This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

EDCA-TM: IEEE 802.11e MAC Enhancement for Wireless Multi-hop Networks

Min-Soo Kim, Deepesh Man Shrestha and Young-Bae Ko

College of Information & Communication, Ajou University, Suwon, Republic of Korea

{ms7777, deepesh, youngko}@ajou.ac.kr

Abstract—In this paper, we propose the EDCA-TM, a new architecture which tunes the enhanced distributed channel access (EDCA) for multi-hop networks. In EDCA-TM, we suggest a new module for dynamic access category adaptation (DACA) over the existing channel access mechanism such as EDCA. It processes received data packets for assigning an appropriate access categories (AC) to provide delay and rate guarantee to the QoS traffic. In addition, we adopt the earliest deadline first queuing (EDF) scheme to reschedule buffered packets in the AC queues according to their urgency for actual transmission. We have simulated our proposal in the network simulator, ns-2.31, and compared it with EDCA and APHD. Results show that our schemes outperform existing protocols and provide quality service for the traffic flows within the stated requirements.

***Index Terms*—IEEE 802.11e, medium access control, QoS, wireless multi-hop network**

I. INTRODUCTION

THE WLAN mesh network based on the IEEE 802.11 radio is widely becoming popular for home, community and enterprise networking etc. Due to the use of unlicensed radio spectrum and accessible off-the-shelf components, wireless networking has provided user convenience and marketability compared to wired-LANs. However, performances in terms of throughput, latency and quality of service (QoS) have been remaining as major challenges that the wireless community still faces today for its wider deployment.

The present IEEE 802.11 MAC operates in two distributed modes: (i) Distributed Coordinated Access (DCA) [1] and (ii) Enhanced DCA (EDCA) that supports QoS [2]. Since these protocols are designed for a single-hop WLAN network, several design aspects of these systems have low compatibility with multi-hop wireless mesh networks. For example, physical and virtual carrier sensing mechanisms used in a single-hop WLAN network have been reported inadequate for mitigating the hidden and exposed terminal problems in multi-hop scenarios [3]. Similarly, enhancements in EDCA for supporting QoS are not capable of providing end-to-end service guarantees in multi-hop environment. This is primarily because packets are treated in a similar fashion with each intermediate node as with the source node regardless of varying conditions. In this paper, we focus on these issues and propose a multi-hop service differentiation to satisfy end-to-end requirements for multihop sessions.

The dynamic AC adaptation in EDCA has been previously proposed in [4]-[6]. Lera et al [4] proposed a scheme to improve QoS and throughput in single and multi-hop WLAN through a dynamic priority assignment where traffic priorities are determined dynamically at each hop, which is distinguishable from the fixed traffic categorization in EDCA. Each node computes the average transmission delay for an ACs based on the delay incurred by transmitted packets. When a station has a frame ready for transmission, the *maximum transmission delay* required for that frame is compared with the average delay. The packet is expedited if the maximum delay becomes larger than the average by setting a higher priority. Li et al [5] proposed a scheme, called “Adaptive Per Hop Differentiation (APHD)” that computes a per-hop delay budget at each participating node for the packet based on end-to-end delay requirement supplied by the application. The per-hop delay budget is the amount of time a packet is allowed to spend at one node such that it can meet the total delay requirement. When a node is aware that the budget is low, it speeds up the transmission by raising the priority level of such packets over those that have higher budget. ReAP scheduler proposed in [6] is similar to that of [5] in that the variation on the mechanism of how delay is calculated. In this scheme, each packet has a deadline based on which the laxity is computed at each hop. The priority is then recomputed as a ratio of current laxity to the remaining hops, giving higher priority to those that have high laxity and longer hops to traverse.

