-1}), \ldots, q_2(t-d_2), q_1(t)\} \quad (1)$$

where $q_i(t)$ represents the queue length of sensor node i at time t and d_i is the one way delay (forward delay) between sensor node i and the sink node. As the length of CI field is 3 bits, $q_i(t)$ is always a number between 0-7.

After a packet arrives at the sink node, $CI(t)$ can be used to determine the congestion density in the network. When $CI(t)$ is close to 7, it means that at least the queue of one sensor node along the path is close to being full. In this case to prevent any packet loss the transmission rate of all traffic flows is decreased. On the other hand when $CI(t)$ is close to 0, it means that the queue length of all sensor nodes along the path is low. In this case to increase the network throughput, the transmission rate of all flows is increased.

To detect congestion efficiently, two different thresholds $\tau_{\min}$ and $\tau_{\max}$ are defined. With 3 bits in the packets header reserved for congestion indicator, we have only 8 possible values 0 to 7 for $\tau_{\min}$ and $\tau_{\max}$. In the simulations we set $\tau_{\min}$ and $\tau_{\max}$ to 1 and 6, respectively. If $CI(t) > \tau_{\max}$, the network is congested. In this case the flow rates are decreased. On the other hand, when $CI(t) < \tau_{\min}$, there is no congestion in the network. In this case the flow rates are increased. When we have $\tau_{\min} \le CI(t) \le \tau_{\max}$, no action is taken. In this case the flow rates are not changed.

B. Rate Adjustment

To reduce the negative impact of congestion, whenever congestion is detected, it is necessary to decrease the transmission rate of all traffic flows. On the other hand, when the congestion index is low, to use the network resource efficiently, the flow rates should be increased. TCP and most transport control mechanisms use an AIMD approach to additively increase flow rates in the absence of congestion, and multiplicatively decrease flow rates when congestion is detected. In the PCC-PRG protocol, we propose an end to end priority based rate adaptation policy to adjust the transmission rate of all traffic flows efficiently. In the proposed protocol, the rate allocation to each traffic flow is based on its priority. Suppose each traffic flow j has priority $P_j\ (j = 1, \ldots, N)$ where N is the number of active flows in the network. Under the PCC-PRG protocol, traffic flows that have high priority receive more bandwidth.

Fig. 2 provides more details on the proposed congestion control and rate adjustment algorithm. When a new traffic flow needs to be established, a connection request packet is transmitted toward the sink node. The sink node assigns an initial transmission rate to each traffic flow j in proportion to the flow's priority. Suppose r_0 is the initial total transmission rate in the network. Based on the number of active traffic flows (N) and the flow's priority (P_j), each traffic flow j gets an initial transmission rate, namely, $r^j(0)$. Thus high priority flows start their transmission with more initial rate than the low priority flows. Clearly, r_0, the sum of all initial flow rates should be $r_0 = \sum_{j=1}^{N} r^j(0)$

Congestion detection and rate adaptation algorithm

A. Connection establishment phase:

1. The source node transmits a connection request packet to the sink node. This packet includes the flow priority.
2. Suppose there are N active flows in the network (includes the new one), for each traffic flow j, the sink node computes the initialize transmission rate $r^j(0)$ as:

$$r^j(0) = r_0 \cdot \frac{P_j}{\sum_{i=1}^{N} P_i}$$

3. The sink node informs the source node of the initialize transmission rate of new incoming traffic flow.

4. The sink node allocates a flow record to the new incoming traffic flow.

B. Data transmission phase:

-At the intermediate nodes:

5. Each sensor node among the path converts its current queue length to a 3-bit binary number.

6. If the current queue length is more than the value of the Congestion Index field in the packet header, it is replaced with the current queue length.

-At the sink node:

7. By detecting any out of order packet at time t, using ALI method [18] compute the packet loss ratio of each traffic flow j ($P^j{}_{loss}(t)$).

8. For each traffic flow j, compute the value of $D^j(t)$:

$$D^j(t) = \frac{1 - P^j{}_{loss}(t)}{1 + P^j{}_{loss}(t)}$$

9. Compute :

$$D^{\max}(t) = \max_{j=1}^{N}\{D^j(t)\}$$

10. Set the value of $CI(t)$ to the value of Congestion Index field existing in the received packet's header.

11. If the time elapsed since last rate adaptation is more than $RTT_{\max}$ do:

12. If $CI(t) > \tau_{\max}$, compute new rate for each traffic flow j:

$$r^j(t) = r^j(t).D^{\max}(t)$$

13. Else if $CI(t) < \tau_{\min}$ compute new rate for each flow j:

$$r^j(t) = r^j(t) + \lambda.\frac{P_j}{\sum_{i=1}^{N} P_i}.C$$

where, C is the maximum capacity of the sink node, and λ is the adaptation parameter fixed at 0.05.

14. If $r^j(t) > r^j{}_{\max}$, then set $r^j(t) = r^j{}_{\max}$

15. If $\tau_{\min} \le CI(t) \le \tau_{\max}$, flow rates are not changed.

16. Inform all active flows of the new flow rate.

C. Connection release phase:

17. When the traffic flow does not have any data to be sent, a connection release packet is transmitted toward the sink node.

18. Upon receiving this packet at the sink node, the flow record is deleted.

19. The sink node decreases the number of active flows by one. (i.e N=N-1)

Fig. 2. The PCC-PRG congestion detection and rate adaptation algorithm

At the source node, all packets are classified as HIGH or LOW based on their precedence as described earlier. If the packet contains important data, the value of Type bit in the packet's header is set to 1; otherwise it is set to 0. Furthermore, according to the value of Type bit in the last packet, the Last Type bit is set. In all intermediate sensor nodes, the current buffer occupancy is converted to a 3-bit binary number. If the buffer occupancy is greater than the current value of CI field existing in the incoming packet's header, the value of this field is replaced by the current buffer occupancy.

