On-Demand Reliable Medium Access in Sensor Networks

Ratnabali Biswas, Vivek Jain, Chittabrata Ghosh and Dharma P. Agrawal
OBR Center for Distributed and Mobile Computing
Department of ECECS, University of Cincinnati
{biswasr, jainvk, ghoshc, dpa}@ececs.uc.edu

Abstract

A wireless sensor network typically consists of a dense deployment of sensor nodes to achieve higher resolution and better network coverage. Having a dense network increases the fault-tolerance and robustness of the system. However, if not properly handled, it can lead to more collisions during transmission and also network congestion. Furthermore, wireless communication is inherently unpredictable and error-prone. Hence, it is imperative to design an efficient medium access control (MAC) protocol that facilitates guaranteed delivery of data over unreliable wireless links. In this paper, we have designed an on-demand reliable MAC (RMAC) protocol that enables timely delivery of data. We have demonstrated its superior performance over existing reliability-enforcing approaches in terms of reliability, latency, scalability and energy-efficiency.

1. Introduction

Sensor networks represent a new paradigm for reliable environment monitoring and information collection. A wireless sensor network is comprised of a large number of tiny wireless sensor nodes that are capable of sensing the environment and communicating in an ad-hoc manner to deliver relevant information to the user. The small form factor of these nodes limits the battery life available for their operation. Sensor networks are generally deployed in inhospitable terrains and consist of a dense deployment of sensor nodes. Furthermore, wireless communication is inherently unpredictable and error-prone. Hence, it becomes essential to employ a MAC protocol that efficiently shares the wireless channel. For many real-life applications of sensor networks, guaranteed delivery of data is very important. A critical event detected by the sensor network should be delivered to the user as soon as possible. Thus, for sensor networks, reliability and latency are important design parameters, in addition to energy efficiency.

In this paper, we present a novel medium access control protocol that ensures reliability in contention-based and error-prone channel conditions. The rest of the paper is organized as follows. Section II discusses related work on enforcing reliability at MAC layer in sensor networks. Section III presents MAC related issues for providing reliable data communication. Section IV describes our proposed protocol, on-demand *reliable medium access control* (*RMAC*). Section V compares the performance of the proposed protocol with existing mechanisms for providing link layer reliability. Section VI concludes the paper highlighting its contributions and our future work.

2. Related Work

The basic requirement of a sensor network is reliable delivery of data with minimum latency and energy consumption. There are two basic methods for achieving reliable data communication viz. forward error correction (FEC) and retransmissions. Since a sensor node typically has low processing power and a small memory, only FEC schemes having relatively low complexity may be used. Furthermore, since packets are transmitted intermittently, encoding schemes that need to work with multiple packets at a time can result in an unacceptable increase in latency. Consequently, most of the existing schemes use retransmissions or transmission of multiple copies of packets to achieve reliable transmission of information between remote nodes over multiple hops, despite channel errors, collisions and congestion.

The HHR (Hop-by-Hop Reliability) and HHRA (Hop-by-Hop Reliability with Ack) schemes [1], rely on sending multiple copies of the same packet. The required number of copies is determined from a locally estimated packet error rate, the desired packet delivery ratio and the hop-distance to the sink. The HHB (Hop-by-Hop Broadcast) and HHBA (Hop-by-Hop Broadcast with Ack) schemes [1] further reduce the

This work has been supported in part by the Ohio Board of Regents, Doctoral Investment Funds and National Science Foundation Grant CCR-0113361

number of copies retransmitted by exploiting the broadcast property of the wireless medium, since a broadcast packet is considered successful if any of the sender's neighbors (towards the destination) forwards the packet further. The authors have concluded that it is better to use an acknowledge-based scheme in sensor networks, since it does not perform too poorly in mild conditions but significantly better in testing conditions. ReInForM [2] is a similar scheme where multiple copies of the same packet are transmitted over randomly chosen routes. In all these schemes [1, 2], a collision-free environment has been considered by assuming the presence of a TDMA MAC layer.