Our work in this paper is based on the similar observations, in which application requirements are embedded in the data packets during transmission. We consider a *bit-rate requirement* and *delay* that could be specified for any network applications [7]. For instance, multimedia streaming requires guaranteed throughput to ensure minimum quality, likewise voice over IP (VoIP) requires strict limits on jitter and delay whereas link emulation requires guaranteed throughput, jitter and latency. In our proposed architecture, which tunes EDCA for multi-hop networks, we implement a dynamic AC adaptation (DACA) module over existing channel access mechanism. It considers these requirements and other varying parameters, based on which we confirm whether a packet can be delivered within a pre-assigned duration. This decision

“This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)” (IITA-2008-C1090-0801-0015)

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

along with the estimated bandwidth and the status of the link will subsequently control the priority of transmission. In addition, we propose a next level of delay optimization, namely *earliest deadline first queuing* (EDF), which schedules packets in the queue that have their deadline to reach the destination near expiration. Unlike previous schemes, it manages faster transmission for the packets that has intensive delay requirements. We compare our schemes with the APHD [5] and EDCA to show that our bandwidth estimation based transmission in EDCA-TM performs superior than other schemes for both delay tolerant and normal traffic.

In the next section we describe the overview of EDCA. In Section III, our MAC architecture, packet structure and the operation of our scheme is presented. Section IV presents the simulation results and its analysis and finally we conclude in section V.

II. Background: EDCA in IEEE 802.11e

IEEE 802.11e MAC [2] standard provides quality of service provisioning through hybrid coordination function (HCF) on top of existing IEEE 802.11 PHY. HCF comprises of two parts: (i) Contention Period (CP) in which nodes contend in a distributed manner for accessing the medium and (ii) Contention Free Period (CFP) where channel access is centrally controlled by a coordinator node. EDCA is the recommended scheme to use during the CP. Here, we briefly describe EDCA from the perspective of structures defined for providing QoS services.

EDCA is based on a statistical, class-based QoS provisioning with three major functions -- (i) Traffic classification into eight different user priorities mapped onto four access categories (AC) as follows: background, best-effort, video and voice in ascending order of priority. Background traffic has a least priority of AC level 3 whereas voice traffic has the highest priority with level 0. (ii) Packet scheduling performed by the help of the queues associated with each AC. Priority is assigned in the MAC frame header that notifies receiving nodes about the AC level. A receiver maps the packet to the corresponding queue after extracting the priority information. The actual transmission schedule from the queue is based on the simple policy of 'first come first serve (FCFS)'. (iii) Differentiated channel access by the arbitrary inter-frame space time (AIFS) and the contention window (CW) parameters. For the ACs of higher priorities, these AIFS and CW duration are smaller compared to those of the low priorities.

Since the EDCA was originally designed for single-hop WLAN networks, values for AIFS and CW are supposed to be allocated initially by the coordinator node, like an access point for the stations. In this manner the coordinator can fairly distribute transmission opportunities to the stations. When a station has a packet to transmit, it first performs channel sensing. If the channel state is idle, it waits for the AIFS duration and starts backoff timer countdown for a random duration selected from the range of allocated CW. If the station senses a busy medium, it freezes the timer until AIFS time and restarts the countdown after the medium is idle again. If the medium is free, any queued packet is transmitted when the countdown reaches zero. Upon collision or packet loss, sender node doubles its CW range (until it reaches maximum) and repeats the wait for AIFS to attempt retransmission of the packet. To avoid collision among the ACs in the same node, virtual collision handler is defined which enforces the low priority AC to backoff in a similar way to the node that encountered a collision (refer to Fig. 1).

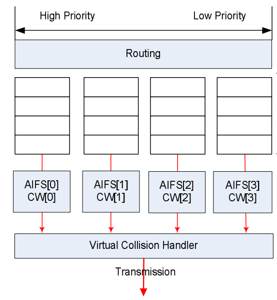


Fig. 1 The architecture of EDCA in IEEE 802.11e

The major causes for delay in EDCA are queuing, channel access, and the transmission delay. Queuing delay is the time period that a packet has to wait in the queue before being scheduled for transmission. A channel access delay includes the time duration spent by AIFS, CW and SIFS (Short Inter-Frame Space), whereas a transmission delay includes the time for sending a packet depending upon its size and data rate.

