

Providing MAC QoS for multimedia traffic in 802.11e based multi-hop ad hoc wireless networks [☆]

T. Bheemarjuna Reddy ^a, John P. John ^b, C. Siva Ram Murthy ^{a,*}

^a *Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai 600 036, India*

^b *Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195, United States*

Received 26 July 2005; received in revised form 11 December 2005; accepted 11 April 2006

Available online 22 May 2006

Responsible Editor: M. Buddhikot

Abstract

Ad hoc wireless networks with their widespread deployment, now need to support applications that generate multimedia and real-time traffic. Video, audio, real-time voice over IP, and other multimedia applications require the network to provide guarantees on the Quality of Service (QoS) of the connection. The 802.11e Medium Access Control (MAC) protocol was proposed with the aim of providing QoS support at the MAC layer. The 802.11e performs well in wireless LANs due to the presence of Access Points (APs), but in ad hoc networks, especially multi-hop ones, it is still incapable of supporting multimedia traffic.

One of the most important QoS parameters for multimedia and real-time traffic is *delay*. Our primary goal is to reduce the end-to-end delay, thereby improving the *Packet Delivery Ratio* of multimedia traffic, that is, the proportion of packets that reach the destination within the deadline, in 802.11e based multi-hop ad hoc wireless networks.

Our contribution is threefold: first we propose dynamic ReAllocative Priority (ReAP) scheme, wherein the priorities of packets in the MAC queues are not fixed, but keep changing dynamically. We use the laxity and the hop length information to decide the priority of the packet. ReAP improves the PDR by over 28% in comparison with 802.11e, especially under heavy loads. Second, we introduce Adaptive-TXOP (A-TXOP), where transmission opportunity (TXOP) is the time interval during which a node has the right to initiate transmissions. This scheme reduces the delay of video traffic by reducing the number of channel accesses required to transmit large video frames. It involves modifying the TXOP interval dynamically based on the packets in the queue, so that fragments of the same packet are sent in the same TXOP interval. A-TXOP is implemented over ReAP to further improve the performance of video traffic. ReAP with A-TXOP helps in reducing the delay of video traffic by over 27% and further improves the quality of video in comparison with ReAP without A-TXOP. Finally, we have TXOP-sharing, which is aimed at reducing the delay of voice traffic. It involves using the TXOP to transmit to multiple receivers, in order to utilize the TXOP interval fully. It reduces the number of contentions to the channel and thereby reduces the delay of voice traffic by over 14%. A-TXOP is implemented over ReAP to further improve

[☆] This work was supported by the iNautix Technologies India Private Limited, Chennai, India and the Department of Science and Technology, New Delhi, India.

* Corresponding author.

E-mail addresses: arjun@cs.iitm.ernet.in (T. Bheemarjuna Reddy), johnjp@u.washington.edu (J.P. John), murthy@iitm.ac.in (C.S.R. Murthy).

the performance of voice traffic. The three schemes (ReAP, A-TXOP, and TXOP-sharing) work together to improve the performance of multimedia traffic in 802.11e based multi-hop ad hoc wireless networks.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Multimedia traffic; QoS; MAC; Ad hoc wireless networks; 802.11e; Voice; Video

1. Introduction

An ad hoc wireless network is a collection of mobile nodes that can communicate with each other over radio in the absence of any infrastructure. If two nodes lie within the transmission range of each other, then they can communicate directly. Two nodes that cannot directly communicate, can do so in a multi-hop manner in which the other intermediate nodes function as routers. Such networks are used in military applications and in emergency situations as they permit the establishment of a communication network at very short notice with a very low cost. However, these networks are limited by constraints in their bandwidth and power consumption.

With their widespread deployment, ad hoc wireless networks now need to support applications that generate real-time traffic. Applications such as voice communication, video-on-demand, video conferencing, and radio broadcasting require the network to provide guarantees on the Quality of Service (QoS) of the connection. One of the most important QoS parameters for multimedia and real-time traffic is *delay*. Multimedia traffic is delay-sensitive and live audio-visual communication requires that the end-to-end delay be less than a certain value. Thus, each packet has to reach the destination within the specified deadline, after which it becomes useless.

In a single-hop network, every node is within the range of every other node and hence, the queuing delay is not too large. However, in the multi-hop scenario, the source and destination may be several hops away, and packets need to be forwarded by the intermediate nodes. As a result, delays can become quite large, especially due to packets of a multi-hop flow contending with each other for the channel at successive hops, thereby making audio and video transmissions infeasible.

In this work, we consider the problem of QoS provisioning in a CSMA-based multi-hop ad hoc wireless network, where the QoS constraint on the flows is that of delay. Our focus is the *Packet Delivery Ratio (PDR)*, which is a measure of the percentage of packets that reach the destination within the specified deadline. The flows originating in the net-

work are either voice or video flows. Other types of flows, which are not delay-sensitive are not considered here (since our focus is on multimedia traffic only). In a wireless LAN, the QoS guarantees can be provided by the Access Point (AP) since it acts as a central coordinator. However, in ad hoc networks, the absence of a central coordinator and the multi-hop nature of the network, wherein packets have to be forwarded over multiple broadcast regions, make the provisioning of end-to-end QoS guarantees a very challenging problem.

The following sub-section provides a brief introduction to the working of 802.11e, followed by an introduction to our work.

1.1. The IEEE 802.11e

The upcoming 802.11e standard [1] enhances the current 802.11 Medium Access Control (MAC) [2] to support applications with QoS requirements. It provides a channel access function, called Hybrid Coordination Function (HCF). The HCF uses both contention-based channel access method, called Enhanced Distributed Channel Access (EDCA) mechanism for contention based data transfer, and centrally controlled channel access, referred to as HCF Controlled Channel Access (HCCA) mechanism, for contention free data transfer. The HCF is usable only in infrastructure-based wireless networks, where it uses a QoS-aware point coordinator (typically co-located with a QoS AP), called Hybrid Coordinator (HC), for coordinating access to the channel. However, since we do not have any infrastructure in ad hoc wireless networks, only EDCA mechanism can be used while contending for the channel.

The EDCA provides differentiated and distributed access to the wireless medium. It works with four Access Categories (ACs), where each AC achieves a differentiated channel access (see Fig. 1). This differentiation is achieved by varying the amount of time a node would sense the channel to be idle and the length of the contention window during a backoff. Each frame from the higher layer carries its user priority (UP). The EDCA supports eight

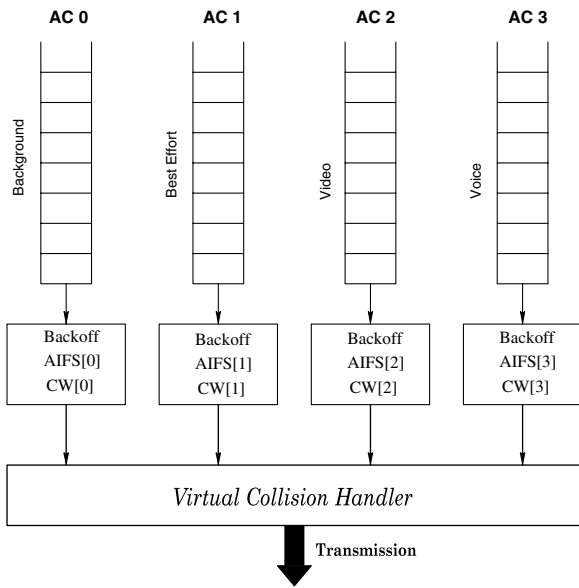


Fig. 1. Original 802.11e with four ACs.

Table 1

User priority to access category mappings

User priority (UP)	Access category (AC)	Designation
1	0	Background
2	0	Background
0	1	Best effort
3	1	Best effort
4	2	Video
5	2	Video
6	3	Voice
7	3	Voice

different UPs. After receiving a frame, the MAC layer maps it into one of the four ACs, shown in Table 1. Each AC has a set of access parameters, such as initial contention window size (CW_{min}), maximum contention window size (CW_{max}), and arbitration inter-frame space ($AIFS$). Flows that fall under the same AC are effectively given identical priority to access the channel. If one AC has a smaller $AIFS$, CW_{min} , or CW_{max} , the ACs traffic has a better chance to access the channel earlier. Each AC contends for transmission opportunities (TXOPs) using a set of EDCA access parameters that are unique to the AC of the packet to be transmitted. On obtaining the TXOP, a node can send one or more packets present in its queue.

The TXOP is defined as an interval of time during which a node has the right to initiate transmissions. It is characterized by a starting time and a

maximum duration called $TXOP_{Limit}$. Depending on the duration of TXOP, a node may transmit one or more frames. If a frame is too large to be transmitted in a TXOP (that is the case with video frames), it should be fragmented into smaller frames. Within a node, if there is more than one AC finishing the backoff at the same time, the highest priority AC frame is chosen to transmit by the virtual collision handler.

1.2. Our work

In this work, we introduce a scheme for enhancing the performance of multimedia traffic in 802.11e based multi-hop ad hoc wireless networks. Our scheme has the advantage that it requires minimal modifications to the existing 802.11e MAC protocol. Our improvements are based on the observation that a purely local rescheduling of packets, can significantly increase the overall PDR. The inherent broadcast nature of the wireless network is another feature we exploit. Our scheme has three parts to it. First, we have ReAP, wherein packets within an AC are prioritized based on their laxity as well as the number of hops they have to traverse to reach the destination. Second, we introduce the concept of A-TXOP, which dynamically modifies the TXOP limit for video packets based on the packets currently in the queue. Finally, we devise a feature called TXOP-sharing, which uses the broadcast nature of the network to reduce the control overhead and also the overall delay, for voice packets. We compare the performance of our proposed scheme with the existing 802.11e MAC protocol, through a series of exhaustive simulations.

The rest of this paper is organized as follows. Section 2 briefly explains the related work in this area and Section 3 describes the ReAP scheme. In Section 4, we discuss A-TXOP and in Section 5 we present TXOP-sharing. The simulation scenarios and results are described within each section. Section 6 briefly explains the relationship among the three schemes. Finally, Section 7 concludes the paper.

2. Related work

In wireless networks with APs, recent works have shown how best to achieve QoS guarantees on fairness and delay [3,4]. However, in ad hoc wireless networks, there are more challenges, such as, lack of central coordination, constraints on the information exchange between the nodes, and contention

among packets of a multi-hop flow for accessing the channel, in successive hops. The performance analysis of 802.11e EDCA is presented in [5–7].

Most of the work dealing with MAC QoS is based on changing the inter-frame spacing (IFS) and the contention window size. The authors of [8] showed that a slow decrease of the contention windows (instead of decreasing to CW_{\min}) after each successful packet transmission reduced the collision rate and hence the number of retransmissions. The authors of [9] use Adaptive EDCF (A-EDCF), wherein the contention window of each traffic class is adapted according to the estimated collision rate in order to improve the goodput of the traffic under heavy loads. The authors of [10] tried to improve the throughput by replacing the exponential backoff mechanism by an adaptive one. To provide service differentiation, the authors of [11] proposed the usage of different contention window sizes, inter-frame spacing, and maximum frame size for services of different priorities.

The authors of [12] proposed a global data parameter control scheme for 802.11e WLANs. In the proposed scheme, the AP dynamically controls the parameters for best-effort traffic based on traffic conditions. Voice and video traffic are subject to a centrally assisted distributed admission control scheme, wherein nodes listen to available budgets from the AP to make decisions on acceptance or rejection of a voice or video stream. The authors of [13] proposed modification of the CW and the AIFS based on the instantaneous network conditions. The network conditions are inferred from the collision rate and the packet drops at the network layer queue. However, it is discussed only in the context of a single-hop network.