Since, in a decentralized environment it is difficult to synchronize nodes and change slot assignments to adapt to topology changes, traditional TDMA systems suffer from scalability problems. Consequently contention-based on-demand MAC protocols have been the preferred choice for distributed infrastructure-less wireless sensor networks. Most of the contention-based MAC protocols proposed in the literature follow the operational model of CSMA that use a back-off mechanism to reduce probability of collisions. CSMA-based methods have a lower delay and better throughput at lower traffic loads and hence are well-suited for wireless sensor networks.

3. MAC Issues

In order to design a good reliable MAC protocol, the following medium access control issues need to be addressed:

- *Collision*: When a receiver node receives more than one packet at the same time, the packets are said to have collided, even if they coincide partially. All collided packets have to be discarded and hence retransmitted, thereby increasing the energy-consumption and end-to-end latency.
- *Congestion*: Congestion leads to packet loss due to buffer overflow. Dropping a packet at an intermediate node implies wasting the energy expended in forwarding the packet till that node. Hence some rate control mechanism should be enforced at the nodes to prevent congestion.
- *Channel error*: If the channel error is high, then the packets received at the physical layer may be corrupted requiring retransmission of the packet. This can be done by using positive (or negative) acknowledgements for each successful (or unsuccessful) reception or sending multiple copies of the same packet on the same or multiple routes.
- *Control overhead*: Since energy is consumed in sending and receiving control packets, minimal number of control packets should be used for data transmission.
- *Idle listening*: A node should avoid wasting energy listening to an idle channel while waiting for possible traffic.
- *Overhearing*: This problem also leads to energy waste when a node receives packets destined for other nodes.
- *Hidden node*: The hidden node problem exists between every other pair of nodes along the route from the source to the destination and depending on the data rate could lead to significant number of collisions. Request-to-Send (RTS) and Clear-to-Send (CTS) packets are usually used to mitigate the hidden node problem [3]. However for sensor networks where the packet size is small, a RTS-CTS-DATA-ACK handshake can constitute a large overhead [4].
- *Transmission rate control*: When the route is more than two hops long, the transmitting node needs to throttle its transmission rate so as to allow the second hop node to forward the packet. Otherwise, overlapping transmissions by the source node and the second hop node can lead to collision at the first hop node. This is due to hidden terminal problem and affects every node which is two or more hops away from the destination node.
- *Latency*: A packet may experience various delays at each hop of the network. Carrier sense delay is introduced when the sender performs carrier sense. Backoff delay happens when carrier sense fails, either because the node detects another transmission or because collision occurs. Transmission delay is determined by channel bandwidth, packet length and the coding scheme. Propagation delay is determined by the distance between the transmitting and receiving nodes. Processing delay happens when the receiver needs to process the packet before forwarding it to the next hop. Queuing delay depends on traffic load.

4. Proposed Protocol

RMAC is a CSMA/CA-based MAC protocol and hence has two important components viz. the listening mechanism and the backoff scheme. The listening mechanism ensures that the channel is available for transmission i.e. no other node is transmitting at that time. The backoff scheme is used to reduce contention. The idea of backoff is to restrain a node from accessing the channel for a random period of time hoping that the channel would become free after the backoff period. The basic operation of RMAC is shown in Fig. 1.

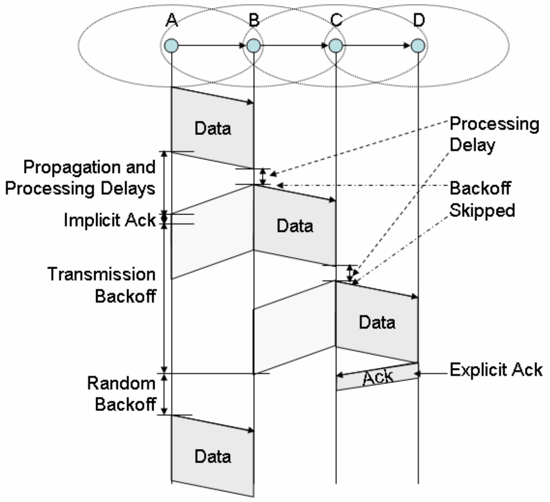


Figure 1. RMAC operation

Data forwarding: A node wishing to transmit generates random backoff duration based on its contention window. If the channel becomes busy during the backoff duration, the node freezes its counter and waits till the channel is idle again. Once the channel is free, it again starts decrementing the backoff counter till it becomes zero. At the end of the backoff period, if the channel is idle, the node transmits, otherwise a new random backoff is calculated and the process continues. An intermediate forwarding node, on the other hand, *skips* the random backoff duration after successful reception of data. The forwarding node processes the data and then transmits it immediately if the channel is found idle. This reduces the overall backoff delay of the system. The random backoff is calculated as follows,