Applications that desire QoS services state specific requirements such as delay and bitrate, which are eventually mapped to ACs in EDCA. However, this mapping does not change throughout the transmission at intermediate hops, regardless of the changes in multi-hop environment. Therefore, in EDCA there is no flexibility for selecting AC according to the latest flow specific characteristics. For example, a video or voice packet is relevant only if it reaches to the destination within stipulated time. After the deadline is over, they have no relevance and can be discarded immediately. On the other hand, data of file transfer protocol (FTP) is relevant only if all packets reach the destination, but likewise does not have a strict delay requirement. Static durations such as AIFS and CW may also be delay factors. For instance, even in the absence of higher priority traffic, nodes strictly wait throughout the AIFS and CW durations for the lower priority ACs. Thus, a significant channel access delay is incurred and much of the bandwidth is wasted unnecessarily.

Our proposal, DACA, uses a novel mechanism for bandwidth estimation to maintain a bit rate and keep track of the delay status at each hop. In addition, we use a queue scheduling mechanism at intermediate nodes such that the delayed packets get an opportunity to transmit faster than others with flexible deadline through earliest deadline first queuing mechanism. We show that adjusting the EDCA parameters using dynamic mechanism improves performance by multiple folds in terms of throughput and delay.

III. Proposed scheme: EDCA-TM

As mentioned earlier, we propose the enhancements on

EDCA by satisfying the stated requirements of the flow in multi-hop networks. Since the requirements stated in the packet at the source node might change in the multi-hop environment, they are re-computed at each hop based on the initial set of parameters. In our proposal, we consider the end-to-end latency and bit rate as the transmitted packets' QoS requirement to dynamically determine AC for the packet at each hop. Our modules, DACA and EDF, are implemented at the link layer with ability to access information from the routing and physical layers. First, we describe extensions on the MAC header and then our DACA and EDF algorithms for dynamic AC adaptation will be explained.

A. MAC Header Extension

Frame Header					Extended Header			
Frame control O:2				O:30	Static fields O:8	Dynamic field O:4	O:0-2312	O:4
Protocol Version	Type	Sub Type	Existing fields	Other existing fields	bitrate_req, delay_req, total_hops	delay_so_far hops_so_far	Frame body	FCS

Fig. 2 Proposed MAC frame format for DACA in octets (O)

Fig .2 depicts the MAC frame used in DACA. Shaded fields in the extended header are new parameters that use bits from the frame body of the existing MAC frame in IEEE 802.11e [2]. The unique value is assigned at the shaded subtype field in the frame header to identify the packets that are processed in our proposal. Using this technique we ensure that only frames with an extended header are processed by our DACA module. Static fields in an extended header include *bitrate_req*, *delay_req* and *total_hops*, which are fixed at the source node once. They do not change throughout the lifetime of the packet. *bitrate_req* denotes the data rate required by the packet to maintain the throughput. *delay_req* represents the deadline for the packet to reach the destination node, which are stated according to the application requirement. *total_hops* represents the number of hops to the destination that is reported by the routing function. The source routing algorithm such as dynamic source routing (DSR) [8] defines total number of hops which does not change during the packet transmission. However, this is not always true in other routing protocols. In case the intermediate hop changes the path to reach the destination, the end-to-end hop count becomes invalid and must be updated. In our implementation, we remove the local route repair function in AODV [9] such that the intermediate nodes are consistent with the initially discovered path from the source to the destination.

Other two dynamic fields are updated at each hop. *delay_so_far* represents the actual delay experienced by the packet. To compute this parameter, the generation time is noted when the link layer processes the packet at the source node. When the packet is at the next hop node, its generation time is deducted to the current time of day. Thus, *delay_so_far* includes all delays that are incurred since the generation of the packet such as queuing, access, propagation and re-transmission delays. This method of computing *delay_so_far* however requires synchronization among the clocks of distributed nodes. Alternative method of computing such delays by estimating the propagation duration of the transmitted packets as described in [5] can also be used in our protocol. Similarly, the hop count is updated by incrementing the *hop_so_far* variable at each node. It also indicates the current position of the packet in the flow with respect to the source and destination. In our implementation, these parameters are extended in the frame body without requiring any change in other existing MAC functionalities.