The sink node which usually has more energy and more computation power performs the congestion detection and rate adjustment. For this purpose, it is necessary to estimate the packet loss probability of each traffic flow. To estimate the packet loss of each individual traffic flow, we use the Average Loss Interval (ALI) method given in [18].

For each traffic flow j, let $S_k (k = 1,..,8)$ be the number of packets in the k-th most recent loss interval. Let the most recent interval S_0 be defined as the interval containing the packets that have arrived since the last loss. Two variables $\hat{s}_0$ and $\hat{s}_1$ are used:

$$\hat{s}_0 = \frac{S_0 + S_1 + S_2 + S_3 + 0.8S_4 + 0.6S_5 + 0.4S_6 + 0.2S_7}{6}$$

$$\hat{s}_1 = \frac{S_1 + S_2 + S_3 + S_4 + 0.8S_5 + 0.6S_6 + 0.4S_7 + 0.2S_8}{6} \tag{2}$$

The packet loss probability of traffic flow j ($P^j{}_{loss}$) is then obtained as follows [18]:

$$P^j{}_{loss} = \frac{1}{\max\{\hat{s}_0, \hat{s}_1\}} \tag{3}$$

Fig. 3 shows an example of the ALI method. In this example the value of $\hat{s}_0$ and $\hat{s}_1$ are calculated as: $\hat{s}_0 = \frac{20.6}{6} \approx 3.4$ and $\hat{s}_1 = \frac{20}{6} \approx 3.3$, respectively. In this example, the packet loss probability is equal to $\frac{1}{\max(3.4,3.3)} = \frac{1}{3.4} = 0.294$.

Using the ALI method described above, the sink node calculates $P^j{}_{loss}$, the packet loss probability of each traffic flow j. For traffic flow j the time-dependent factor $D^j(t)$ is computed as shown in Line 8 of the proposed algorithm. As the rate adjustment unit should adjust the flow rate by considering the worst case in the network, similar to RCRT protocol, the PCC-PRG uses also the maximum value of $D^j(t)$ for rate adaptation.

The sink node extracts the congestion index CI exists in each packet. Note that the current queue size of sink node is also considered to obtain the correct value of the CI field. At each time t, based on the current value of $CI(t)$, the sink node is able to detect any congestion in the network.

When $CI(t) > \tau_{\max}$, it means that there is congestion in at least one node in the path between source and sink. To prevent any further packet loss the flows rate should be decreased multiplicatively. Based on Line 12 of the proposed algorithm, when congestion is detected all flow rates are decreased using multiplicative factor $D^{\max}(t)$. When the packet loss probability of all traffic flows is zero (which means there is no congestion in the network), the value of $D^{\max}(t)$ is equal to 1. In this case the flow rates aren't changed. Whenever there is congestion in the network, the packet loss probability is increased. By increasing the packet loss probability, the value of $D^{\max}(t)$ is decreased toward zero which causes the decrease in the flow rates. By decreasing the flow rates, the congestion density is also decreased.

When $CI(t) < \tau_{\min}$, it means that all buffers in the path are underutilized. To use network capacity efficiently, the flow rates are increased additively. Line 13 of the proposed algorithm shows that the amount of increase applied to each flow rate is directly related to its priority. This means that the transmission rate of high priority traffic flows are increased more than that of low priority flows. The maximum transmission rate of each traffic flow j is limited to a predefined value $r^j{}_{\max}$.

Whenever PCC-PRG determines that it is necessary to change the flow rates, it calculates the new rate of all traffic flows and sends the new rate information of each flow to its corresponding sensor node. As the new rate information takes time to propagate to the sources, we need to avoid potential oscillations in rate allocation. The PCC-PRG protocol ensures this is achieved by not making another rate adaptation decision until it observes the effect of the previous decision. For this purpose, PCC-PRG estimates the maximum round trip time ($RTT_{\max}$) and waits for at least $RTT_{\max}$ time units, before any rate changes. This waiting time allows enough time for the rate feedback to reach the sources.

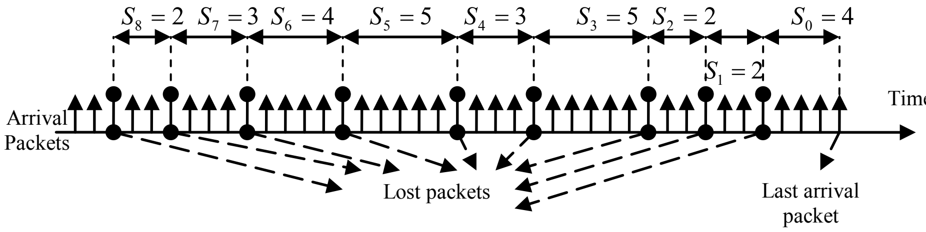


Fig. 3. An example of average loss interval method used to compute packet loss probability

IV. PCC-PRG LOSS RECOVERY

As we mentioned in Section 1, in wireless sensor networks the hop-by-hop reliability guaranty model is more suitable than the end-to-end model. To consume the network's energy more efficiently, the PCC-PRG protocol uses the hop-by-hop reliability guaranty model. For many applications (such as video streaming) having a 100% packet delivery ratio is not necessary. Some real time applications are tolerant against packet loss. For this type of applications, providing a high throughput and low delay is more important than low packet loss. For example in the MPEG-4 standard a video stream is encoded to 3 different frames types, I, P and B, where the P and B frames depend on the I frame. Loosing an I frame causes a noticeable worsening of the video quality of all the frames in the group. The PCC-PRG protocol provides partial reliability for these types of applications. As we mentioned before, at each traffic source in the proposed model, all packets are marked in two different classes includes: high priority class (for example I frame) and low priority class (for examples B and P frames). These two types of packet are referred to as HIGH and LOW packets, respectively. In each sensor node i, the normalized rate of HIGH and LOW packets are denoted by ρ^i_{HIGH} and ρ^i_{LOW}, respectively. Note that sum of HIGH and LOW packet rates is always equal to 1 ($\rho^i_{HIGH}+\rho^i_{LOW}=1$).