Some of the existing proposals, such as [14,15] considered the delay requirement while assigning priorities to the packets. They assume a single-hop network and hence consider only the delay requirement (in terms of laxity). For example, [14] assumes sensor network is divided into several cells (nodes in each cell form a single-hop network) and intra-cell packets are exchanged inside each cell using earliest deadline first (EDF). The authors of [15] employed EDF for maximizing network throughput in wireless LANs. However in multi-hop ad hoc networks, since packets have to traverse several hops, in addition to laxity, hop count also needs to be considered while assigning priorities to the packets. The authors of [16] proposed distributed priority scheduling for end-to-end QoS guarantees in ad hoc net-

works. In their work, the priority of the head of the line (HOL) packet is broadcast to all the one-hop neighbors and the backoff mechanism is modified based on the priority information. They also proposed multi-hop coordination, wherein the downstream nodes adjust the priority levels based on the delays experienced by the packets upstream. However, their scheme requires each node to maintain the state information for all its one-hop neighbors. The authors of [17] used a priority reallocation mechanism for improving overall throughput. In their work, the priorities of flows were changed dynamically with the aim of maintaining an equal number of packets in the queues of all the ACs. However, some of the high priority flows end up starving when their priority is reduced to that of non-real time flows. Also, it is not extensible to multi-hop networks. In our work, we try to achieve improvements in PDR and delay in multi-hop ad hoc networks, while not deviating much from the 802.11e MAC protocol.

3. Dynamic ReAP

In this section, we describe a scheme for dynamically modifying the priority of a packet, with the goal of increasing the proportion of packets that reach their destinations within the specified deadline. While the 802.11e MAC protocol does provide service differentiation (with the aid of multiple ACs), its performance in a multi-hop ad hoc network is quite unsatisfactory while carrying multimedia traffic. Packets of all flows are considered equal, irrespective of the number of hops a packet has to travel. Therefore, a packet whose destination is one hop away has the same chance of capturing the channel as does a packet whose destination is six hops away.

As a result, we find that most packets which have only a few hops to traverse reach the destination well within the deadline, with time to spare. On the other hand, packets which have a destination several hops away, rarely meet the deadline. This not only reduces the PDR of the network, but also results in wasted bandwidth, since these delayed packets are no longer useful. The reason a k -hop flow performs worse than a 1-hop flow is quite obvious: the k -hop flow has to endure the queuing delay, the channel access delay, the transmission delay and the propagation delay k times over, at the source and also at each intermediate node on the path to the destination. Since the packets which have to tra-

verse many hops suffer larger delays, we have to give them higher priority, while at the same time not starving other packets. Hence in multi-hop ad hoc networks, laxity and hop count need to be considered while assigning priorities to the packets.

Since the propagation and transmission delays are fixed, we have to optimize on the queueing and channel access delays. We are constrained by the fact that the proposed scheme should entail minimal modifications to the existing 802.11e MAC protocol, that is, no sending of additional control messages and minimum changes to the frame formats. 802.11e EDCA has four queues at the MAC layer, one for each AC (Fig. 1). Upon receiving a packet from network layer, the MAC layer stores several attributes related to that packet while inserting it into one of its four queues. Some of these attributes are the transmission rate to be used for sending this packet, a flag indicating whether it needs to be encrypted or not, the encryption key if it should be encrypted, and the kind of preamble that needs to precede this packet. Since adding some more attributes will have a negligible impact on the memory consumption, we modify the queues at the MAC layer for AC2 and AC3 (video and voice, respectively). As shown in Fig. 2, we allow the storage of certain additional attributes for keeping track of the priority of the packet. The relevant attributes are the deadline and the number of hops remaining.

The deadline is calculated as the sum of the laxity of the packet and the current local time of the node. The reason behind storing deadline, instead of laxity, as an attribute of the packet in the MAC queue and the laxity updation mechanism in a multi-hop scenario are explained later in this section.

In order to know the hop count for a packet, we need support from the routing protocol. Our scheme uses Dynamic Source Routing (DSR) [18] as the routing protocol, since the number of hops left to the destination is stored as a part of the DSR header in the field named *segsLeft*. When the routing module passes a packet to a MAC queue (either AC2 or AC3), it also provides the value of the number of hops remaining for that packet. We then have ReAP schedulers for both these queues which decide which packet has to be transmitted when the channel is captured. The network layer control packets (such as route requests, route replies, and route errors) have the highest priority. They are placed in the queue for AC3 and are sent ahead of all other packets in the queue.

The channel access mechanism is identical to the 802.11e protocol, and is left unchanged. Each AC competes for the channel, and after resolution of any virtual collisions that may occur, the channel is captured by one of the ACs. Instead of transmitting packets in the FIFO order (as in the case of 802.11e) or EDF order (as in the case of [14,15]),

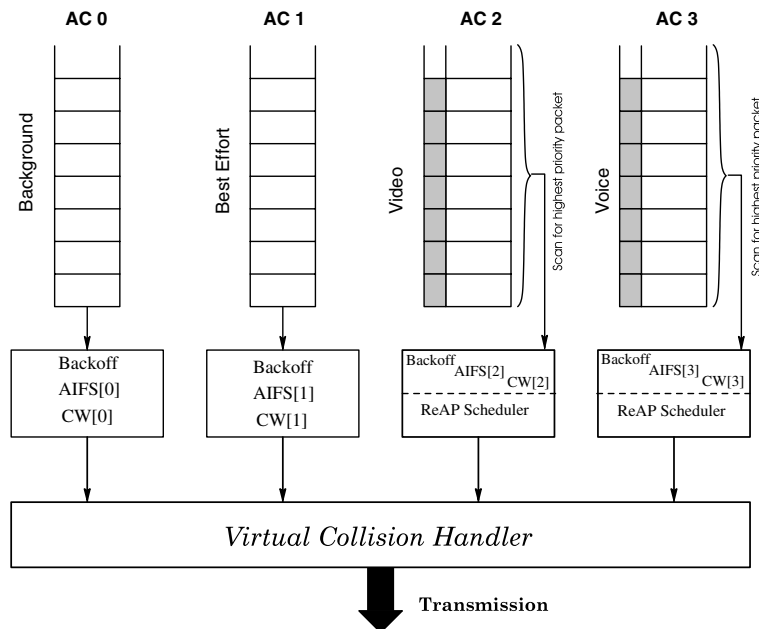


Fig. 2. ReAP: the modified MAC queues and the ReAP scheduler.

we prioritize the packets with respect to the *laxity* and the *hop count*. We use the measure of (*laxity*/*hopsLeft*) to decide the priority of the packet. This value gives us a rough estimate of how much delay the packet can tolerate at each hop. Hence, the packet with the lowest value of (*laxity*/*hopsLeft*) is given the highest priority. If two packets have the same lowest value of (*laxity*/*hopsLeft*), we resolve the conflict by sending the packet which has more hops to travel. Using the value of the deadline present in the attribute field of the MAC queue, the laxity $l = (\text{deadline} - \text{currentLocalTime})$ is calculated. If the laxity value becomes negative, the packet is immediately removed from the queue and dropped as it is useless at the destination. This approach reduces the network's load and benefits already queued packets with lower priority to reach their destinations. The number of hops left (*hopsLeft*) is also obtained from the attribute field. The ReAP scheduler then computes the priority index $p = \frac{l}{\text{hopsLeft}}$ for each packet present in the MAC queue, and selects the packet with the least value of p . The selected packet is then transmitted over the channel. Note that the priority is computed at the last moment (just before transmission) so as to utilize the latest information. This is required since the relative priorities of packets within queues is not fixed but keeps varying with time. For example, consider two packets *A* and *B* in the queue.

$\text{laxity}_A = 80$ ms and $\text{laxity}_B = 50$ ms $\text{hopsLeft}_A = 4$ and $\text{hopsLeft}_B = 2$. Therefore: $p_A = \frac{\text{laxity}_A}{\text{hopsLeft}_A} = 20$ and $p_B = \frac{\text{laxity}_B}{\text{hopsLeft}_B} = 25$, meaning packet *A* has higher priority. Now, suppose the node captures the channel after 30 ms. At this instant the laxities for packets *A* and *B* are: $\text{laxity}_A = 50$ ms and $\text{laxity}_B = 20$ ms, and their priorities are: $p_A = 12.5$ and $p_B = 10$.

Hence the priorities have reversed and packet *B* now has a higher priority. As a result, it is possible that a packet that just enters the queue, but having a tight deadline to meet, will be sent immediately, ahead of all the other packets that were waiting in the queue.

The advantage of this scheme is that it is a completely distributed scheme and requires no central coordination or exchange of control messages between nodes.

3.1. Laxity updation mechanism

In order to determine the priority of the packet, ReAP scheme requires the availability of laxity of

the packet at the MAC layer. The value of the laxity parameter can be determined as the difference between deadline¹ of the packet and the current local time of the node. However, the exploitation of the deadline parameter requires that all communicating nodes within the multi-hop ad hoc network are synchronized with each other, which cannot be considered realistic given the precision required for the MAC layer operation. Thus we straight away work with the laxity parameter and update its value whenever the packet experiences some delay on its way from the multimedia source to the destination. We account for the queuing delay at each node, and the propagation and transmission delays at each hop along the path. For simplicity, we assume that processing delay is negligible. Our laxity updation mechanism works as follows.

The source node's application layer, along with each multimedia packet (voice or video frame), supplies the corresponding laxity parameter to the lower layers in the protocol stack. The laxity parameter is invalid for other kinds of traffic. Since User Datagram Protocol (UDP) is typically used at the transport layer for carrying multimedia traffic [19,20], we also employed the same in our work. When the network layer receives a multimedia packet from the higher layer, it checks its routing table for finding a path to the destination. If the path is available, the routing module immediately passes the packet to the MAC layer without incurring any connection setup delay. In this case, the value of laxity parameter is left unchanged and supplied as it is to the MAC layer. Otherwise, the routing module puts a timestamp on that packet and initiates the route discovery process. On finding a path to the destination, the routing module of the source node calculates the elapsed time of the packet as follows: $\text{elapsedTime} = \text{currentLocalTime} - \text{timeStamp}$. Then it updates the value of laxity parameter ($\text{laxity} = \text{laxity} - \text{elapsedTime}$) and passes the modified laxity parameter to the MAC layer.

As we mentioned earlier, the MAC layer queues for AC2 and AC3 contain an additional attribute for storing the deadline of the packet. The reason behind storing the deadline, instead of the laxity parameter supplied by the network layer, is to reduce the memory space required for implementing these queues. If we store the laxity parameter as it is

¹ The deadline parameter is set by the application layer of the multimedia source node. $\text{deadline} = \text{laxity} + \text{currentLocalTime}$, where laxity is set to 150 (400) ms for voice (video) packets.

an attribute of the packet in the queue, we need one more attribute (i.e., timestamp) for calculating the elapsed time of the packet in the queue. In our scheme, when the MAC layer receives the laxity parameter for a packet from the network layer, it calculates the deadline (deadline = laxity + currentLocalTime) and stores that one as an attribute of that packet in the queue. When an AC captures the channel, its ReAP scheduler determines which one of the pending packets has the highest priority at that time for transmitting over the channel. Using the value of the deadline attribute, the ReAP scheduler dynamically calculates the laxity, $\text{laxity} = (\text{deadline} - \text{currentLocalTime})$, and uses this value along with the number of hops remaining attribute for determining the priorities of packets in its queue.