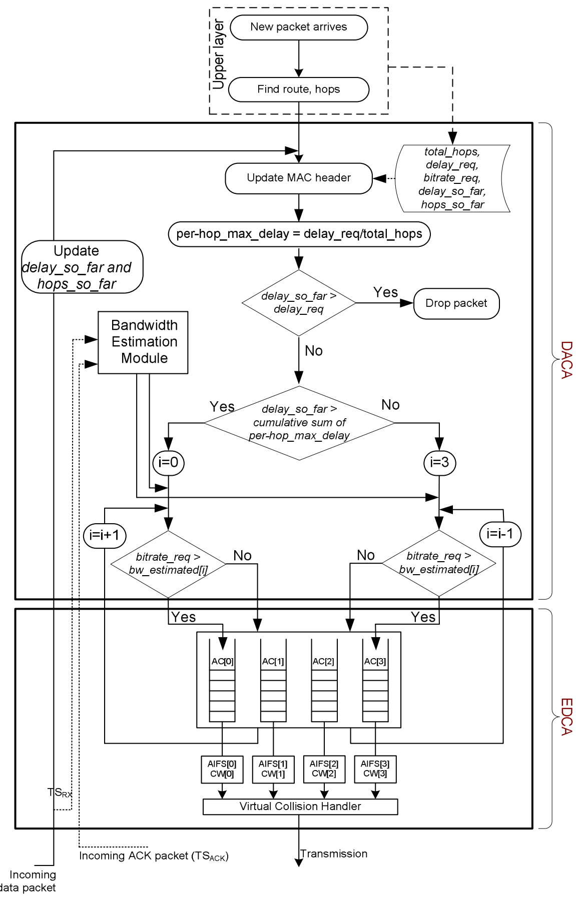


Fig. 3 Architecture of EDCA-TM showing DACA module over EDCA

B. Dynamic access category adaptation (DACA)

Now we describe our DACA algorithm based on the parameters shown in the previous section. As shown in Fig. 3, DACA is implemented in the link layer below the routing layer. Packets arrive at the link layer, either from the lower layer (for forwarding) or from the application layer when the node is a packet source. If a node is a source, values for the static fields are updated from the upper layers. Thus, the link layer agent updates values for the *bitrate_req*, *delay_req* and *total_hops* from the application layer and the routing layer, respectively. Similarly, *delay_so_far* and *hops_so_far* is initialized to zero. The packet is then mapped to ACs after being processed through our DACA algorithm. Details of our proposal are described later in this section.

The incoming packets from the medium are handled by the DACA in a slightly different manner. First, *delay_so_far* and *hop_so_far* is updated when the packet is received at the link layer. Since the requirements are already embedded in the packet header, our algorithm compares the experienced delay to the delay requirement. If that delay has exceeded the requirement than the packet is dropped assuming that it is expired for the purpose of application. For example, audio traffic has very less tolerance for delay. However, if the packet arrives after the deadline, the message gets distorted and can convey a different meaning than the original. Moreover, dropping such packets provide usable bandwidth for other packets in transfer. As shown in Fig. 3, if the deadline is not expired, DACA computes *per-hop_max_delay* as a ratio of *delay_req* to the *total_hops*. Finally, the cumulative sum of *per-hop_max_delay* of the receiving node is compared with *delay_so_far* resulting into one of the two cases described below:

- If the packet has arrived earlier, i.e., the *delay_so_far* is lesser or equal to the cumulative sum of *per-hop_max_delay*, we queue the packet in the AC with low priority (i=3). However, before fixing the assignment we compare the *bitrate_req* with the average bandwidth of the AC resulted from the bandwidth estimation module. This module for comparing the *bit_rate* is explained in Subsection C.
- If the value of *delay_so_far* is greater, it needs more resource to meet the delivery. A higher priority AC is expected to reduce the delay and gradually meet the *per-hop_max_delay*. Thus we try to assign a queue searching from (i=0). However, a channel condition could be in adverse state for that AC, thus again we rely on our bandwidth estimation module to report the network status as in the previous case. Thus, if the bandwidth is not sufficient we choose the next lower priority AC that satisfies the *bitrate_req*.