$$Random\ Backoff = Uniform[0,2^{CW} * slotTime], \quad (1)$$

where, *CW* is the contention window of the node and slot time. Thus, to safely implement *skip-backoff mechanism* we assume that minimum contention window, CW_{min}, is kept such that the average initial random backoff is larger than processing delay. The average initial random backoff is given by,

$$Average\ Initial\ Backoff = 2^{CWmin\ -1} * slotTime. \quad (2)$$

Enforcing reliability: RMAC uses both implicit and explicit acknowledgements. Every intermediate node skips backoff and immediately transmits a successfully received packet after processing, to the next hop on the route. Taking advantage of the broadcast property of the wireless medium, this transmission can be considered to be an implicit Ack by the previous hop node. For example in Fig. 1, node A transmits a packet to node B. If node B receives the packet successfully, it skips backoff and forwards the packet to the next hop node C.

Thus, if node A overhears the ongoing transmission from node B to node C after processing and propagation delays, node A considers it to be positive Ack. However, if node A does not receive the implicit Ack, it waits for a backoff period and retransmits the packet to node B. If the receiving node is the final destination of the packet, then it transmits an explicit Ack, indicating successful reception of the packet, as by node D in Fig. 1.

Adaptive retransmission: Instead of having a constant number of maximum retransmission attempts for any packet, RMAC adaptively increases the bound depending on channel conditions. In general, retransmission attempts are used to retransmit collided packets. However, in a situation with high channel error, a significant number of packets may be dropped due to corruption at the physical layer. Thus, depending on the BER of the previously received packets, RMAC dynamically increases the maximum retransmission attempts in order to ensure reliability. If p_e is the packet error rate at a node, then the number of retransmission attempts is given by,

$$Retransmission\ Attempts = Tx\_Attempts + \left\lfloor \frac{1}{1-p_e} \right\rfloor, \quad (3)$$

where, *Tx_Attempts* is the maximum number of retransmissions allowed.

Transmission rate control: RMAC mitigates the hidden node interference problem by enforcing a transmission rate control mechanism at every node. A transmitting node on receiving an implicit Ack for the last packet sent, refrains from transmitting for duration, *transmission backoff*, equal to twice the communication delay involved in data transmission from one node to its next hop node. Since we employ the skip-backoff mechanism, this communication delay includes only the transmission delay and the processing delays (we assume that propagation and switching delays are negligible). However, a node can receive during the transmission backoff.

Buffer management: We assume three fields in the MAC header of each data packet: *TTL*, *birthTimeStamp* and *currentTimeStamp*. TTL indicates the number of hops remaining to final destination. This number is decremented at every hop, a concept similar to *time-to-live* (*TTL*) used extensively in multihop wired networks. In general, time-to-live (TTL) value is used to discard packets that have been in the network for too long. The TTL value is set by the source of the packet and is passed to the MAC layer by the routing layer using a routing protocol or GPS mechanism. The *birthTimeStamp* is the timestamp when the packet was generated. Initially, both the timestamps are the same, set by the source node. Each transmitting node adds to the *currentTimeStamp*, the time spent by the packet waiting in the queue. On the other hand, a receiving node adds to it the time taken by the packet to transmit over one hop, which is one communication duration. Depending on these parameters, a node decides whether the packet is *dead* or *aging*[1]. In either case packet is dropped from the queue and is not transmitted further. By discarding the aging packets, a node buffer can be made available for more recently generated packets (that generally better characterize the current environmental conditions), thereby reducing the collective end-to-end latency of the system. Also, by discarding an aging packet that would eventually be discarded by the system, the energy spent in forwarding the packet can be saved and buffer at intermediate nodes can be made available for newer packets.