At each intermediate nodes (including sink node), the packets which are received in the correct order are placed in the receive buffer. A copy of each HIGH received packet is also saved in a cache memory for any possible retransmission. Note that the LOW packets are not saved in the cache memory. When a node receives the ACK of its already submitted HIGH packet, it removes the packet from its local cache. Packets which are received out of order are forwarded to the retransmission buffer. In the proposed model, each node on the forward path from source to sink caches the HIGH packets. To detect any gap in received packets, each packet must contain a sequence number. When a node detects any gap in the sequence number of received packet it realizes that a packet is lost. It looks at the Last Type bit in the arrived packet. If this bit shows that the previous lost packet is a HIGH packet, a NACK message is sent to the next hop on the reverse path toward the source. But whenever the lost packet is a LOW packet, the node does not send any NACK packet. If the requested lost HIGH packet is found in the local cache, a copy of the lost packet is retransmitted. If not, the NACK message is forwarded to the next hop toward the source. Caching of the HIGH packets along the forward path is used to limit power waste due to end-to-end retransmission. Each node uses a timer-based loss detection mechanism. When the requested lost packet doesn't arrive in a predefined time interval, an NACK message is sent again to the next hop in the reverse path. The value of timer could be dynamically tuned based on the delay in the network.

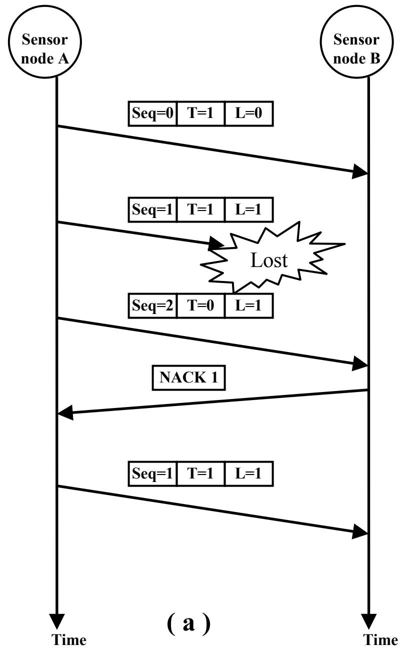
Whenever a node detects an out-of-order packet, if the lost packet is a HIGH packet then the out-of-order packet is placed in the retransmission buffer. But when the lost packet is a LOW packet, the out of order packet is placed in the receive buffer. In this case, the content of the lost LOW packets are estimated at the sink node using prediction techniques.

To understand the loss recovery protocol used in the proposed PCC-PRG, we use an example. Consider Fig. 4 which shows two neighboring sensor nodes A and B. Some data packets are transmitting from node A to node B. The symbols Seq, T and L shown in this figure refer to the packet sequence number, packet type and the last packet type, respectively. As shown in Fig. 4(a), due to some noise in the channel packet #1 which is a HIGH packet is lost. When the next packet arrives at node B, by looking at the sequence number of the received packet, the sensor node B determines that there is a gap in received packet sequence numbers. Thus, it will recognize that packet #1 has been lost. Before sending any NACK message, node B looks at the Last Packet Type bit in the received packet. Since this bit is set to 1, node B will know that the last lost packet must be a HIGH packet. So it sends an NACK message to node A and asks it to retransmit the packet #1. Upon receiving the NACK message, the node A looks at its local cache and retransmits the lost packet #1 to node B. Using this simple method, the lost HIGH packets are recovered at intermediate nodes. Fig. 4(b) shows a situation that a LOW packet is lost. In this figure, packet #2 which is a LOW packet has been lost. When the next packet (packet #3) is received at node B, by looking at the sequence number of received packet, this node realizes that the packet #2 has been lost. But as the last packet type is set to 0, (L=0), node B understands that the lost packet is a LOW packet, so it places the received packet in its receive buffer and no NACK message is sent.

Note that the proposed loss recovery mechanism works well when there is single packet error. If some burst errors occur in the network, some sequence of packets will be lost. In this case, the sensor node does not know whether the lost packets are HIGH or LOW, thus, performance in loss recovery will be affected. We propose two possible solutions for this problem of burst errors. First, we can extend the size of last packet type field so that the sender node is able to report the type for more transmitted packets. In this case, when a burst error occurs in the network, upon receiving the next correct packet, the node can determine which of the lost packets are HIGH packets by looking at the Last Packet Type field. In this case the node transmits an NACK message for the lost HIGH packets. A second possible solution is to transmit NACK messages for all lost packets. When such NACK messages are received at the sender node, it searches its cache memory for the lost HIGH packets. If it finds any HIGH packet that corresponds to an NACK message, it retransmits the lost HIGH packet. When it does not find any HIGH packet reported in the NACK message, it realizes that the lost packet is a LOW packet and ignores the NACK message.

V. SIMULATION RESULTS

Using computer simulations, we evaluated the performance of the proposed PCC-PRG protocol under different scenarios. We then compared with the previously proposed RCRT protocol, which is closest in principle to the proposed method. To our knowledge, the RCRT protocol is the only rate-base protocol that supports joint congestion control and reliability guaranty in wireless sensor networks. Although PCCP is one of the best existing protocols for priority-based rate control for the wireless sensor network, it does not support any loss recovery. For the purpose of performance evaluation, we simulated a cluster-based wireless sensor network with 20 sensor nodes as shown in Fig. 5. Nodes 10, 17 and 20 are the cluster heads and the others are cluster members. All sensor nodes have a random service time. All nodes have the same buffer size, set at 100 packets. The mean service time of all nodes is equal to 0.01s. The simulation time is set to 500 seconds. For simplicity, we assume that each sensor node produce only one traffic flow. For each traffic flow j the maximum flow rate ($r^j{}_{\max}$) was set to 5 packet/s. $(\tau_{\min}, \tau_{\max}) = (2,5)$