C. Bandwidth Estimation Module (BEM)

MAC level bandwidth is estimated on the basis of the size of data packet and the time spent for successful transmission since the packet has arrived at the node. The timestamp for the received packet is stored immediately after it is received and deducted after the ACK for the same is received as shown in Fig. 3. We consider unicast packets that are successfully transmitted and acknowledged by the received ACK message.

$$BW\ [i] = \frac{PacketSize\ \ (Kb)}{TS_{ACK} - TS_{RX}\ (Sec)} \qquad (1)$$

$$BW_{estimated}\ [i] = (1-\alpha) * BW_{estimated}\ [i] + \alpha * BW\ [i] \qquad (2)$$

(where, $i \,\varepsilon\, 0..3$)

In eq. (1) BW[i] represents the bandwidth measured in Kbps, where *i* denote the AC level selected by our DACA module from which the packet is transmitted towards the next node. TS_{RX} is the timestamp when the data packet is received at the node and TS_{ACK} is the timestamp when the same packet is successfully transmitted to the next node (notified by the receipt of the ACK packet). The *PacketSize* represents the actual size of the packet. Thus, *BW[i]* reflects the network condition of each AC in which it was queued.

We compare the bitrate requirement with the weighted average of bandwidth that is computed as shown in eq. 2. In this equation α is the configurable parameter which is set to 0.6 in our simulation. As shown in Fig. 3 this module computes average bandwidth and gives input to the DACA module for comparing the bit rate requirement of the packet.

D. Earliest Deadline First Queuing (EDF)

The EDCA-TM and EDCA both use FCFS queuing mechanism. The packets are queued in the order of its reception at the link layer. However, in multi-hop mesh environments this is not sufficient to provide guaranteed service to the high priority flows. The packets are not further prioritized to provide fine-grained service within the queues of AC.

EDF scheduling is a dynamic scheduling algorithm used in real-time operating systems. It schedules the task, which has the least time left to the deadline. The complexity of EDF algorithm is not very high compared to other real-time scheduling algorithms because it just needs to find the task with the nearest deadline in the queue. In our EDF scheme, the *remaining_perhop_delay* is computed as the ratio of remaining delay (numerator) and hops (denominator) as shown in eq.3. The *remaining_perhop_delay* is used as a key in every entity of AC queue. The queued packets are scheduled in order of this key value such that the delayed packets get more opportunity for transfer.

$$remaining\_perhop\_delay = \frac{delay\_req - delay\_so\_far}{remaining\_hops} \qquad (3)$$

Based on this key, the packets with the least remaining delay (closest to the deadline) are scheduled for early transmission. In the simulation results we show that the latency for the flows with smaller deadline shows better latency results than the FCFS queuing technique used for EDCA.