In order to provide the value of laxity parameter to the downstream node along the path, the DATA frame header of the 802.11e protocol has to be modified to include a field of length 4 bytes for storing this value. It is to be noted that the laxity field is an optional field and presents only in the multimedia data frames. For accurately accounting the queuing delay experienced by the packet, the value of laxity field is calculated at the last moment (i.e., after receiving CTS frame from the downstream node) by using the corresponding deadline attribute from the queue and then the multimedia frame is transmitted over the channel. Upon receiving the multimedia frame, the downstream node estimates the transmission delay of that frame as follows:

$$\text{transmission_delay} = \frac{\text{frame_size}}{\text{channel_capacity}} + \text{synchronization_time}. \quad (1)$$

According to the 802.11e MAC specifications, the synchronization time is 192 micro seconds [21] and propagation delay is 1 μ s. After estimating the transmission delay, the MAC layer of downstream node extracts the value of laxity parameter from the received frame and updates that as follows: $\text{laxity} = \text{laxity} - \text{transmission_delay} - \text{propagation_delay}$. Along with a multimedia packet, the MAC layer supplies the corresponding laxity parameter to the network layer. The laxity parameter is invalid for other kinds of packets. If it is an intermediate node, the network layer passes the packet and its laxity parameter to the MAC layer for forwarding that packet to the downstream node along the path. Otherwise, it passes the packet and its laxity parameter to the application layer of the multimedia des-

tinuation node. In this manner, the value of laxity parameter is updated dynamically along the path in multi-hop ad hoc networks.

3.2. Simulation studies

In order to see how this scheme would work in realistic conditions, extensive simulations are carried out. The Glomosim network simulator [22] was used, over which the 802.11e MAC protocol and the ReAP schemes are implemented. The simulations are performed for multi-hop ad hoc networks, and the nodes are not mobile. The reason we do not consider mobility is because mobility causes path breaks and then the performance depends on the routing protocol employed. Our study of the MAC layer is independent of the routing protocol. The parameters of the simulation are specified in Table 2. The 802.11e-specific parameters, such as CW_{\min} , CW_{\max} , $AIFS$, and $TXOP_{\text{Limit}}$ use the default values as mentioned in Ref. [1].

The topology of the network and the flows are both randomly generated. We have an equal number of voice and video flows. Simulation runs are carried out for various seeds and the results are discussed in the following section. Each simulation run is for a duration of 15 min and all the flows are present throughout the duration of the simulation. The PDR is calculated as the ratio of number of packets received within the deadline by the application layer of the destination, to the number of packets sent by the application layer at the source node. Each simulation run has flows with random source–destination pairs and thus different hop lengths. The PDR value for k -hop flows (in one simulation run) is computed as the average PDR value of all the flows with k hops in that run. All the simulation results presented in this paper conform to 95% confidence levels.

Table 2
Parameters used in the simulation

Parameter	Value
Number of nodes	75
Terrain area	1000 m \times 1000 m
Transmission range	282 m
Channel capacity	11 Mbps
Routing protocol	DSR
<i>Data rate</i>	
Voice	64 kbps
Video	\approx 128 kbps
Number of seeds	20

3.3. Simulation results

From Figs. 3–5, it can be clearly observed that as the load increases, the PDR decreases due to congestion. At low loads (20 flows), there is no observable difference between the two schemes. This is because there is no congestion, and also there will be very few nodes that forward packets belonging to multiple flows. For a single flow, the packets are sent in FIFO order and thus is equivalent to

802.11e. As the hop length increases, the PDR decreases, as expected. However, the performance of ReAP does not degrade as drastically as that of 802.11e. As the hop length increases, the performance gain of ReAP over 802.11e increases. This fact agrees with the reasoning that packets with larger hop lengths to traverse get a higher priority in ReAP, thereby enabling them to reach the destination within the deadline. The difference between the two schemes is quite significant as is reflected

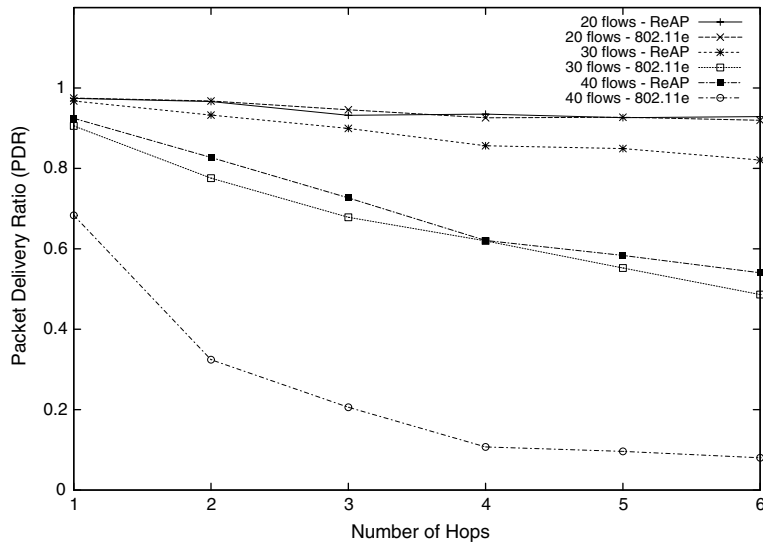


Fig. 3. Variation of PDR vs hop count for video traffic.

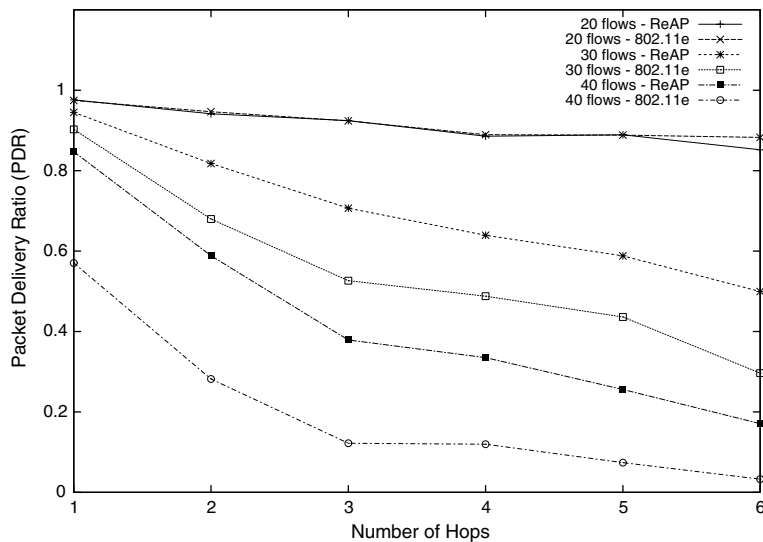


Fig. 4. Variation of PDR vs hop count for voice traffic.

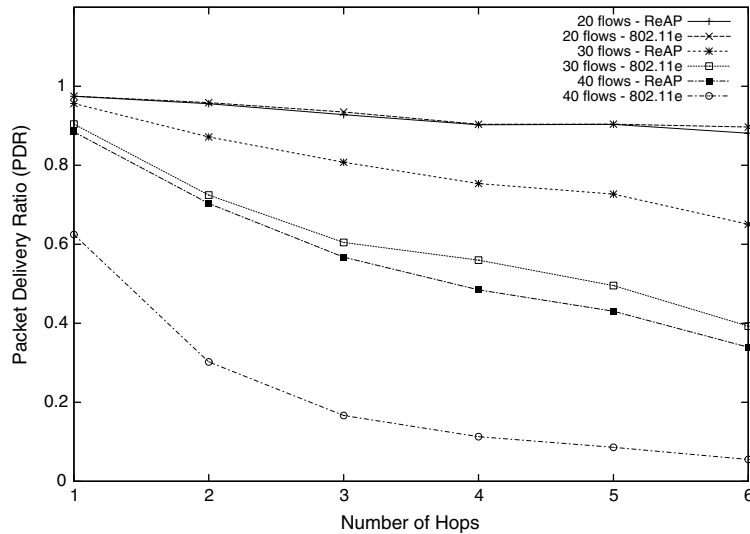


Fig. 5. Variation of PDR vs hop count for combined video and voice traffic.

by the fact that ReAP with 40 flows performs as effectively as 802.11e with 30 flows. ReAP improves the average PDR by about 28% over 802.11e for moderately heavy loads (30 flows).

4. Adaptive TXOP

Our second contribution is the introduction of a feature which we call Adaptive-TXOP (A-TXOP). This is aimed at reducing the delay and hence improving the PDR of video traffic over ad hoc networks.

In a wireless LAN, the Access Point (AP) decides duration of the TXOP for each node, based on the offered load. Since the AP dynamically varies the TXOP duration, the network is able to attain the QoS targets. However, in the ad hoc case, the TXOP limits are fixed for each AC and hence not flexible [1,23]. The insight behind adaptive-TXOP is closely linked with the observed inherent traffic characteristics of video traffic.

Video data is usually voluminous and hence it is compressed before transmitting over the network. There are many compression schemes for video such as MPEG and H.264. All these use *Motion Compensation Prediction* to exploit the temporal redundancy in a video stream. Compressed video typically has a variable bit rate (VBR), that is, all frames are not of the same size. Also, all frames are not of the same importance. There are typically three types of frames in a compressed video stream: I-frames, P-frames, and B-frames. Of these, I-frames are the most impor-

tant frames, and also the largest. Each I-frame is associated with a set of nine P-frames and B-frames. This set of 10 frames is called a Group of Pictures (GOP). Note that the length of the GOP may vary depending on the nature of the video stream and the compression technique employed. The I-frame is the independent picture frame, which provides a starting point and also a synchronization point for recovery after errors. The I-frame provides the base picture and the other P and B frames in the GOP are enhancements to it. If the I-frame is lost or delayed beyond its deadline, all the other frames in its GOP are useless and have to be discarded. Hence, it is important to ensure that the I-frames reach within their deadlines.

Multimedia applications can tolerate the loss of a few packets. In addition, they prefer a steady data rate rather than the bursty data rate associated with window-based network protocols. Hence they typically use RTP/UDP rather than TCP at the transport layer [19,20,24]. Unlike TCP, UDP does not fragment the video packets even if they exceed the underlying network's maximum packet size.² UDP simply hands down each video frame to the network layer. At the network layer, the MTU is set to be 2324 bytes, which is the maximum MAC Service Data Unit (MSDU – the packet delivered to the

² The Maximum Transfer Unit (MTU) is the maximum size of a packet sent to the network card by the protocol stack. The link layer is responsible for discovering this MTU and reporting it to the protocols above the link layer.

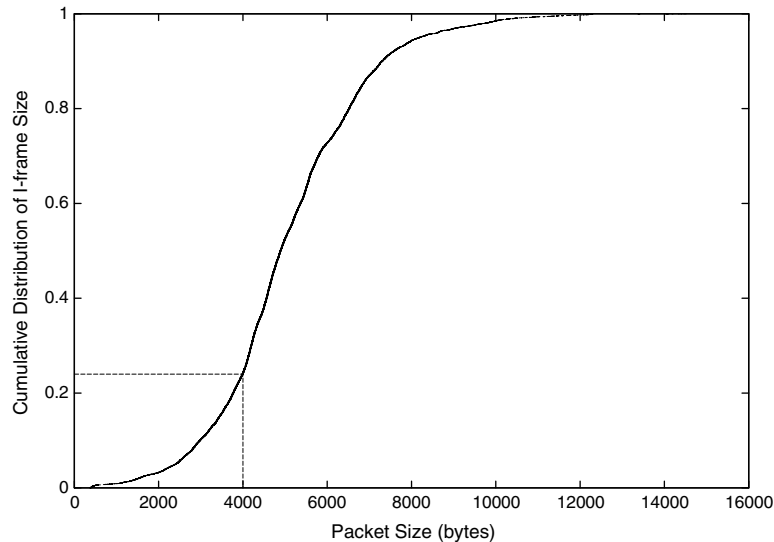


Fig. 6. The cumulative distribution of packet sizes of I-frames, for the trace available at [25].