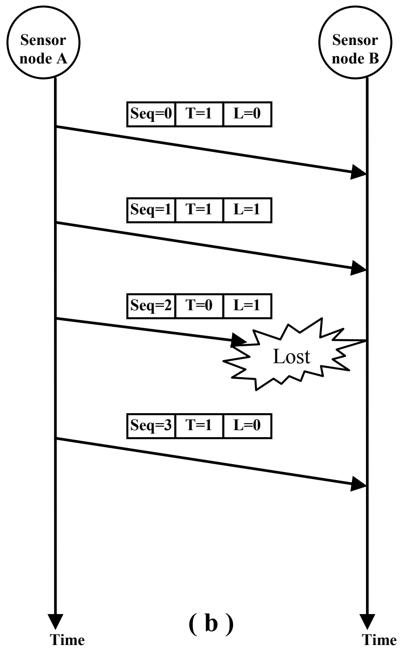


Fig. 4. An example of loss recovery in PCC-PRG protocol (a) a HIGH packet is lost; (b) a LOW packet is lost

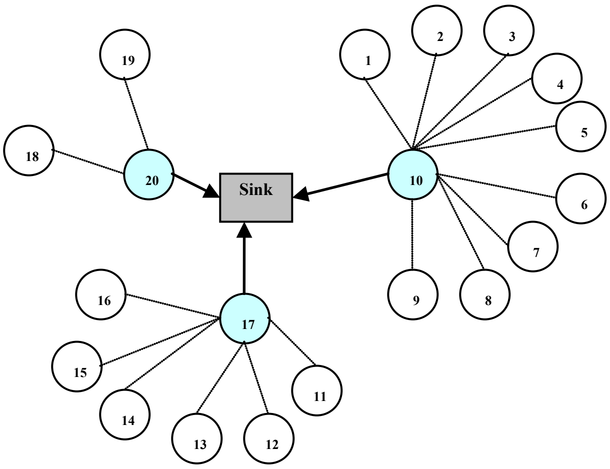


Fig. 5. Network topology used in the simulation

A. *Effect of Thresholds*

We first evaluated the effect of τ_{max} and τ_{min} on the performance of PCC-PRG. All traffic flows have the same priority $(P_j = 1, j = 1,...,20)$, with all packets marked as HIGH packets. The results are given in Fig. 6. The number of NACKs and the total utilization are plotted against different value of (τ_{min} , τ_{max}) pair, at different levels of packet error rate (PER). By increasing the PER, the number of lost packets is also increased. By increasing the number of lost packets also leads to an increased number of NACKs. Furthermore, from Fig. 6(b), it is observed that increasing the PER leads to some decrease in utilization. The figure further shows that by increasing (τ_{min} , τ_{max}) both the number of NACKs and the channel utilization should be increased. This is because when thresholds are increased, the probability of congestion detection is decreased. So PCC-PRG doesn't detect congestion and increases the flow rates. By increasing the flow rates the number of transmitted packets and also the utilization is increased.

In Fig. 7(a), we plot the variation of flow rate at source number 5 over simulation time at PER=4%, for two different values of (τ_{min} , τ_{max}) pair. The results show that using (τ_{min} , τ_{max})=(1,3) results in average flow rate of 4 packet/s, while using (τ_{min} , τ_{max})=(2,5) the average is 4.23 packet/s. Note that since the maximum source rate is fixed to 5 packet/s, the flow rate can not exceed this number. For two different values of (τ_{min} , τ_{max}) pair the total flow rate is given in Fig. 7(b). Results confirm that when the thresholds are set to higher values $((\tau_{min}, \tau_{max}) = (2,5))$, the total flow rate is equal to 85 packet/s while when $(\tau_{min}, \tau_{max}) = (1,3)$ the total flow rate is equal to 80 packet/s. Note that as the maximum rate of each traffic flow is set to 5 packets/s, the total flow rate can't exceed 20*5=100 packets/s.