5. Performance Evaluation

We have used PARSEC [5] to simulate and compare the performance of RMAC with that of the CSMA with Ack [3], multipacket and multipath forwarding schemes [1, 2]. CSMA with Ack employs positive acknowledgment for each successfully received packet. Multipacket forwarding scheme transmits $\lceil 1/(1-p_e) \rceil$ copies of the same data packet, where p_e is the channel error. Multipath forwarding scheme assumes GPS capability at each node and data is forwarded to at most three nodes which are nearer to the destination. The important simulation parameters are listed in Table 1. Each simulation is run for 100 seconds and is averaged over 10 iterations. Packet arrival at source nodes is considered to be Poisson process. The channel error is depicted by the packet error rate and is varied from 0 to 0.6. This is the probability with which a successfully received packet will be dropped by the receiver considering the packet as corrupted due to channel error.

We study the performance of RMAC with respect to three evaluation metrics. *Packet delivery ratio (PDR)* is the ratio of the total number of packets successfully received at the destination to the total number of packets sent by the source nodes. PDR measures the reliability of the system. *Latency per hop* is the average time spent by a packet at every hop from its generation at the source node to its successful reception at the destination node. The *energy utilization* of each protocol is calculated as follows:

$$\frac{\text{Total energy spent by the network}}{(\text{Number of nodes})\times(\text{Packets received at destination})}$$

Note that total energy expended in the system remains same for all the protocols, since $P_{Sensing} = P_{Tx} = P_{Rx}$.

Table 1. Simulation Parameters

Parameter	Value
Packet size	50 bytes
Ack size	4 bytes
Transmission rate	40 kbps
slotTime	100 μs
T_{proc}	10 μs
T_{SIFS}	20 μs
TTL	30
packetLifeTime	0.3 s (30 packet durations)
CW_{min}	1
CW_{max}	4
Tx_Attempts	5
$P_{Sensing} = P_{Tx} = P_{Rx}$	30 mW
Buffer size	30 packets

5.1. Effect of channel error with increasing number of hops

Figs. 2-5 compare the attained PDR, latency and energy expenditure of our RMAC protocol with that of the CSMA with Ack, multipath and multipacket schemes for increasing number of hops and different channel errors. RMAC-1, CSMA-ACK-1, Multipath-1 and MultiPacket-1 depict the performance for data arrival rate of 1 packet/second while RMAC-5, CSMA-ACK-5, MultiPath-5 and MultiPacket-5 depict that of 5 packets/second.

Fig. 2 shows the results for an error-free channel. At a low data rate of 1 packet/second, all four schemes perform very well with a PDR of 1 and minimum latency. The CSMA-ACK scheme exhibits slightly more latency due to the delay involved in transmitting explicit Acks. However, for a data rate of 5 packets/sec, the PDR decreases in the other protocols because of the increase in the number of collisions due to hidden node interference. The RMAC protocol remains unaffected due to the transmission rate control mechanism. The increase in collisions also results in increased latency of the CSMA-Ack scheme. Note that the energy utilization for the data rate of 5 packets/sec is less than that of 1 packet/sec. Higher data rates imply more packets are generated by the network, resulting in better energy utilization. For an error-free channel, the energy utilization is similar for all the schemes except the multipath scheme where more nodes are involved in data transmission.

Fig. 3 shows the results for a channel with packet error rate of 0.2. Even for data rate 1 packet/sec, the PDR of the multipath and multipacket schemes drop significantly as the hop-distance between source and destination increases. The multipath scheme has the least latency, but the latency of the multipacket scheme increases considerably because more number of

[1] $\text{packetLifeTime} + \text{birthTimeStamp} < \begin{cases} \text{currentTimeStamp; Dead Packet} \\ \text{currentTimeStamp} + \text{TTL} * \text{communicationDuration; Aging Packet} \end{cases}$

packets needs to be transmitted at each hop. Note that for the packet error rates of 0 and 0.2, the RMAC protocol maintains a PDR of 1 even with increasing number of hops. This proves the scalability of the protocol, which is a vital issue for sensor networks.