MAC layer by the higher layer) size for the 802.11e. Because video frames are typically larger than this threshold, the network layer (i.e., IP) divides them into fragments and passes down to the MAC layer.

The analysis of several video traces showed that while the average video frame size was around 2000 bytes, almost all the I-frames were larger than this. From Fig. 6, it can be seen that over 70% of the I-frames are larger than 4648 bytes, meaning they have to be split into three or more fragments. Fragmented frames have a greater disadvantage than smaller, unfragmented frames. This is because all the fragments have the same deadline (since they belong to the same frame), but require multiple channel accesses to transmit them, and hence suffer larger delays. While the introduction of TXOP mitigates this, the TXOP duration is found to be insufficient for most of the I-frames. This can be seen from the following data.

The TXOP limit is 6.016 ms for AC2 (video traffic) [1]. Taking into account the synchronization delays and the transmission delays for the RTS, CTS, DATA, and ACK packets, we observe that only three DATA frames of size 2360 bytes (the maximum size of the packet delivered to the physical layer by the 802.11e) each can be sent in a single TXOP interval. As a result, a node will require multiple accesses to the channel to send an I-frame. In Fig. 7, we have a 2-hop flow with source S , I as the intermediate node, and D as the destination. For the purpose of illustration, we have assumed

that only two DATA frames can be sent in one TXOP interval. S has an I-frame (or some other large video frame) in its queue, which is split into three fragments: P , Q , and R . The following is a possible sequence of steps that occurs in the transmission of the I-frame from S to D and is shown in Fig. 7.

- Step 1:* Node S captures the channel and transmits fragments P and Q in its TXOP to node I .
- Step 2:* Node I captures the channel and forwards P and Q in its TXOP to the destination node D .
- Step 3:* Node S now sends the last fragment R in its next TXOP to the intermediate node I .
- Step 4:* Node I forwards R in its next TXOP to the destination node D .
- Step 5:* Node D now has all the fragments. They are reassembled here to get the complete video frame.

Thus, in the case when we have a static TXOP limit, as in 802.11e, we require four channel accesses, with S and I each accessing the channel twice. What we propose is instead of having a fixed TXOP limit, the duration of the TXOP should be adaptive, depending on the video frames in the queue. In short, all the fragments of a video frame should be sent together in a single TXOP. This is what would happen in the above scenario, if we use A-TXOP, and this is shown in Fig. 8.

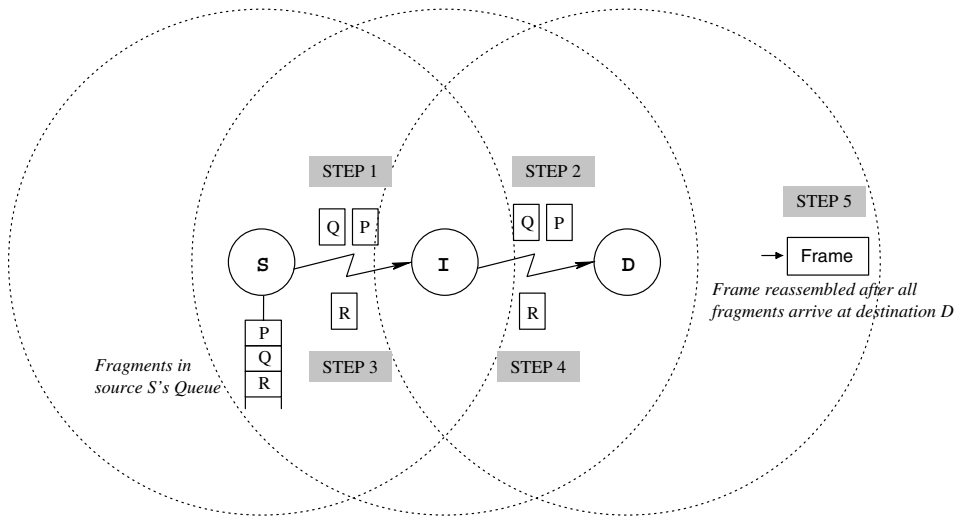


Fig. 7. An example showing transmission of a large video frame (split into three fragments *P*, *Q*, and *R*) without *A-TXOP*.

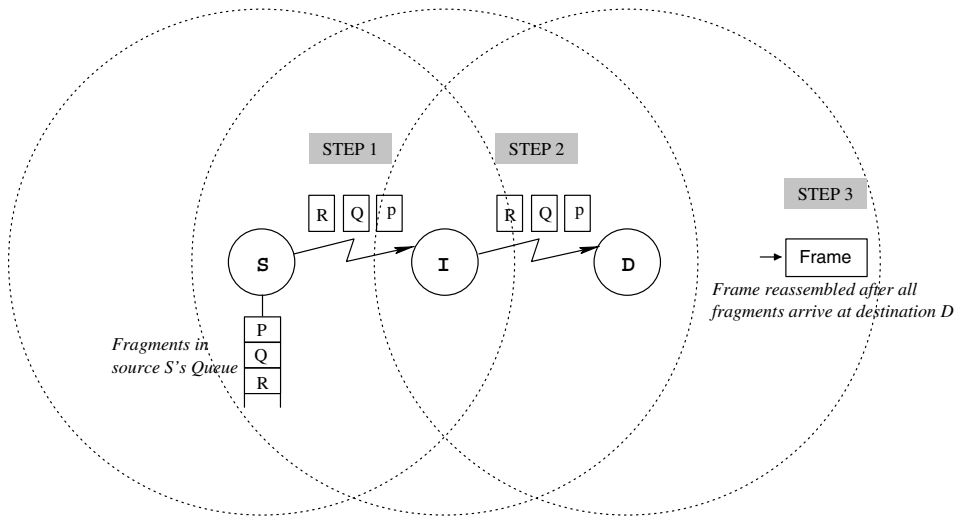


Fig. 8. An example showing transmission of a large video frame (split into three fragments *P*, *Q*, and *R*) with *A-TXOP*.

Step 1: Node *S* captures the channel and transmits fragments *P*, *Q*, and *R* (in its extended TXOP).

Step 2: Node *I* now forwards *P*, *Q*, and *R* (in its extended TXOP) to the destination node *D*.

Step 3: Node *D* now has all the fragments. They are reassembled here to get the complete video frame.

As we can see, this simple extension of the TXOP interval can lead to a significant reduction in the end-to-end delay even for a 2-hop flow. Here, we require only two channel accesses as opposed to

four accesses in the earlier case. The performance improvement will be a lot higher for flows spanning several hops. This improvement will greatly benefit the I-frames, since they are the ones which are most fragmented. As a result, the PDR of the I-frames would increase, and thereby increase the overall video quality.

However, for its operation A-TXOP requires that appropriate information in respect to video fragmentation is available to the MAC scheduler. This information is typically not available to the MAC layer, as video frames are fragmented at the network layer by the IP fragmentation module. In

order to provide such information to the MAC layer, we need support from the network layer protocol (i.e., IP). When the network layer passes a video fragment to the MAC queue of AC2, it extracts the values of three fields (`ip_typeOfService`, `ip_identification`, and `ip_sourceID`) from the IP header of corresponding frame and passes them to the MAC layer. We modify the queue for AC2 (video) to store these additional attributes for keeping track of the fragmentation details of video frames. This modification allows the MAC scheduler to transmit all fragments of the given video frame in a single channel attempt by adapting the TXOP interval appropriately.

A-TXOP can be integrated with either 802.11e or ReAP. Since in Section 3 we already showed that ReAP outperforms 802.11e, A-TXOP is implemented over ReAP (see Fig. 23) to further improve the performance of video traffic. This requires modifying the queue for AC2 to allow the storage of seven additional attributes for keeping track of the priority and the fragmentation details of video packets. In the following section, we show how *ReAP with A-TXOP* reduces the end-to-end delay and improves the quality of video traffic in comparison with *ReAP without A-TXOP*.

4.1. Simulation studies

We have said that A-TXOP is useful in reducing the channel access delay and hence the end-to-end delay as well. Note that A-TXOP is implemented

over ReAP. Through the following simulations, we have shown the improvement in delay obtained with the use of A-TXOP. We also see how the PDR of I-frames is affected by A-TXOP, and how the PDR varies with the number of fragments in a video frame. In addition to PDR, we also compute the Peak Signal to Noise Ratio (PSNR). PDR has the limitation that it treats all frames with the same importance. However, we have seen that I-frames are much more important than the P and B frames. In order to capture this difference in the levels of importance, we use PSNR for comparison. PSNR is the most widely used objective metric for comparing video streams. The PSNR for a frame in the video sequence is defined as

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{255^2}{\text{MSE}} \right),$$

where MSE is the mean square error between the frame that is sent at the source and the frame that is received at the destination. Note that PSNR is equivalent to MSE, except for the fact that it is measured in the log scale. We therefore plot the MSE values as they provide a more intuitive measure of the errors in the video stream. The simulations were run with 20 flows (10 voice and 10 video) in a multi-hop network.

4.2. Simulation results

From Fig. 9, it can be seen that the mean end-to-end delay increases with hop length for both *ReAP*

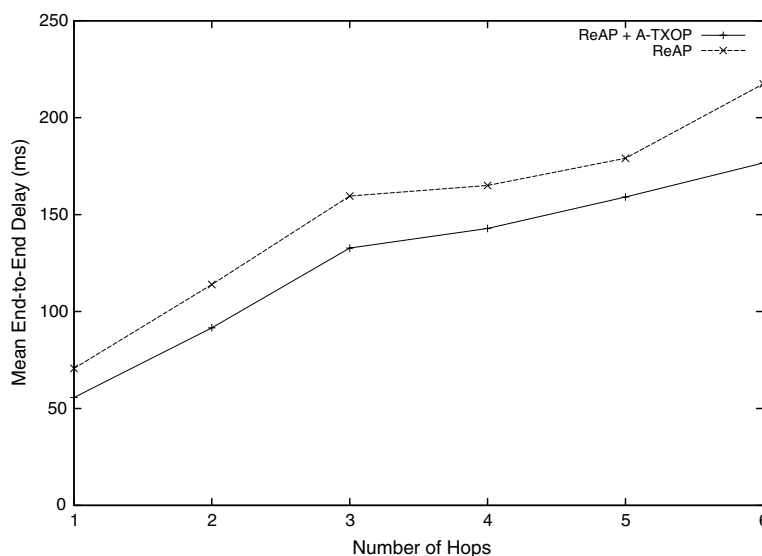


Fig. 9. Variation of end-to-end delay vs hop count for video traffic.

without A-TXOP and ReAP with A-TXOP. However, the delay with A-TXOP is consistently lower than the mean delay without A-TXOP. Also, the difference between the two schemes is more significant at higher hop lengths. This is because the total number of channel accesses is proportional to the hop length, and hence the delay increases more rapidly without A-TXOP.