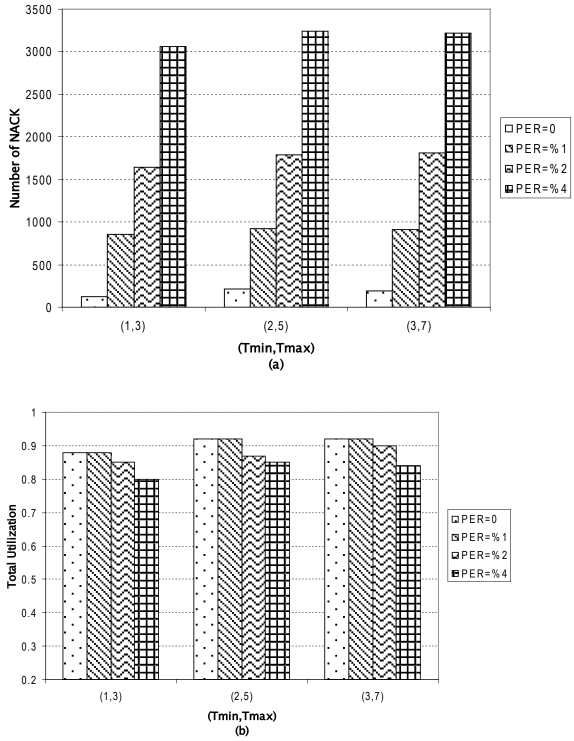


Fig. 6. Effect of PER and thresholds

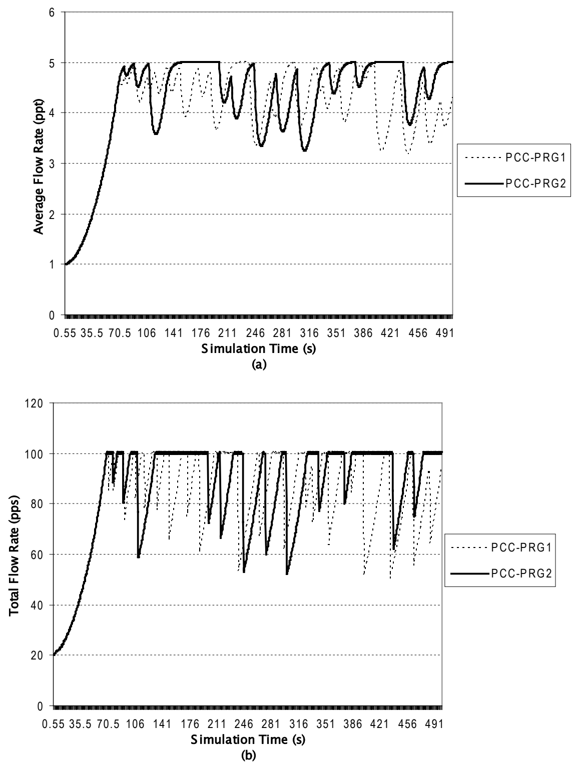


Fig. 7. The flow rate at two different values of thresholds (PER=4%): (a) flow rate of source 5; (b) total flow rate

(PCC-PRG1 and PCC-PRG2 corresponds to threshold pairs equal to (1,3) and (2,5), respectively).

B. Channel Utilization and Packet Delivery Ratio

To evaluate performance in terms of packet utilization and packet delivery ratio, the ($\tau_{\min}$, $\tau_{\max}$) threshold pair was set to (1,6). Fig. 8 shows the performance on number of NACKs normalized by total received packets, channel utilization and packet delivery ratio with respect to PER. We have also included results for the RCRT protocol for comparison. With increasing PER, the number of NACKs also increases. The results show that when the PER is high, the RCRT fails to a produce a good performance. Simulation results confirm that when the PER is more than 3%, the utilization for RCRT, decreases rapidly. Results in Fig. 8(c) confirm that both RCRT and PCC-PRG have a near 100% packet delivery ratio at different error rates.

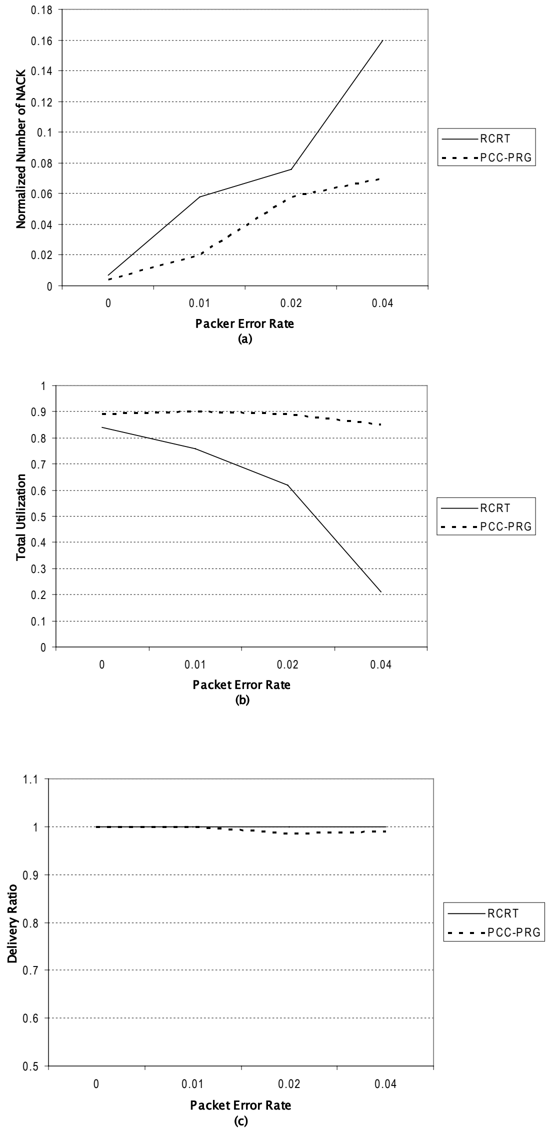


Fig. 8. Performance of PCC-PRG and RCRT at different values of PER: (a) normalized number of NACKs vs PER; (b) total utilization vs PER; (c) packet delivery ratio vs PER

In Fig. 9 the number of NACKs and the packet delivery ratio are plotted against different values of HIGH packet rate (ρ^{i}_{HIGH}). PER was fixed at 2%. In the PCC-PRG protocol, the NACK is transmitted for only HIGH lost packets. Thus, as can be observed, by decreasing the HIGH packet rate, the number of NACKs is also decreased. From Fig. 9(b), for packets that are of type HIGH, the packet delivery ratio of PCC-PRG is very close to that of RCRT. Note that as the PER is low, most packets including both HIGH and LOW packets are not lost. Thus, the delivery ratio of LOW packets is always high (more than 90%), and is not dependent on the HIGH packet rate.

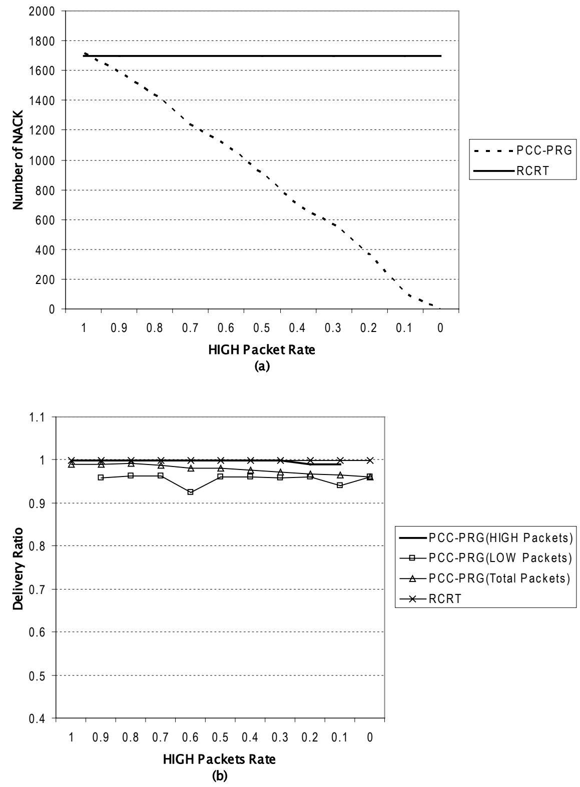


Fig. 9. Performance of PCC-PRG under different values of HIGH packet rate: (a) Number of NACKs; (b) packet delivery ratio (PER=2%)