As shown in Figs. 4-5, even at low data rates, the PDR for the multipath and broadcast schemes drops drastically as the number of hops increases. Hence, we can conclude that the multipacket and multipath schemes are not suitable for providing reliability in sensor networks where the source and destination may be a separated by large number of hops and the channel is inherently error-prone. Consequently, in the remainder of this section, we compare the performance of the RMAC protocol with CSMA-ACK for various scenarios.

5.2. Multiple Hop Performance of RMAC vs. CSMA-ACK

Fig. 6 shows the effect of traffic load on the performance of RMAC and CSMA-ACK protocols under various channel conditions. RMAC-0.0 and CSMA-ACK-0.0 depict performance for error-free channel, RMAC-0.2 and CSMA-ACK-0.2 depict that for a channel with packet error rate of 0.2, and so on. Here, the source and destination is assumed to be separated by 8 hops. Since from the analysis in Section 5.1, it is evident that the RMAC protocol is more energy-efficient, we have not included the results for energy utilization in this section. Instead, we compare the retransmission attempts per packet for both the schemes. Apart from channel error, retransmissions are required for contention-related problems. Hence, a scheme that requires lesser retransmissions has a better collision-avoidance mechanism and is more energy-efficient.

Fig. 6(a) emphasizes the reliability of RMAC protocol, which for all channel error values provides a much higher PDR than the CSMA-ACK protocol. . In fact, for packet error rates of 0 and 0.2, RMAC protocol maintains a constant PDR of 1, in spite of increasing traffic load. Even for a very high packet error rate of 0.6, irrespective of the data rate, RMAC protocol delivers successfully more than double the number of packets delivered by the CSMA-ACK scheme. This is due to transmission rate control and adaptive retransmissions employed by the transmitting nodes in RMAC. As is evident from Fig. 6(b), the latency of RMAC protocol is also significantly lesser than the CSMA-ACK protocol, despite the delay introduced by its transmission rate control mechanism. This is because RMAC minimizes the backoff delay and also requires lesser retransmissions for successful delivery of packets, as is evident from Fig. 6(c). The latency of both schemes is the same for packet error rate of 0.6, since in this case the CSMA-ACK scheme is unable to deliver half the number of packets delivered by RMAC. For packet error rates of 0, 0.2 and 0.4, RMAC requires considerably lesser retransmissions, even though it has a much better PDR. This can be mainly attributed to the transmission rate control mechanism that curbs the possibility of collisions. For a very high packet error rate of 0.6, RMAC uses slightly more retransmission attempts than CSMA-ACK to achieve better reliability, thereby delivering double the number of packets as shown in Fig. 6(a).

5.3. Multiple Flow Performance of RMAC vs. CSMA-ACK

Figs. 7-8 compare the performance of RMAC and CSMA-ACK in the presence of a bottleneck node. Fig. 7 presents the results for the case where a pair of 2-hop routes has the same intermediate node (topology 1). Fig. 8 shows the results for the case where two routes each of 8 hops, have the center most node as common in both the routes (topology 2). In both the cases, RMAC shows better PDR and latency than CSMA-ACK. Also from Figs. 7(a) and 8(a), it can be observed that the PDR of RMAC does not reduce significantly for increasing hops. This reiterates the scalability of RMAC protocol, even in the presence of a bottleneck node.

However, surprisingly the PDR of CSMA-ACK is better for the 8-hop topology than the 2-hop topology. The reason is as follows. In case of 2-hop topology, both source nodes contend to transmit to the next hop common node, resulting in large number of collisions at the bottleneck node which decreases the PDR. This also leads to increased per hop latency and increased retransmission attempts by the source node as observed from Figs. 7(b) and 7(c), respectively. However, in case of the 8-hop topology, due to the hidden node interference problem, collisions occur at the second node of each route. These collisions inadvertently introduce a transmission rate control mechanism, which reduces collisions at the bottleneck node leading to low per hop latencies for both protocols as shown in Fig 8(b). Note that this transmission rate control mechanism comes at the cost of increased collisions and hence is certainly not energy-efficient. RMAC, on the other hand, employs a proactive transmission rate control mechanism at every node, thereby avoiding collisions. This leads to reduced retransmission attempts as shown in Fig. 8(c) and thus saves energy.