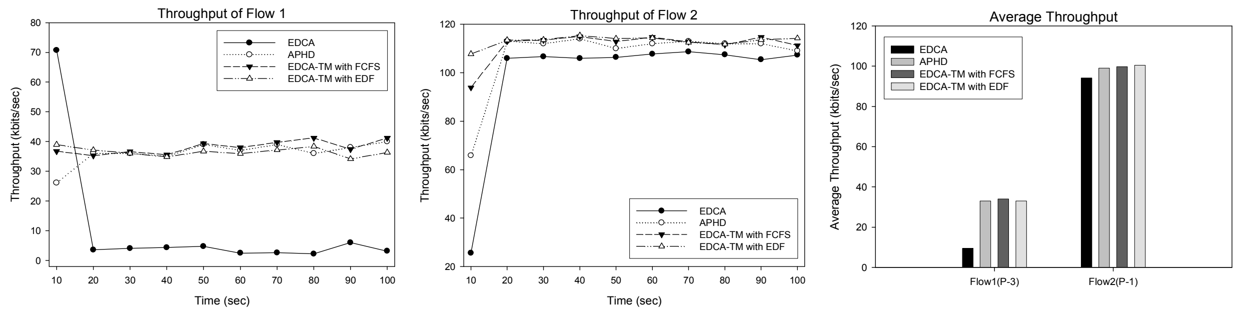
IV. Performance Evaluation

We performed extensive simulations using NS-2 (version 2.32) to evaluate the performance of EDCA-TM. We compared our schemes with APHD [5] and basic EDCA using the TKN-EDCA version [10]. The EDCA settings for contention window are same as in the IEEE 802.11e standard. Pre-experiments were made to derive the optimal value of α for bandwidth estimation, which we found the value to be 0.6. The routing protocol used is AODV [9]. A node's transmission range is 250 meters while the carrier-sensing range is 550 meters. Link bandwidth is 1Mbps and RTS/CTS are not used.

A. Simulation Model and Results for Simple Scenario

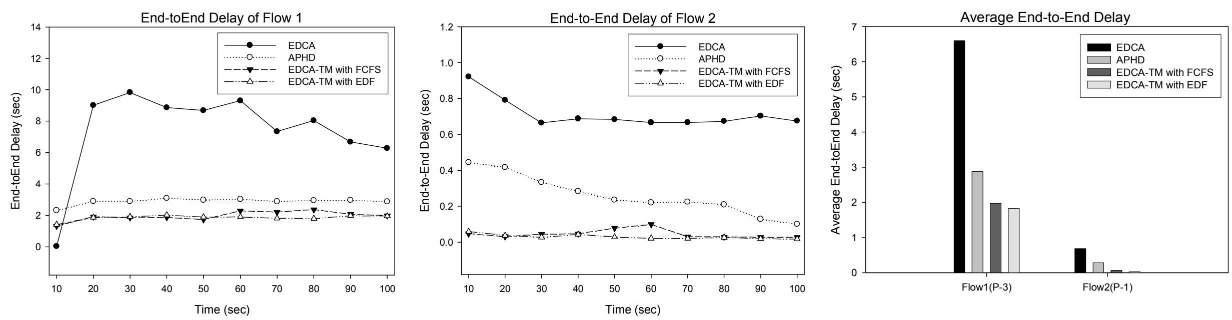
In the first scenario, we first use a simple topology shown in Fig. 4 with two flows. Flow 1 (priority 3) has lesser priority because it has lower delay and bitrate requirement than flow 2 (priority 1). This scenario clearly shows the unfairness caused by the high priority flow on the low priority flow. Priority values are only used for EDCA, whereas delay and bit rate requirements are used in APHD and EDCA-TM. The size of the data packet is set to 150 bytes and total simulation time is 100s.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.



(a)Flow 1 Throughput *(b) Flow 2 Throughput* *(c) Average Throughput (during 100 sec)*

Fig. 5 Performance Comparison in terms of *Throughput* with the Simple Topology of Fig. 4



(a)Flow1 End-to-End Delay *(b) Flow 2 End-to-End Delay* *(c) Average End-to-End Delay (during 100 sec)*

Fig. 6 Performance Comparison in terms of *End-to-End Delay* with the Simple Topology of Fig. 4

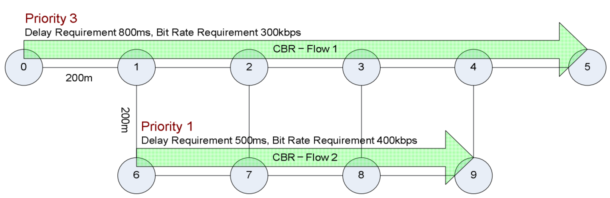


Fig. 4 Simulation Topology in Simple Scenario

The throughput is measured as an amount of data received within subsequent interval of 10 seconds at the receiver. Similarly end-to-end delay is also computed by averaging the delays of the packets received within 10 seconds interval.