C. Effect on Energy Consumption

In the next simulation trial we evaluate the energy consumption of the PCC-PRG and compare it with that of the RCRT protocol. As the PCC-PRG introduces some additional bits in the header of each transmitted packet, upon a casual consideration, it will seem that PCC-PRG has more energy consumption than RCRT. But as PCC-PRG is also able to decrease the number of NACKs as well as the number of retransmissions. Thus, overall, we should expect the total energy consumption for PCC-PRG to be less than that of the RCRT protocol. To evaluate the energy consumption for the two protocols, we define a performance metric called *normalized energy consumption due to retransmission,* $\overline{E}_C$, as follows:

$$\overline{E}_C = \frac{E_R}{E_T} \tag{4}$$

where E_T is the total energy consumption and E_R is the energy consumption due to retransmission and overhead (note that PCC-PRG adds 5 redundancy bits to each packet). For PCC-PRG protocol E_R consists of 3 parts: energy consumption due to transmission of NACK packets, energy consumption due to retransmission of lost packets, and energy consumption for transmitting the overhead fields. Note that for the RCRT protocol there is no energy consumption due to overhead fields. E_R values for RCRT and PCC-PRG protocols are calculated as follows:

$$E_R^{RCRT} = (n_{NACK}.N_{size} + n_{Lost}.P_{size})h.E_b$$

$$E_R^{PCC-PRG} = (n_{NACK}.N_{size} + n_{Lost}.P_{size})\bar{h}.E_b + n_T.b.h.E_b \quad (5)$$

where n_{NACK} is number of NACKs, n_{Lost} is the number of lost packets, N_{size} and P_{size} are the size of NACK packet (which was set to 10 bytes) and data packets in bits, h is the number of hops between source nodes and sink, $\bar{h}$ is the average hop cont for NACK transmission, n_T is the total number of transmitted data packets, b is the number of overhead bits in the proposed protocol (which is equal to 5 bits) and E_b is the required energy to transmit and receive one bit (which was set to 5 nJ). For both protocols the E_T is calculated as follows:

$$E_T = n_T.P_{size}.h.E_b \quad (6)$$

The results are shown in Fig. 10. Fig. 10(a) shows the variation of normalized energy consumption $\overline{E}_C$ against versus packet size at different values of HIGH packet rate. It can be seen that when the rate for HIGH packets is decreased, the energy consumption of PCC-PRG protocol is also decreased. Results confirm that for any value of HIGH packet rate, the energy consumption of the proposed PCC-PRG protocol is less than that of RCRT protocol. Fig. 10(b) shows the performance in terms of $\overline{E}_C$ at different PERs. It is observed that by increasing the PER, the number of NACKs and the number of retransmission packets are also increased which lead to an increase in the energy consumption.

D. *Packet Loss Ratio and End-to-end Delay*

We evaluated the packet loss ratio and end to end delay, for both RCRT protocol and the proposed PCC-PRG protocol. Fig. 11 shows the results. As can be observed, since PCC-PRG controls congestion based on maximum queue length of the intermediate nodes, its packet loss performance is better than that of RCRT. The performance of both protocols in terms of end-to-end delay is shown in Fig. 11(b). On average, the PCCP-PRG protocol results in a slightly more end-to-end delay than the RCRT protocol.

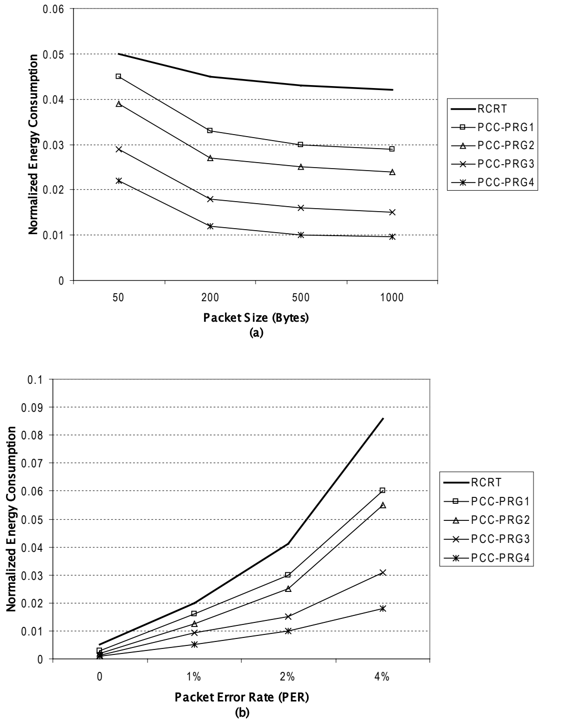


Fig. 10. Energy consumption performance: (a) normalized energy consumption vs packet size (PER =2%); (b) normalized energy consumption vs PER (packet size=500 bytes) (PCC-PRG1, PCC-PRG2, PCC-PRG3 and PCC-PRG4 corresponds to HIGH packet rate equal to 1,0.8,0.5 and 0.3, respectively)