The delay for voice traffic (from Fig. 10) is higher with A-TXOP than without. This is because A-

TXOP benefits only AC2, that is, video traffic. At the same time, the average duration for which AC2 captures the channel increases. This in turn increases the delay for other ACs and so voice traffic gets affected. However, the increase in delay is quite small and hence unlikely to cause problems.

Fig. 11 compares the PDR of I-frames for both cases: with and without A-TXOP. As expected, A-TXOP does benefit the I-frames, particularly the ones that have to traverse further (since more

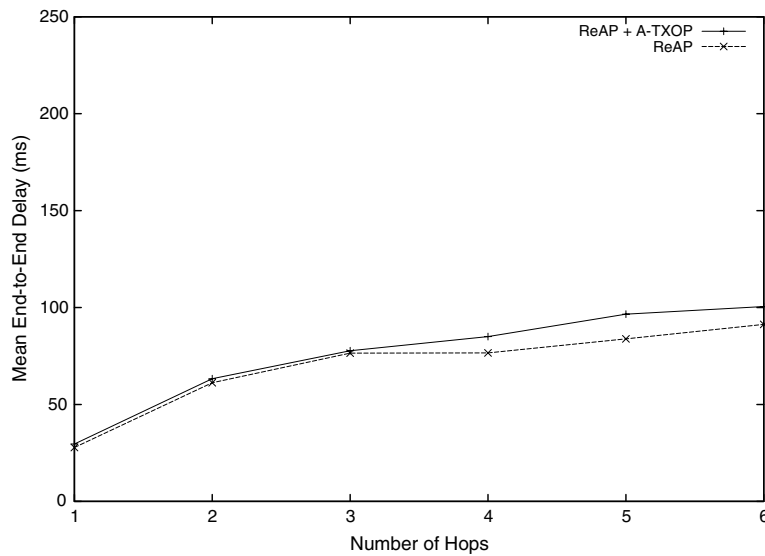


Fig. 10. Variation of end-to-end delay vs hop count for voice traffic.

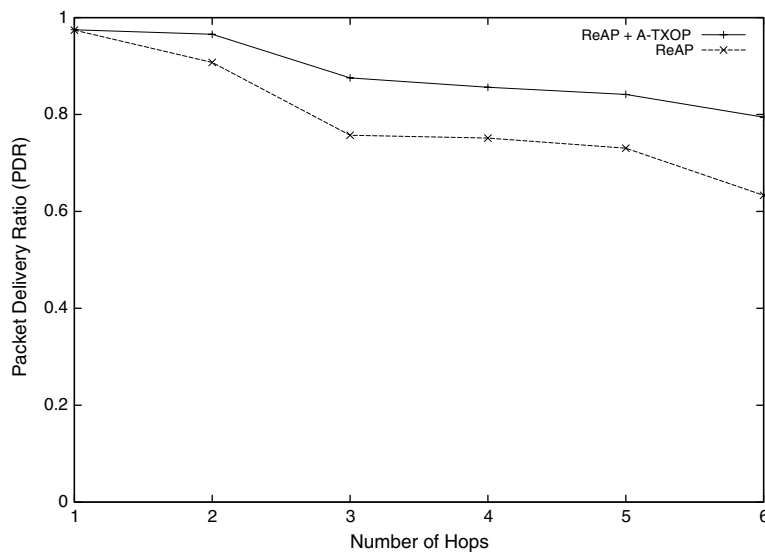


Fig. 11. Variation of PDR vs hop count for only the I-frames in the video stream.

fragments and longer hop lengths will benefit most from A-TXOP). The difference in PDR between the two schemes is not significant when the packet size is less, but A-TXOP performs much better than its counterpart for larger packets, as can be observed in Fig. 12. This is because A-TXOP only affects packets which have several fragments (i.e., larger packets), and hence, when packet sizes are small, it is equivalent to the case without A-TXOP. A-TXOP reduces the end-to-end delay of video

packets by 27%, and this improves the PDR by 10% when compared to the case without A-TXOP.

Fig. 13 shows the frame-by-frame MSE values with and without A-TXOP. It can be observed that there are more peaks (errors) without A-TXOP than with A-TXOP. As expected, the overall MSE is higher without A-TXOP. This is because without A-TXOP, the I-frames have a greater probability of missing the deadline, and as more I-frames are lost, the degradation in quality is more drastic. As

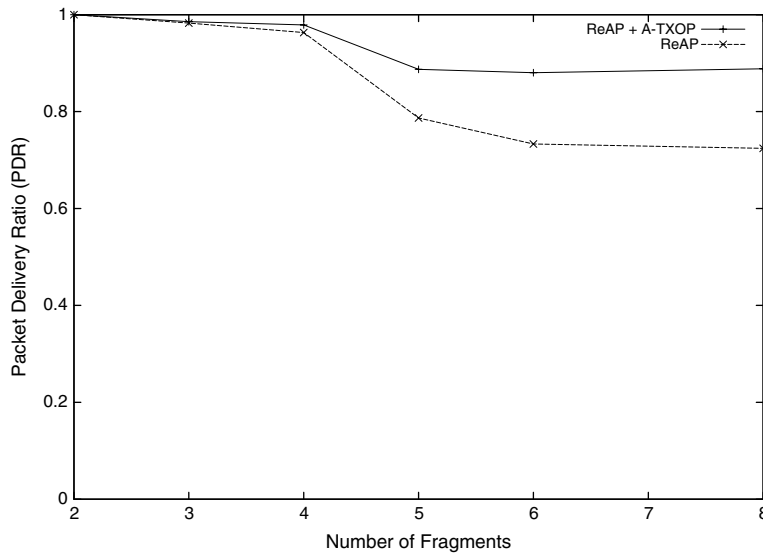


Fig. 12. Variation of PDR vs number of fragments for all frames in the video stream.

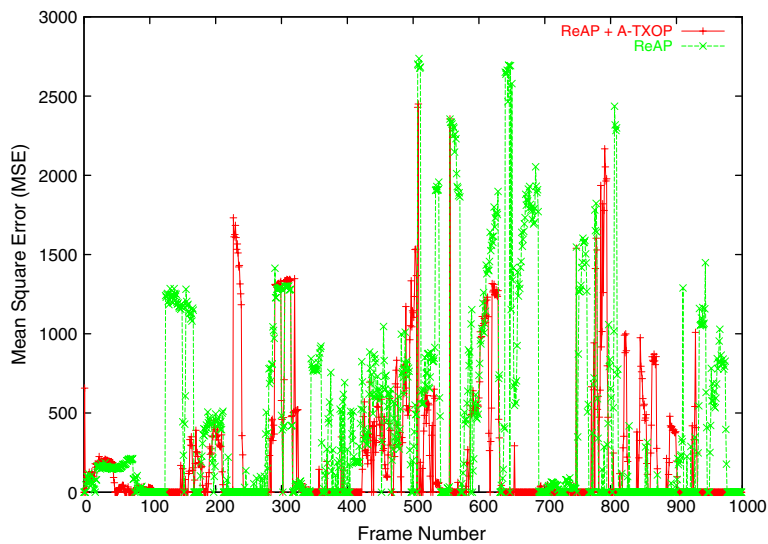


Fig. 13. Mean square error (MSE) values per frame for a 3-hop flow.

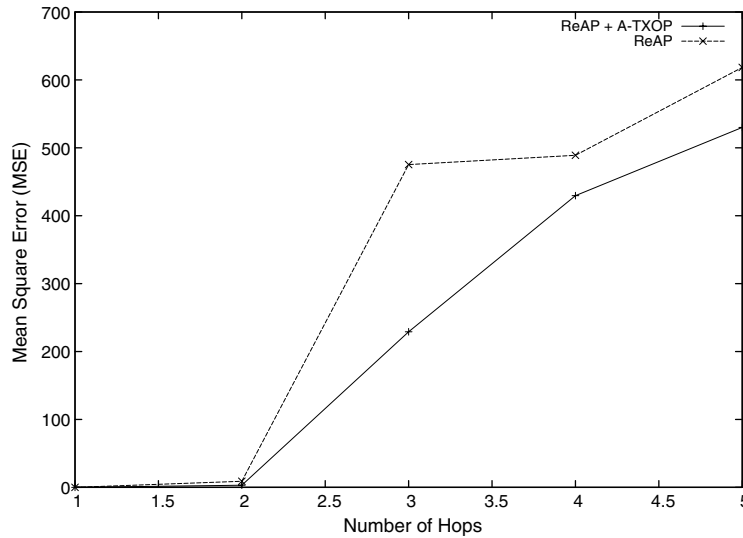


Fig. 14. Variation of MSE vs hop count.

the hop length increases, the quality decreases (MSE increases) as seen in Fig. 14.

4.3. Theoretical analysis

The end-to-end delay comprises the queuing delay, the channel access delay, the transmission delay, and the propagation delay. While ReAP increases the PDR by reducing the maximum lateness, A-TXOP attempts to reduce the channel access delay. The channel access delay is the amount of time the node has to wait before sending the packet, once the channel has been sensed idle i.e., it is the time during which the node contends for the channel. Observe that the total access delay experienced by a packet depends on the number of times the node needs to access the channel to send the complete packet, which in turn depends on the size of the packet (or equivalently the number of fragments in the packet). Hence we get

$$D_{\text{avg}} = \frac{D_{\text{access}} \times N_{\text{access}}}{N_{\text{packets}}}, \quad (2)$$

where D_{avg} is the average MAC delay experienced (due to all its channel accesses) by the packet, D_{access} is the average channel access delay, that is, the average delay to access the channel once, N_{access} is the number of channel accesses required to transmit all the packets, and N_{packets} is the total number of packets.

D_{access} depends on the number of nodes contending for the channel, the offered load, the number of

collisions, and also on the duration for which a node holds the channel. From Eq. (2), we can see that the MAC delay can be reduced by reducing the number of accesses to the channel, and this is the focus of A-TXOP.

We now analyze the performance of A-TXOP for a single flow. It is sufficient to consider the case of a single flow since we are focusing only on the number of channel accesses and that is independent of other flows and depends only on the packet sizes of this flow. For simplicity, the packet sizes are assumed to be integral multiples of the *maximum fragment size* and all fragments have the same size. We also assume a saturated network, that is, each node always has a packet to send and enough fragments ready to fully utilize the next TXOP interval.

Suppose a node can transmit upto k fragments in a TXOP interval. The TXOP can then be considered to have k slots, such that in each slot, a node can transmit exactly one fragment. Now, the number of accesses needed by a packet depends on the number of fragments in the packet and also on the slot in which the first fragment of this packet was transmitted. For example, if a packet has five fragments, and the TXOP has three slots, then if the first fragment is transmitted in Slot₃, then it will take two more channel accesses to get the packet through: three and one fragments, respectively in the next two TXOPs. Therefore, the node will require a total of three channel accesses to send the packet.