Fig. 5 shows the throughput results at destinations while applying EDCA, APHD and EDCA-TM using FCFS and EDF. Throughput in EDCA (Fig. 5(a)) is initially high but reduces quickly when more nodes participate in transmission along with the introduction of flow 2 with higher priority. As mentioned earlier, in EDCA, the packet priority remains unchanged throughout the traversal from source to destination. Therefore, all nodes in the path process the packets in a similar manner irrespective of the changing network conditions. Competing packets of flow 2 with higher priority grabs the bandwidth, which makes it difficult for flow 1 to create opportunities for transmission due to its lengthy AIFS and CW duration. Throughput remains similar in our scheme and APHD because many packets are successfully transmitted to their destination compared to loss of packets in EDCA. Due to AC adaptation, fair amount of opportunities are created for flow 1 to transmit packets toward the destination. Thus we observe that in multi-hop scenarios, static priority assignment is ineffective in maintaining the consistent throughput for the flows. Throughput of flow 2 is higher in all cases compared to that of flow 1 as it has a higher priority and less number of hops for the packets to reach the destination. Fig. 5(c) presents the average throughput for each flow showing that higher priority flows have similar outcome, however the throughput for low priority flow in EDCA is squeezed throughout the transmission. Thus in conclusion EDCA-TM shows 72% and 13% average improvement in throughput compared to EDCA for flow 1 and flow 2 respectively.

Fig. 6 shows the end-to-end delay of flow 1 and 2 respectively. End-to-end delay is computed by deducting the time at which the packet is received at the destination with the time of origination at the source. The latencies in EDCA for flow 1 of low priority increases faster primarily due to queuing and access delay at each hop during transmission. In dynamic schemes APHD and EDCA-TM, transmission can be made faster by increasing the priority which reduces the significant access delay and queuing delay. In both fig. 6(a) and (b) we show that the latency of our proposed scheme EDCA-TM is better than the APHD. Packets skip such ACs that is already congested and has low bandwidth because of our BEM module for bandwidth estimation. Since the service for queues are performed in a round robin manner, more packets get opportunity to be transmitted instead of being dropped from the queue due to congestion. The EDF performs better than FCFS as it forwards the marginal packets with low delay budget. In average, our scheme shows 75% and 96% better latency than EDCA and 35% and 91% improvement compared to APHD for flow 1 and flow 2, respectively.

B. Simulation Model and Results for Realistic Mesh Scenario

This simulation is performed in the 6x6 grid of 36 nodes with the distance between each node fixed to 200m. A CBR source at the up-right corner generates priority 1 traffic for the destination at the down-left corner. The background Pareto traffic of priority 3 is increased from 1 to 5 flows to show the effect of such traffic to the priority flows. The packet size for high priority flow is 150 bytes with the bit rate requirement set to 200Kbps. Similarly, the background traffic has the packet size of 300 bytes and 50Kbps rate, and the source and destination of background traffics are randomly selected. The results in Fig. 7 and 8 show the average throughput (at receiver) and end-to-end delay of the high priority flow, respectively.

As seen in Fig. 7, throughput declines for all protocols when the background traffic is increased. This happens due to increase of interference and collision. Average throughputs for our schemes and APHD are better than EDCA since the packets are categorized in four ACs according to the delay requirement and varying conditions.

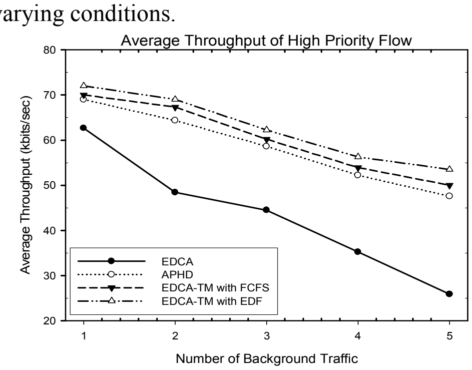


Fig. 7 Average throughput of high priority flow against increasing injection of low priority background flows