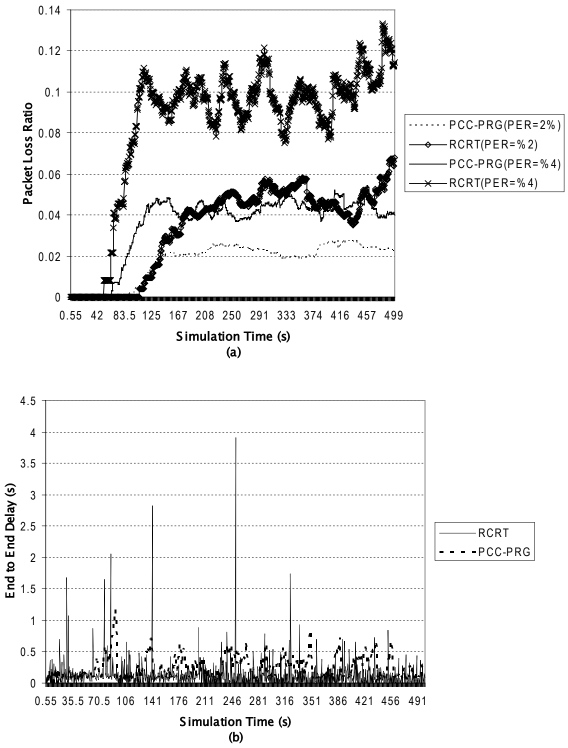


Fig. 11. Performance of RCRT and PCC-PRG with respect to packet loss ratio (a) and end-to-end delay (b).

To evaluate the performance of PCC-PRG and RCRT protocols under sudden changes in traffic load, we performed another simulation trial. In this case, at time 250 s, 5 sensor nodes 3 to 7, go on. Fig. 12 (a) shows the variation in flow rate for source number 1 over simulation time. Observe the increase in the flow rate of ON sources around the 250s time instant. Clearly, the figure shows that PCC-PRG is better that RCRT in detecting increases in traffic load. The packet loss ratio is plotted in Fig. 12(b). At the beginning of simulation time, all queues are empty, and hence there is not any packet loss in the network. At time instant 250s when some flows go ON, there is a sudden increase in the packet loss ratio of the RCRT protocol, while the loss ratio for PCC-PRG remains relatively stable. The reason is that PCC-PRG uses the maximum buffer occupancy as the congestion indicator. Thus, whenever the data rate increases, the PCC-PRG protocol can tune the data rate of its active sources in a manner that avoids a sudden increase in the packet loss ratio for active flows.

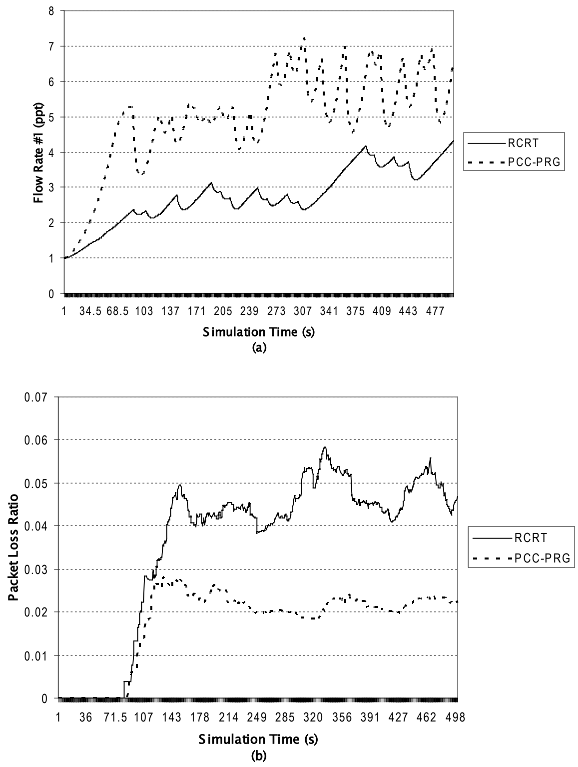


Fig.12. Performance of under sudden increase in traffic load; (a) transmission rate for flow #1; (b) total packet loss .

E. *Discriminating Between Packet Classes*

To evaluate the ability of PCC-PRG protocol in discriminating between low and high priority traffic flows, we performed a new simulation experiment to measure the flow rate at different nodes. In this case, each traffic flow j has a different priority which is equal to its flow number ($P_j = j, j = 1,...,20$). This means that a traffic flow with a higher number have a higher priority. Fig. 13 (a,) shows the results for selected flows. The average flow rate for flows 1, 5, 10 and 20 were recorded as 0.39, 1.95, 3.89 and 7.79 packets per second (pps), respectively. It is observed that the PCC-PRG is successful in assigning flow rates based on flow priority. Fig. 13(b) shows the total average flow rate for both protocols plotted over simulation time. The average total flow rate of RCRT and the proposed PCC-PRG were 39.38pps and 82.26pps, respectively. The variation of goodput over simulation time is shown in Fig. 13(c), for both protocols. It can be seen that PCC-PRG has about two times more goodput than the RCRT protocol. This is also very close to the factor in average total flow rate between the two protocols.