Now, to find the probability that the transmission of a packet starts in Slot _{i} , we consider a

Markov chain with k states. The state label i denotes the slot at which the first fragment of the next packet will be transmitted. The probability of transition from any state i to any state j (including itself) is

$$P_{ij} = \begin{cases} f_{j-i}, & \text{if } i \leq j, \\ f_{k+j-i}, & \text{if } i > j, \end{cases}$$

where

$$f_i = P(\text{Packet has } j \text{ fragments} \mid j \bmod k = i).$$

An illustration of the Markov chain for the case $k = 3$ is shown in Fig. 15. For the generic case, we have the following single step transition probability matrix

$$\mathbf{P} = \begin{bmatrix} f_0 & f_1 & f_2 & \cdots & f_{k-1} \\ f_{k-1} & f_0 & f_1 & \cdots & f_{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_2 & f_3 & f_4 & \cdots & f_1 \\ f_1 & f_2 & f_3 & \cdots & f_0 \end{bmatrix}.$$

The steady state probability matrix $\mathbf{\Pi} = [\pi_1 \ \pi_2 \ \cdots \ \pi_k]$ where π_i is the steady state probability that a packet will begin transmission in Slot $_i$, can be computed from following equation:

$$\mathbf{\Pi} \cdot \mathbf{P} = \mathbf{\Pi}. \quad (3)$$

Therefore, we get

$$\begin{aligned} [\pi_1 \ \pi_2 \ \cdots \ \pi_k] & \begin{bmatrix} f_0 & f_1 & \cdots & f_{k-1} \\ f_{k-1} & f_0 & \cdots & f_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ f_1 & f_2 & \cdots & f_0 \end{bmatrix} \\ & = [\pi_1 \ \pi_2 \ \cdots \ \pi_k]. \end{aligned} \quad (4)$$

It is easy to observe that $\pi_1 = \pi_2 = \cdots = \pi_k = \pi$ satisfies Eq. (4). This is because the matrix \mathbf{P} is cyclic and $\sum_0^{k-1} f_i = 1$.

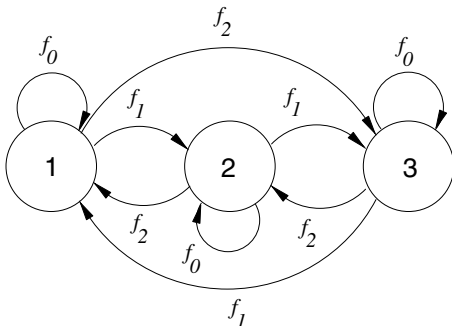


Fig. 15. An illustration of the Markov chain for $k = 3$.

Thus we get that the probability of starting transmission in a particular slot is independent of the packet size distribution and that all slots are equiprobable. The probability of starting transmission in the i th slot, given that there are k slots, is $\pi_i = \frac{1}{k}$.

Once π_i is known, we can compute the average number of channel accesses required by a packet. Given the starting slot for a transmission, and the number of fragments in the packet, the number of channel accesses is a deterministic quantity. We denote: N_{ij} to be the number of channel accesses required for a packet with j fragments, and starting transmission in the i th slot.

The average number of channel accesses required by a packet with j fragments will then be

$$C_j = \sum_{i=1}^k \pi_i \cdot N_{ij} \quad (5)$$

and the average number of channel accesses required by any packet (irrespective of number of fragments) will be

$$C_{\text{avg}} = \sum_j p_j \cdot C_j, \quad (6)$$

where p_j is the probability that the packet has j fragments. This can be obtained from the packet size distribution. N_{ij} can be computed as follows:

Consider a packet that has j fragments. Then we have

$$n \cdot k < j \leq (n + 1) \cdot k, \quad \text{where } n \text{ is an integer } \geq 0.$$

Since the k slots are all consecutive, the number of channel accesses required by the packet is at least $(n + 1)$ and at most $(n + 2)$. Denote

$$l = j - nk, \quad (7)$$

l represents the number of fragments left after filling n full TXOPs. Hence, if at least l fragments are sent (say s , where $s \geq l$) in the first channel access, then the remaining $j - s$ fragments can be sent in the subsequent n accesses to the channel (since $j - s \leq nk$). However, if less than l fragments are sent in the first access, then the node will require another $(n + 1)$ accesses to the channel. Thus we will need $(n + 2)$ accesses, if we start transmission in any of the last $l - 1$ slots of the TXOP. From this argument, we get the value of N_{ij} as

$$N_{ij} = \begin{cases} n + 1, & \text{if } 0 < i \leq (k + 1 - l), \\ n + 2, & \text{if } (k + 1 - l) < i \leq k. \end{cases}$$

Setting $\pi_i = \frac{1}{k}$ and substituting the values of N_{ij} in Eq. (5), we get

$$C_j = \frac{1}{k} [(n+1)(k+1-l) + (n+2)(l-1)] = \frac{j+k-1}{k} \quad [\text{using Eq. (7)}]. \quad (8)$$

Thus, from Eq. (6), we have

$$C_{\text{avg}} = \sum_j p_j \cdot \left(\frac{j+k-1}{k} \right) = \frac{1}{k} \sum_j j \cdot p_j + \left(\frac{k-1}{k} \right) \sum_j p_j = \frac{1}{k} (\bar{J}) + \left(\frac{k-1}{k} \right) = \frac{\bar{J}+k-1}{k}, \quad (9)$$

where \bar{J} is the average number of fragments in a packet.

Hence, we see that *without* A-TXOP, the average number of channel accesses per packet is $C_{\text{avg}}^{802.11e} = \frac{\bar{J}+k-1}{k}$. On the other hand, *with* A-TXOP, each packet requires exactly one channel access. Thus, the average number of accesses per packet will be $C_{\text{avg}}^{\text{A-TXOP}} = 1$. However, D_{access} is proportional to the duration for which a node holds the channel. Since k is the number of fragments sent in a TXOP for 802.11e, we have

$$D_{\text{access}}^{802.11e} \propto k, \\ D_{\text{access}}^{\text{A-TXOP}} \propto \bar{J}.$$

Hence,

$$D_{\text{avg}}^{802.11e} \propto \frac{\bar{J}+k-1}{k} \times k, \\ D_{\text{avg}}^{\text{A-TXOP}} \propto 1 \times \bar{J}.$$

From this we can see that though the channel access delay increases with \bar{J} for A-TXOP, the overall MAC delay is still lesser with A-TXOP. Another advantage of using A-TXOP is that as the number of channel accesses decreases, the contention for the channel decreases, and as a result, the number of collisions decrease and this in turn further reduces the channel access delay.

4.4. Validation

To validate the analysis, simulations were carried out with conditions similar to those in the analysis. There is a single flow in a single-hop network. The number of channel accesses required to send all the packets is measured as the number of RTS requests sent. The average number of channel accesses per packet is measured by dividing this value by the total number of packets sent. The input data is obtained from video trace files at [26] and is not synthetically generated. Different trace files are used to obtain different values of \bar{J} .

From Fig. 16, $C_{\text{avg}} = 1$ when A-TXOP is used, as expected. Without A-TXOP, C_{avg} is expected to increase linearly with \bar{J} and from the results, it can

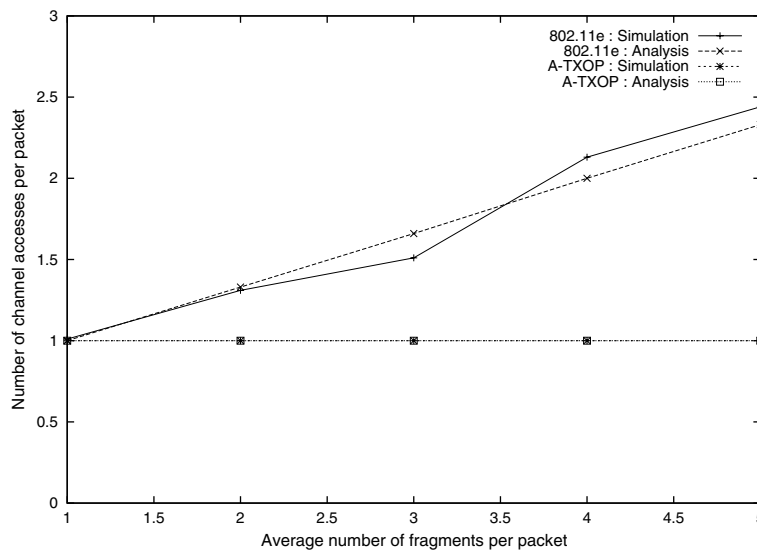


Fig. 16. Number of channel accesses vs Average number of fragments per video frame.

be seen that the curve obtained from the simulations tallies with the expected values.

When there is more than one flow, there is a possibility of collisions. When collisions occur, the total number of channel accesses increases (by a quantity equal to the number of collisions) to $N_{access} + N_{collision}$, where $N_{collision}$ is the number of RTS collisions. Thus we have seen that A-TXOP reduces the number of channel accesses as well as the number of collisions. This is the reason for the reduction attained in the overall end-to-end delay and improvement in the PDR of video traffic, as seen in the previous section (Figs. 9, 11 and 12).

5. TXOP-sharing

Our third contribution is again based on TXOP and is called TXOP-sharing. The objective is improving the performance (viz. reducing delay and improving throughput) of voice traffic over ad hoc networks.

As mentioned earlier, a TXOP is a contention-free interval wherein a node that captures the channel sends multiple packets with a single RTS–CTS exchange. The restriction here is that all the packets must be addressed to a single receiver, that is, if a node does not have enough packets for a particular receiver, it cannot effectively utilize the TXOP interval.

In a wireless LAN, this is not really a restriction, since all the nodes forward their packets to the AP, and hence have only a single receiver. However, in an ad hoc network, especially a multi-hop one, this is not really the case. A node has in its queue, packets for several nodes in its one-hop neighborhood. It may so happen that a node that has obtained the TXOP does not have enough packets (to a single receiver) to fully utilize the TXOP. However, it may have in its queue packets to other receivers which could also be sent in this TXOP, thereby increasing the utilization of the TXOP.

While this argument holds for both AC2 and AC3 (video and voice, respectively), this under-utilization is more prominent in case of voice traffic. This is due to the fact that voice packets are typically smaller in size, allowing for upto five packets to be transmitted in a single TXOP burst. Also, a single voice flow has quite a low bandwidth requirement (usually between 32 kbps and 64 kbps), and hence the number of packets to a single receiver present in the queue at any instant may not be sufficient to fully use the TXOP. In a multi-hop network, an intermediate node will have to forward packets of several flows and so will have packets to several receivers in its queue.

With TXOP-sharing, a node is no longer restricted to transmit to a single receiver in a TXOP burst, but can transmit packets to upto two receivers. This allows better utilization of the TXOP, and reduces the control overhead. In order to implement TXOP-sharing, the RTS header (Fig. 17) has to be modified to include the address of the second receiver (RA_2) to which the node intends to send the packet. This second receiver will also have to be informed as to how many packets will be sent to the first receiver (RA_1), so that it can send the CTS after that is done. Though we can allow sharing of the TXOP with more than two receivers, practical considerations make it infeasible. First and foremost, each additional receiver increases the size of the RTS frame. Next, maintaining coordination becomes difficult in the presence of errors and will result in a performance degradation.

Suppose the source node S has two packets to send to node A and one packet to send to node B . It first transmits the RTS packet to node A , which is overheard by node B . All nodes in the 1-hop neighborhood of node S hear the RTS and set their Network Allocation Vectors (NAVs) appropriately. Node B , which is also a neighbor of node S , sets its NAV and also notes the number of packets which node S will be sending to node A . Node B can overhear the DATA packets being sent from node S to

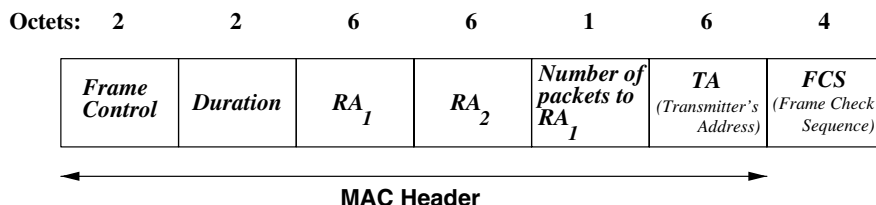


Fig. 17. New RTS frame header format.

node *A*. Suppose T_{ack} is the time it takes to transmit the ACK, then it takes $T_{\text{total}} = \text{SIFS} + T_{\text{ack}}$ for the ACK to reach the source node *S*. Once the last DATA packet has been sent by node *S* to node *A*, node *B* knows that node *A*'s ACK would have reached node *S* after T_{total} time. Node *B* then waits for $T_{\text{total}} + \text{SIFS}$ and then transmits its CTS. Since all other nodes wait for at least DIFS duration before starting a transmission, node *B* is guaranteed to capture the channel. Node *S* then transmits for the rest of its TXOP duration to node *B*. This scenario is depicted in Fig. 18.