6. Conclusions

In this paper, we have designed an on-demand MAC protocol for sensor network applications that require packets to be delivered with high reliability and low latency. The proposed protocol has specialized mechanisms for ensuring better reliability and latency viz. skip backoff, implicit Ack, transmission rate control, cross-layer optimization for TTL and adaptive retransmission attempts. Using extensive simulations, we have compared the performance of our protocol with standard Ack-based, multipacket and multipath forwarding schemes. Our simulation studies reveal that our protocol exhibits better packet delivery ratio, latency and energy efficiency under various channel and traffic conditions.

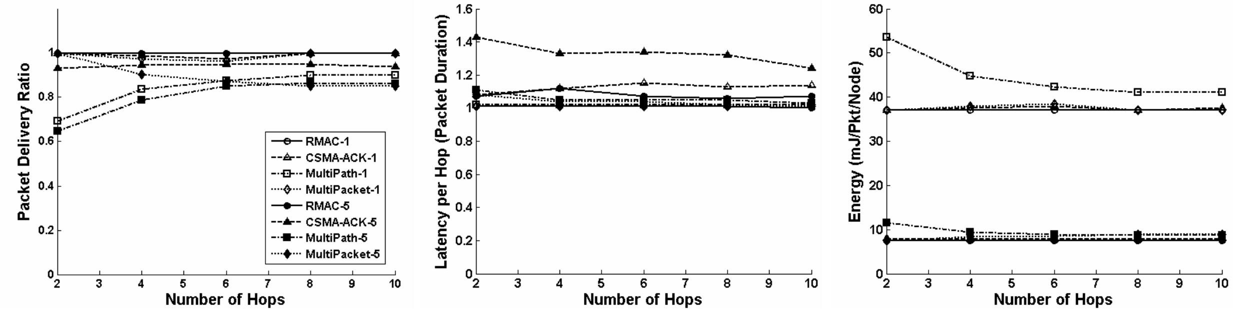


Figure 2. Performance of protocols with channel error as 0.0

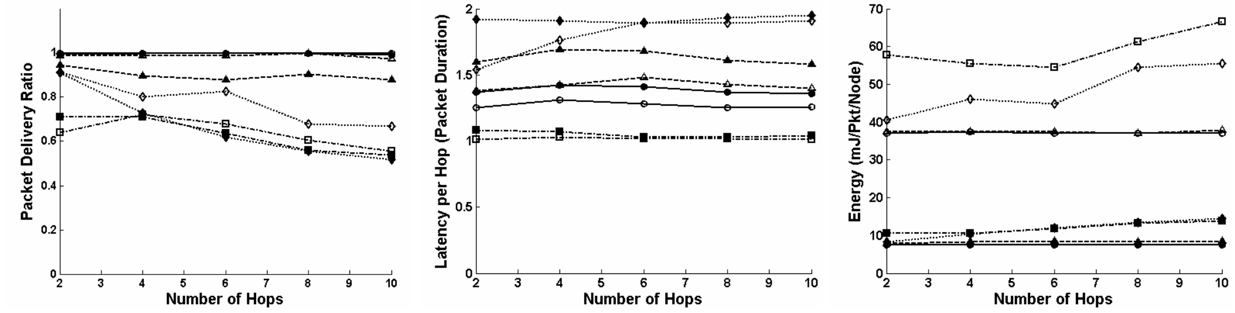


Figure 3. Performance of protocols with channel error as 0.2

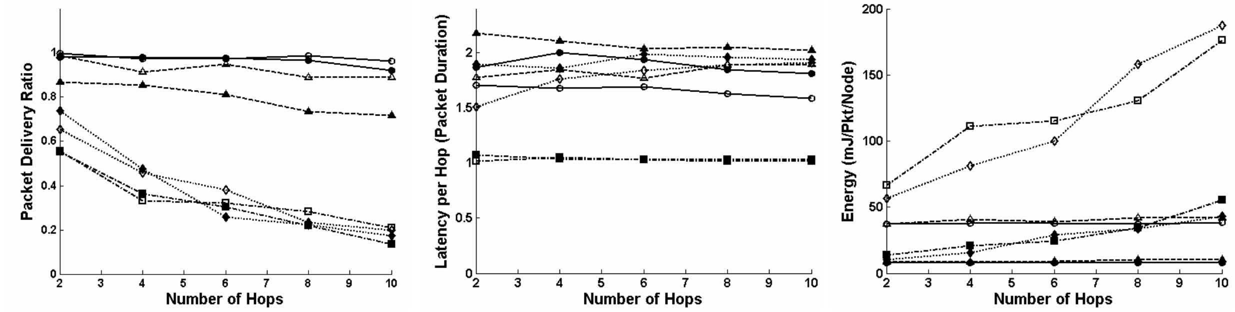


Figure 4. Performance of protocols with channel error as 0.4

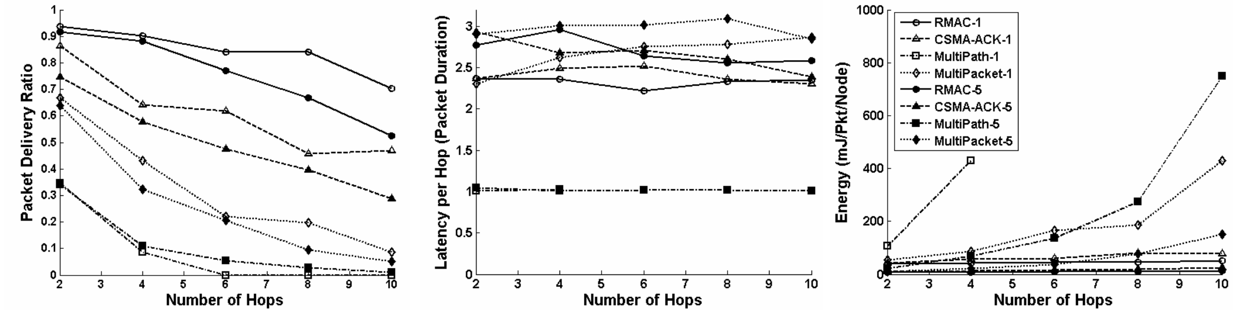
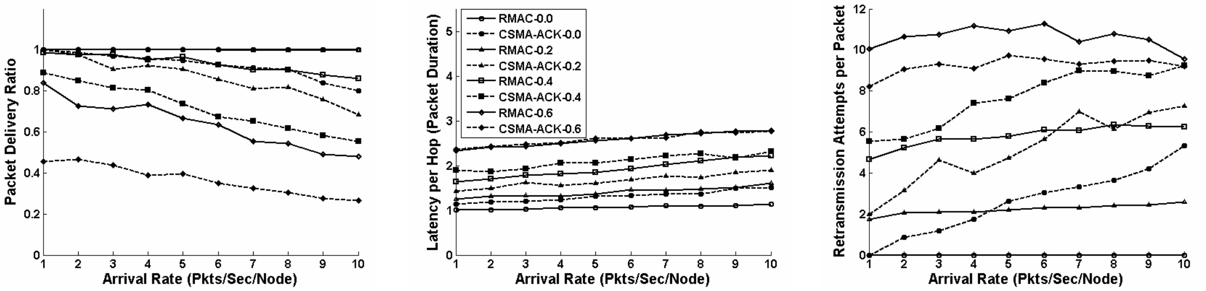