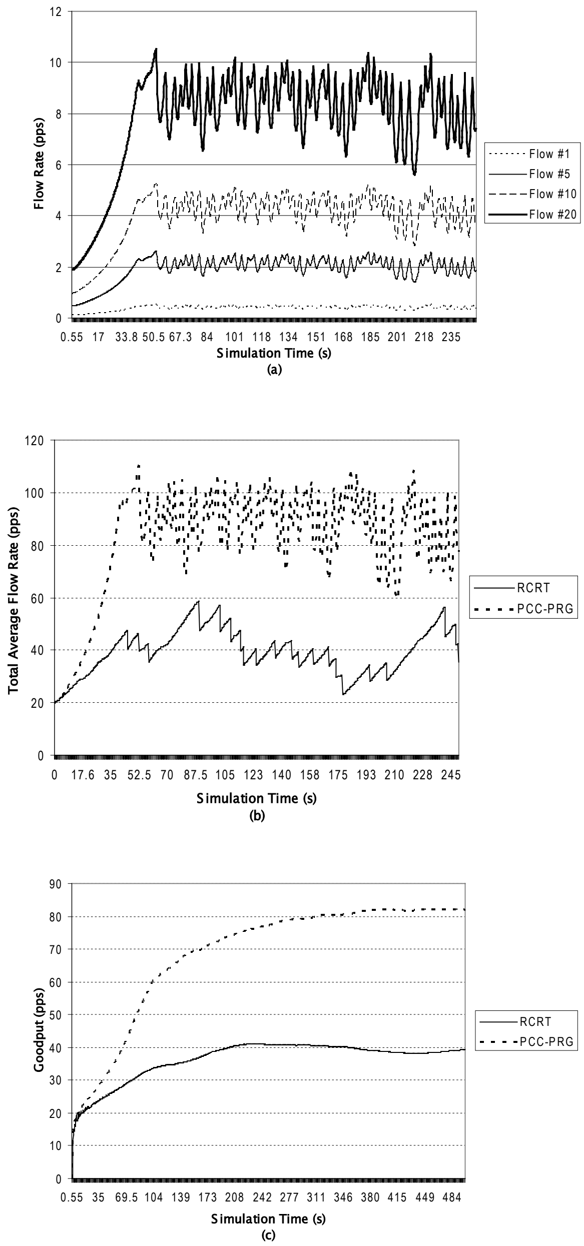


Fig. 13. Performance of PCC-PRG under different flow priority: (a) flow rate at selected nodes; (b) total flow rate; (c) Goodput

VI. CONCLUSION

In wireless multimedia sensor networks, there are a set of real time applications which do not need 100% reliability. For this kind of applications having high throughput is more important than having zero packet loss. In this paper, we have presented a priority based congestion control and partial reliability guaranty protocol for this type of applications in wireless multimedia sensor networks. The proposed protocol can be used for partial order services in sensor networks. In the proposed protocol, at the source nodes based on the precedence of data, all packets are marked as HIGH or LOW. The proposed protocol provides high reliability guaranty for HIGH packets. Using a non-binary feedback mechanism the maximum queue length of the sensor nodes along the path between source and destination is reported to the sink node. Based on the current value of maximum queue length of the sensor nodes in the network and by using two different thresholds, the sink node is able to detect any possible congestion in the network. When the queue length exceeds a predefined threshold, to prevent any packet loss, the transmission rate of all active flows is decreased. On the other hand, when the maximum queue length is less than a predefined threshold, to use the network resource efficiently the flow rates are increased. The amount of increase in the flow rates is proportional to the priority of each traffic flow. To consume the network's energy efficiently, the proposed protocol uses a hop-by-hop reliability guaranty model. All high priority packets are cached in the intermediate nodes for any possible retransmission. Only high priority lost packets are explicitly retransmitted. Lost low priority packets are never retransmitted, but are recovered by prediction or interpolation at the sink. The effect is a significant decrease in the energy consumption, given the decreased number of NACKs as well as number of retransmission packets. Simulation results confirmed that the proposed protocol a better performance compared with the state-of-art.

REFERENCES

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A survey on wireless sensor networks", IEEE Communication Magazin (2007) 108-114.

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

[3] Chonggang Wang, Kazem Sohraby, "A Survey of Transport Protocols for Wireless Sensor Networks", IEEE Network Magazine, May/June 2006, 34-40.

[4] G. Fairhurst, "Advice to link designers on link Automatic Repeat reQuest (ARQ) ", Request for Comments: 3366, RFC3366, August 2002

[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.-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.

[8] 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, vol. 25, no. 4, May 2007, 786-795.

[9] C.-T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," Proc. ACM Sensys '04, Baltimore, MD, Nov. 3–5, 2004.

[10] M.H.Yaghmaee, Donald Adjeroh, "A New Priority Based Congestion Control Protocol for Wireless Multimedia Sensor Networks", will be published in 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2008), Newport Beach, CA, June 23-27, 2008

[11] 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.

[12] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks,"Proc. IEEE SNPA '03, Anchorage, AK, May 11, 2003.

[13] Dunkels et al., "Distributed TCP Caching for Wireless Sensor Networks,"Proc. 3rd Annual Mediterranean Ad Hoc Net. Wksp., Bodrum, Turkey, June 27–30, 2004

[14] Marchi, Bruno; Grilo, Antonio; Nunes, Mario, **"**DTSN: Distributed Transport for Sensor Networks" IEEE Symposium on Computers and Communications, 2007, ISCC 2007,1-4 July 2007

[15] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to- Sink Reliable Transport in Wireless Sensor Networks," Proc. ACM Mobihoc '03, Annapolis, MD, June 1–3, 2003.

[16] 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.

[17] Eongyeup Paek, Ramesh Govindan, "RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks", Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), November 2007

[18] S. Floyd, M. Handley, J. Padhye, and J. Widmer. "Equation-based congestion control for unicast applications", Proc. ACM SIGCOMM Conference, 2000.

Mohammad Hossien Yaghmaee was born on July 1971 in Mashad, Iran. He received his B.Sc. degree in Communications Engineering from Sharif University of Technology, Tehran, Iran in 1993, and M.Sc. degree in communications engineering from Tehran Polytechnics (Amirkabir University of Technology) in 1995. He received his Ph.D. degree in communications engineering from again Tehran Polytechnics in 2000. He has been a computer network engineer working with several networking projects in Iran Telecommunication Research Center (ITRC) since 1992. From November 1998 to July1999, he was with Network Technology Group (NTG), C&C Media research labs., NEC corporation, Tokyo, Japan, as visiting research scholar. September 2007 to August 2008, he was with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA as the visiting associate professor. He is author of 5 books all in Farsi language. He has published more than 100 international conference and journal papers. His research interests are in Wireless Sensor Networks (WSNs), Traffic & Congestion Control, High Speed Networks including ATM and MPLS, Quality of Services (QoS), Fuzzy Logic Control and Smart Grid Networks.

Donald A. Adjeroh received his Ph.D. degree from Chinese University of Hong Kong in 1997. He is currently an associate professor at the Lane Department of Computer Science & Electrical Engineering, West Virginia University. His research interests are in Multimedia Information Systems (images, video, and audio), Distributed Multimedia Systems, Multimedia Data Storage and Compression, Image and Video Processing, Computational Vision and Bioinformatics.