However, things become a little more complex if there are errors or collisions during the RTS phase. Suppose node *S* sends the RTS and node *A* fails to respond (due to its NAV being set or due to a collision), then node *B* can send its CTS after waiting for the duration set in its NAV. A possible problem we might encounter here is that another neighbor of node *S*, say node *C*, also transmits immediately after the expiry of its NAV. This would result in a collision with node *B*'s CTS. We have looked at two possible solutions to this problem:

1. If the RTS is for a TXOP-sharing session, then each node in the neighborhood, except *B*, waits for an additional δ after the expiry of its NAV. This δ is the time it would take for node *B* to send its CTS and then for node *S* to send out its first DATA packet. This will ensure that no node will transmit till *B*'s CTS reaches *S*

$$\delta = \text{SIFS} + T_{\text{cts}} + \tau + \text{SIFS},$$

where τ is the propagation delay and T_{cts} is the time it takes to transmit the CTS. This has the advantage of allowing node *S* to transmit to node *B* without contention. However, if node *B* is also unable to respond with a CTS, it would only add to the delay for other nodes.

2. The second alternative we have is: if node *B* does not hear node *S* transmitting its first DATA packet to node *A* before its NAV expires, then it ignores the RTS it received from node *S*. If node *S* has to transmit to node *B*, it has to contend for the channel again.

We expect that the second approach would perform better than the first one. The reason for this is that *A* will not send out a CTS if either its NAV is set or if some other transmission collides with the RTS. Since both *A* and *B* are in the neighborhood of *S*, it is quite likely that the reason for *A*'s not replying to the RTS would hold for *B* as well, that is, even *B* might have lost the RTS sent by *S* due to some other transmission. Thus, in such a case, waiting for the additional δ duration would be wasteful.

5.1. Simulation studies

The simulations are run for a single-hop network having eight flows without link errors (i.e., packets may be lost due to collisions). We have equal number of voice and video flows. All flows start at the same time and are present throughout the duration of the simulation. The voice traffic is generated with a Constant Bit Rate (CBR) and each packet is of size 100 bytes. Since packet sizes are small, a node can transmit upto five voice packets in a single TXOP burst. Two voice flows (with the same start time) originate from each source node to different destinations, as TXOP-sharing is effective only when packets for multiple destinations are present. A-TXOP is employed for carrying video traffic. We measure the value of delay, control overhead ratio, normalized packet overhead, and the number of channel accesses for voice traffic at various offered loads. The control overhead ratio measures

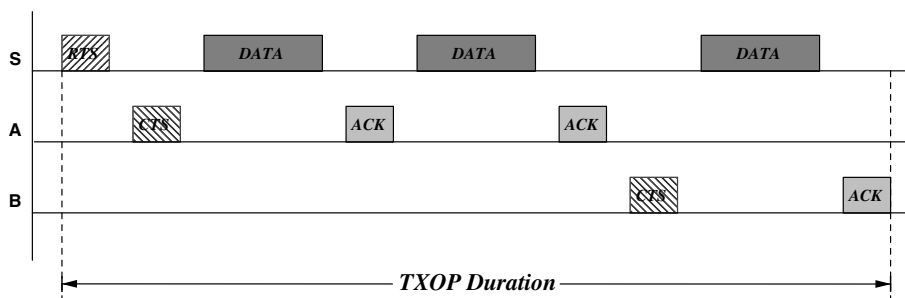


Fig. 18. RTS–CTS–DATA–ACK sequence when using TXOP-sharing.

the total amount of control data generated for delivering the voice traffic. It is measured as a ratio of the number of control bytes transmitted by nodes across the network to the number of data bytes delivered to the destinations. The normalized packet overhead evaluates the overall effort that the protocol expends for the delivery of each data packet. It is measured as a ratio of the number of control and data packets (i.e., RTS, CTS, ACK, and DATA frames) transmitted by nodes across the network to the number of DATA packets delivered to the destinations. The number of channel accesses

reflects the contention for the channel. We get this value by counting the number of RTS packets transmitted in the network. The load is varied by varying the packet rates of voice flows. The simulation results are presented in the following section.

5.2. Simulation results

From Fig. 19, it can be seen that TXOP-sharing reduces the mean end-to-end delay for voice traffic. Since both voice flows start at the same time, when a source node captures the channel it contains packets

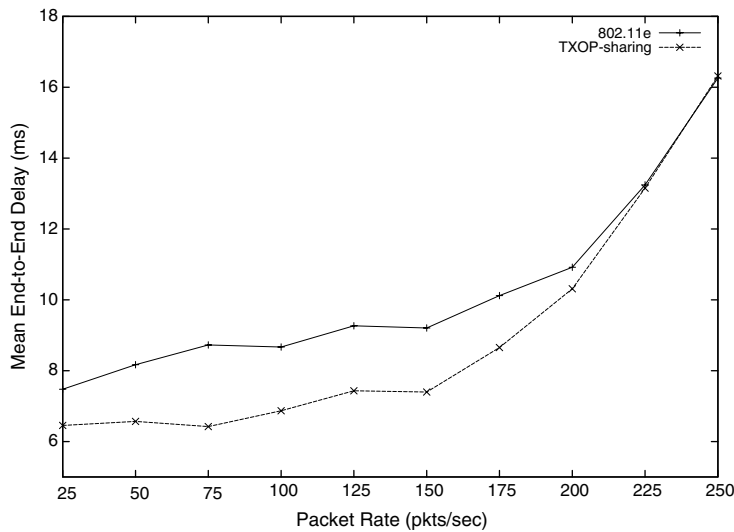


Fig. 19. Variation of end-to-end delay vs offered load for voice traffic.

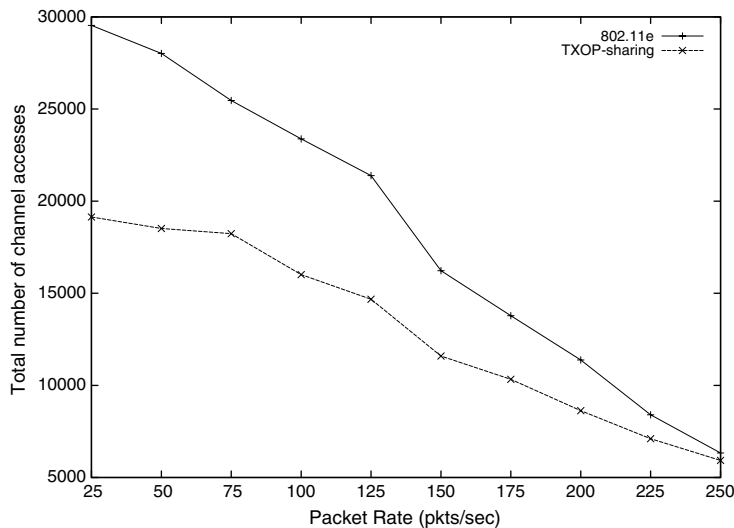


Fig. 20. Number of channel accesses vs offered load for voice traffic.

for both destinations. If the source does not have enough packets available (to fully utilize the TXOP interval) for one destination (this happens upto packet rates 150 packets per second), unlike in 802.11e, in TXOP-sharing scheme the source shares that interval with other destination. Such a sharing reduces the queuing delay and the number of contentions for the channel. Hence TXOP-sharing reduces the mean end-to-end delay and the number

of channel accesses in comparison with 802.11e (see Fig. 20). Since each channel access results in the exchange of control packets, by reducing the total number of channel accesses, TXOP-sharing scheme reduces the control overhead (see Fig. 21). However, at high packet rates, both schemes behave identically and almost have the same number of channel accesses. This can be explained by the fact that at such high packet rates, almost all the times

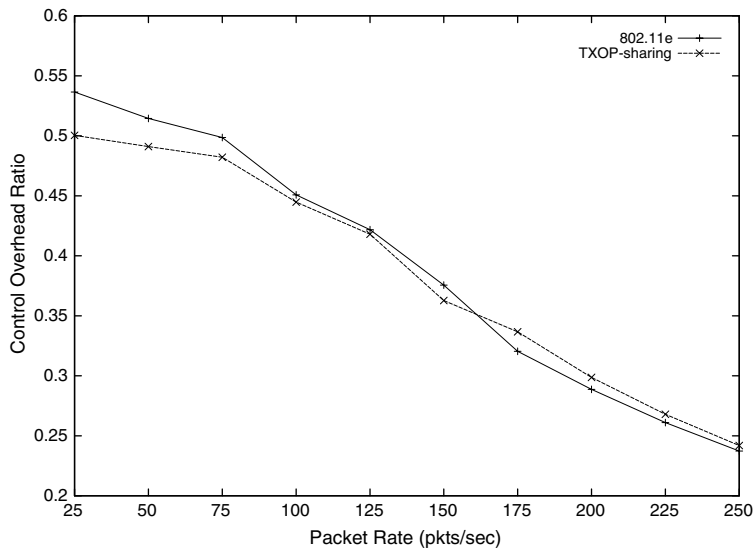


Fig. 21. Variation of control overhead ratio vs offered load for voice traffic.

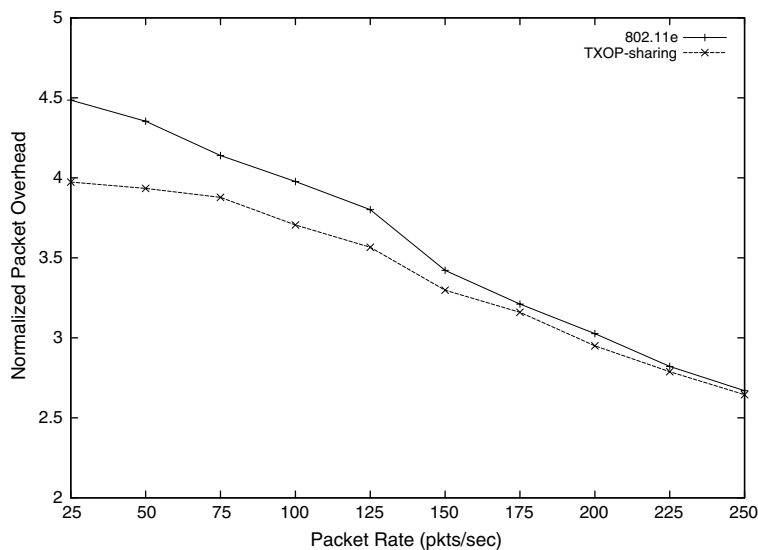


Fig. 22. Variation of normalized packet overhead vs offered load for voice traffic.

there are enough packets in the queue for a single destination to utilize the entire TXOP interval and therefore TXOP-sharing hardly comes into play. Since both schemes almost have the same number of channel accesses and TXOP-sharing increases the RTS header size by 7 bytes, at high packet rates TXOP-sharing slightly increases the control overhead in comparison with that of 802.11e. However, as shown in Fig. 22, the normalized packet overhead is consistently lower in TXOP-sharing as it reduces the average packet transmissions needed for each data packet that is delivered to the destination in comparison with that of 802.11e.