Figure 5. Performance of protocols with channel error as 0.6

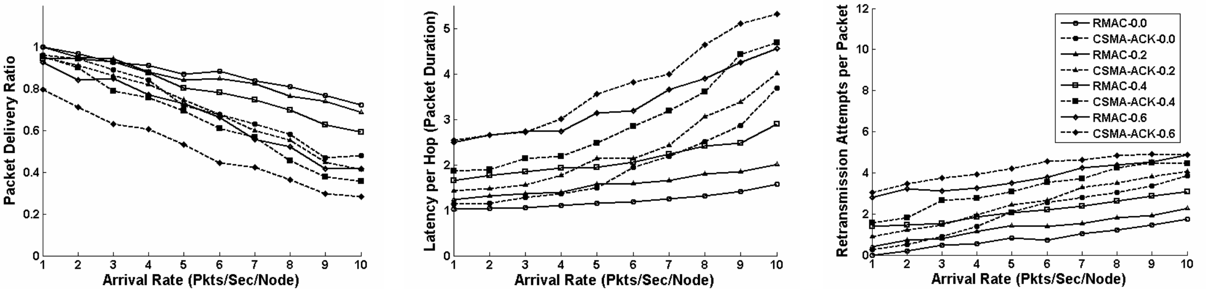


(a) PDR vs. arrival rate of packets at source node

(b) Latency per hop vs. arrival rate of packets at source node

(c) Retransmission attempts per received packet vs. arrival rate of packets at source node

Figure 6. Performance of RMAC and CSMA-ACK with channel error varying from 0.0 to 0.6

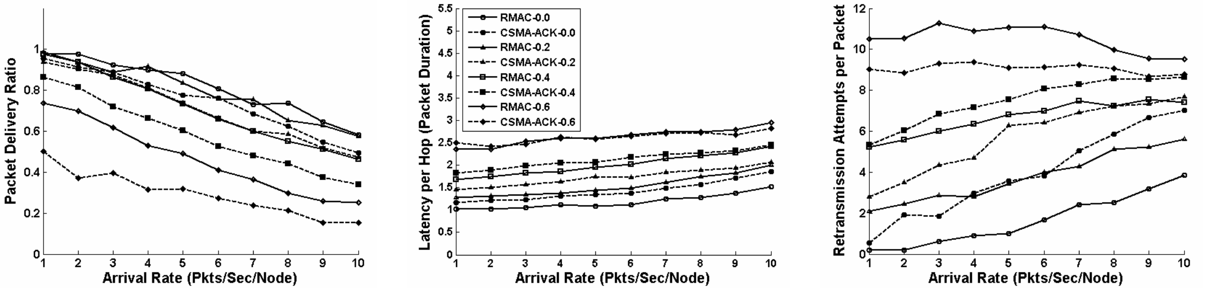


(a) PDR vs. arrival rate of packets at source node

(b) Latency per hop vs. arrival rate of packets at source node

(c) Retransmission attempts per received packet vs. arrival rate of packets at source node

Figure 7. RMAC vs. CSMA-ACK for topology 1 with channel error varying from 0.0 to 0.6



(a) PDR vs. arrival rate of packets at source node

(b) Latency per hop vs. arrival rate of packets at source node

(c) Retransmission attempts per received packet vs. arrival rate of packets at source node

Figure 8. RMAC vs. CSMA-ACK for topology 2 with channel error varying from 0.0 to 0.6

RMAC protocol can be made energy-efficient by preventing energy waste due to idle listening and overhearing. Two techniques have been explored to minimize idle listening periods viz. adaptive duty cycling [6] and wakeup on demand [7]. The integration of such energy saving mechanisms with RMAC is under investigation and is left as future work.

7. References

[1] B. Deb, S. Bhatnagar, and B. Nath, "Information Assurance in Sensor Networks," in *Proc .of 2nd ACM WSNA*, Sept. 2003.

[2] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: Reliable Information Forwarding using Multiple Paths in Sensor Networks," in *Proc. of IEEE LCN*, Oct. 2003.

[3] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part II--The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," in *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1417 – 1433, Dec 1975.

[4] A. Woo and D. E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," in *Proc. of ACM MobiCom*, July 2001.

[5] R. A. Meyer and R. Bagrodia, PARSEC Simulation Language, [online] http://pcl.cs.ucla.edu/projects/parsec.

[6] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. of IEEE Infocom*, June 2002.

[7] M. J. Miller and N. H. Vaidya, "A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio," in *IEEE Trans. on Mobile Computing*, vol. 4, no. 3, pp. 228-242, May-June 2005.