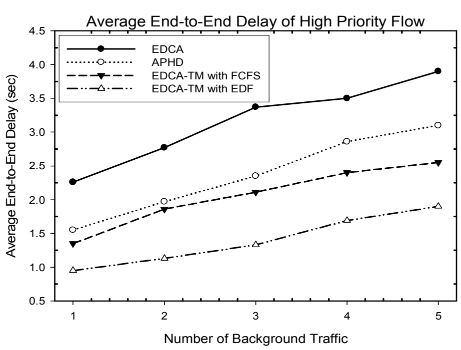


Fig. 8 Average E2E Delay of high priority flow against increasing injection of low priority background flows

The average latency for EDCA-TM using EDF becomes distinctly better than other schemes as shown in Fig. 8. EDF speeds up the packets that have deadlines closer to expiration, ensuring the decrease in end-to-end average latency. Our proposed BEM and EDF works in tandem, first by selecting AC that can satisfy bit rate requirement and pushing the packet forward to be transmitted faster as it becomes urgent. In conclusion our proposed EDCA-TM with EDF shows 27% decrease in average end-to-end latency compared to the FCFS scheme.

CONCLUSION

The enhancement in EDCA scheme for QoS traffic is presented in this paper. With the rapid growth of multi-hop networking technologies, MAC supporting multi-hop transmissions is of important concern. Major motivation of this work lies on the absence of MAC protocols for providing QoS services in the multi-hop environment. Thus, in this paper we studied the present works and proposed our own design for dynamic adaptation of existing QoS structures for better throughput and latency. This paper also highlights the need of bandwidth estimation and bit rate requirement for QoS. The EDF queuing scheme is customized to improve the existing FCFS scheme which significantly reduce latency when the network load is high. Our proposal shows 86% and 42% average improvement in terms of average end-to-end latency and average throughput.

REFERENCES

[1] IEEE 802.11 Standard WG, "IEEE Standard for Information Technology – Telecommunication and Information Exchange Between Systems – LAN/MAN Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE Std 802.11, first ed., 1999

[2] IEEE 802.11 Standard WG, "IEEE Standard for Information Technology – Telecommunication and Information Exchange Between Systems – LAN/MAN Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 8: Medium Access control (MAC) Quality of Service Enhancements", IEEE Std 802.11e-2005

[3] J. Bicket *et al*., "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," *MobiCom '05, Proc. 11th Annual Int'l Conf. Mobile Computing and Networking, Association for Computing Machinery Special Interest Group on Mobility of Systems, Users, Data and Computing*, Cologne, Germany, 2005, ISBN 1-59593-02-000s, Sept. 2005.

[4] A. Lera *et al*," "Improving QoS and Throughput in Single-and Multihop WLANs through Dynamic Traffic Prioritization," *IEEE Network*, pp. 35-44, 2005

[5] J. Li *et al*., "APHD: End-to-end Delay Assurance in 802.11e Based MANETs," *Mobile and Ubiquitous Systems - Workshops, 3rd Annual International Conference*, pp. 1-8, 2006.

[6] T. Reddy *et al*., "Providing MAC QoS for multimedia traffic in 802.11e based multi-hop ad hoc wireless networks," *Computer Networks*, vol. 51, pp 153-176, 2007.

[7] A. Ganz *et al*., "Multimedia Wireless Networks Technologies, Standards, and QoS", Prentice Hall, 2004, ISBN 0130460990

[8] P. Johanson and D. Maltz, "Dynamic source routing in ad hoc wireless networks", *Mobile Computing, Kluwer Publishing Company*, ch. 5, pp. 153-181, 1996.

[9] C. Perkins et al., "Ad-hoc on-demand distance vector (AODV) routing," *IETF, RFC 3561*, July 2003.

[10] S. Wietholter and C. Hoene, IEEE 802.11e EDCF and Simulation Model for NS-2, http:// www.tkn.tu-berlin.de/reserach/802.11e_ns2/.