6. Relationship among ReAP, A-TXOP, and TXOP-sharing schemes

The three schemes (ReAP, A-TXOP, and TXOP-Sharing) work together to improve the performance of video and voice traffic in multi-hop ad hoc wireless networks. The relationship among these schemes is shown in Fig. 23. It is to be noted that ReAP is implemented only on AC2 and AC3 as they carry real-time traffic. A-TXOP is implemented over ReAP (AC2) to further improve the performance of video traffic. TXOP-Sharing is implemented over ReAP (AC3) to further improve the performance of voice traffic.

7. Conclusion

Better QoS support is required at the MAC layer to enable multimedia applications to work seamlessly over 802.11e based ad hoc networks. This paper addressed the QoS issues of delay and PDR for video and voice traffic. We devised a simple mechanism to dynamically reschedule the priorities of packets based on their deadlines and hop lengths to traverse. This enables packets with tight deadlines and larger hop lengths to reach the destination within the deadline, thus improving the PDR. We then introduced a method for reducing the number of channel accesses by dynamically adapting the TXOP to transmit all fragments of a video frame in a single burst. This was shown to reduce the delay for video traffic, while not degrading the performance of voice traffic. Though the I-frames of each flow were found to hold the channel for greater lengths of time, this did not increase the overall delay of other flows, because the I-frames, though important, constitute only a small fraction of the video traffic. Theoretical analysis and simulations were performed to quantify the performance gain obtained by this modification. Finally we devised a scheme for maximizing the utilization of the TXOP interval and reducing the delay and normalized packet overhead for voice traffic.

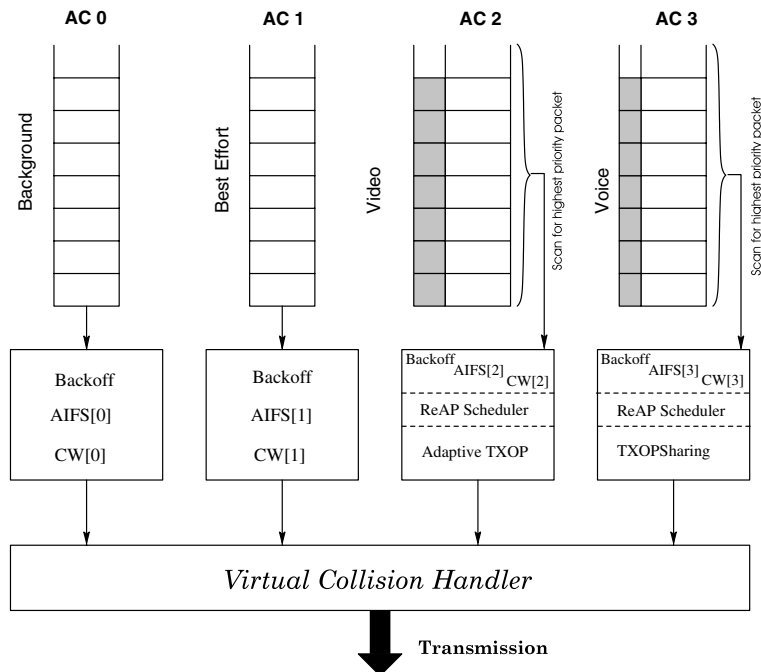


Fig. 23. Relationship among ReAP, A-TXOP, and TXOP-sharing schemes.

In this work simulations were carried out for only stationary ad hoc networks. When nodes are mobile, path breaks occur and then the performance depends on the routing protocol employed. In the future, we plan to study the performance of multimedia traffic in mobile scenarios by employing DSR as the routing protocol. Further, we plan to apply source coding techniques to sustain packet loss bursts due to mobility.

References

- [1] IEEE 802.11 WG, Draft supplement to standard for telecommunications and information exchange between systems – LAN/MAN specific requirements – Part 11: wireless medium access control (MAC) and physical layer (PHY) specifications: medium access control (MAC) enhancements for quality of service (QoS), IEEE 802.11e/D9.0, August 2004.
- [2] IEEE 802.11 WG, Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11, 1999 edition. International standard [for] information technology – telecommunications and information exchange between systems – local and metropolitan area networks-specific requirements – Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.
- [3] T. Nandagopal, T. Kim, X. Gao, V. Bhargavan, Achieving MAC layer fairness in wireless packet networks, in: proceedings of ACM MOBICOM 2000, August 2000, pp. 87–98.
- [4] N. Vaidya, P. Bahl, S. Gupta, Distributed fair scheduling in a wireless LAN, in: Proceedings of ACM MOBICOM 2000, August 2000, pp. 167–178.
- [5] S. Choi, J. Prado, S. Mangold, S. Shankar, IEEE 802.11e contention-based channel access (EDCF) performance evaluation, in: Proceedings of IEEE ICC 2003, vol. 2, 2003, pp. 1151–1156.
- [6] H. Zhu, I. Chlamtac, Performance analysis for IEEE 802.11e EDCF service differentiation, IEEE Transactions on Wireless Communications 4 (4) (2005) 1779–1788.
- [7] J. Hui, M. Devetsikiotis, A unified model for the performance analysis of IEEE 802.11e EDCA, IEEE Transactions on Communications 53 (9) (2005) 1498–1510.
- [8] I. Aad, Q. Ni, C. Castelluccia, T. Turletti, Enhancing IEEE 802.11 performance with slow CW decrease, IEEE 802.11e working group document 802.11-02/674r0, 2002.
- [9] L. Romdhani, Q. Ni, T. Turletti, Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks, in: proceedings of IEEE WCNC 2003, vol. 2, 2003, pp. 1373–1378.
- [10] F. Cali, M. Conti, E. Gregori, IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism, IEEE Journal on Selected Areas in Communication 18 (9) (2000) 1774–1786.
- [11] I. Aad, C. Castelluccia, Differentiation mechanisms for IEEE 802.11, in: Proceedings of IEEE INFOCOM 2001, vol. 1, 2001, pp. 209–218.
- [12] Y. Xiao, H. Li, Voice and video transmission with global data parameter control for the IEEE 802.11e enhanced distributed channel access, IEEE Transactions on Parallel and Distributed Systems 15 (11) (2004) 1041–1053.
- [13] A. Ksentini, M. Naimi, A. Nafaa, M. Gueroui, Adaptive service differentiation for QoS provisioning in IEEE 802.11 wireless ad hoc networks, in: Proceedings of the 1st ACM Workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks, October 2004, pp. 39–45.
- [14] M. Caccamo, L.Y. Zhang, L. Sha, G. Buttazzo, An implicit prioritized access protocol for wireless sensor networks, in: proceedings of IEEE RTSS 2002, December 2002, pp. 39–48.
- [15] M. Adamou, S. Khanna, I. Lee, I. Shin, S. Zhou, Fair real-time traffic scheduling over a wireless LAN, in: Proceedings of IEEE RTSS 2001, December 2001, pp. 279–288.
- [16] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E. Knightly, Distributed multi-hop scheduling and medium access with delay and throughput constraints, in: Proceedings of ACM MOBICOM 2001, July 2001, pp. 200–209.
- [17] M. Li, B. Prabhakaran, Dynamic priority reallocation scheme for quality of service in IEEE 802.11e WLANs, in: Proceedings of SPIE Conference on Multimedia Computing and Networking 2004, vol. 5305, 2004, pp. 83–94.
- [18] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, Mobile Computing 353 (1996) 153–181.
- [19] G. Ding, H. Ghafoor, and B. Bhargava, Error resilient video transmission over wireless networks, in: Proceedings of IEEE WSTFES 2003, May 2003, pp. 31–34.
- [20] Zhenghua Fu, Xiaoqiao Meng, Songwu Lu, A transport protocol for supporting multimedia streaming in mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 21 (10) (2003) 1615–1626.
- [21] C. Lin, H. Dong, U. Madhow, and A. Gersho, Supporting real-time speech on wireless ad hoc networks: inter-packet redundancy, path diversity, and multiple description coding, in: Proceedings of ACM WMASH 2004, October 2004, pp. 11–20.
- [22] <<http://pcl.cs.ucla.edu/projects/glomosim/>>.
- [23] I. Tinnirello, S. Choi, Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs, in: Proceedings of IEEE WoWMoM 2005, June 2005, pp. 220–230.
- [24] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, RFC 1889, 1996.
- [25] <http://trace.eas.asu.edu/scale/Citizen_Kane/qcif/simple/gopone/nrc/5/bpe/zero/frame/verbose.data>.
- [26] <<http://trace.eas.asu.edu/>>.



T. Bheemarjuna Reddy received the B. Tech. degree in Computer Science and Engineering from Andhra University, India, in 2000 and the M.E. degree in Computer Science and Engineering from the National Institute of Technology (NIT), Rourkela, India, in 2002. He is currently a doctoral student in the Department of Computer Science and Engineering at the Indian Institute of Technology (IIT), Madras, India. His

research interests include QoS provisioning and Multimedia transport in ad hoc wireless networks.



John P. John obtained his B. Tech. degree in Computer Science and Engineering in 2005 from the Indian Institute of Technology (IIT), Madras, India. He is currently working towards his Ph.D. in the department of Computer Science and Engineering at the University of Washington, Seattle, USA. His research interests include wireless networks and distributed systems.



C. Siva Ram Murthy received the B.Tech. degree in Electronics and Communications Engineering from Regional Engineering College (now National Institute of Technology), Warangal, India, in 1982, the M.Tech. degree in Computer Engineering from the Indian Institute of Technology (IIT), Kharagpur, India, in 1984, and the Ph.D. degree in Computer Science from the Indian Institute of Science, Bangalore, India,

in 1988.

He joined the Department of Computer Science and Engineering, IIT, Madras, as a Lecturer in September 1988, and became an Assistant Professor in August 1989 and an Associate Professor in May 1995. He has been a Professor with the same department since September 2000. He has held visiting positions

at the German National Research Centre for Information Technology (GMD), Bonn, Germany, the University of Stuttgart, Germany, the University of Freiburg, Germany, the Swiss Federal Institute of Technology (EPFL), Switzerland, and the University of Washington, Seattle, USA.

He is the co-author of the textbooks *Parallel Computers: Architecture and Programming*, (Prentice-Hall of India, New Delhi, India), *New Parallel Algorithms for Direct Solution of Linear Equations*, (John Wiley & Sons, Inc., New York, USA), *Resource Management in Real-time Systems and Networks*, (MIT Press, Cambridge, Massachusetts, USA), *WDM Optical Networks: Concepts, Design, and Algorithms*, (Prentice Hall, Upper Saddle River, New Jersey, USA), and *Ad Hoc Wireless Networks: Architectures and Protocols*, (Prentice Hall, Upper Saddle River, New Jersey, USA). His research interests include parallel and distributed computing, real-time systems, lightwave networks, and wireless networks. He has published more than 100 international journal and 100 international conference papers in these areas.

He is a recipient of Best Ph.D. Thesis Award from the Indian Institute of Science, Indian National Science Academy (INSA) Medal for Young Scientists, and Dr. Vikram Sarabhai Research Award. He is a co-recipient of Best Paper Awards from the 5th IEEE International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), and the 6th and 11th IEEE Annual International Conferences on High Performance Computing (HiPC). He is a Fellow of the Indian National Academy of Engineering, an Associate Editor of IEEE Transactions on Computers, and a Subject Area Editor of Journal of Parallel and Distributed Computing.