ChaPLeT: Channel-dependent Packet Level Tuning for Service Differentiation in IEEE 802.11e

Naomi Ramos, Debashis Panigrahi, Sujit Dey
Mobile Systems Design and Test Lab
Dept. of Electrical and Computer Engg., University of California, San Diego
{naomir, dpani, dey}@ece.ucsd.edu

***Abstract*— With the increased use of IEEE 802.11b wireless networks, there is a strong need to support diverse Quality of Service (QoS) requirements. Recent standardization efforts are being pursued in IEEE 802.11e to introduce a framework for QoS enhancements to the IEEE 802.11b standard. Although IEEE 802.11e provides a level of service differentiation by statically associating different QoS parameters for pre-defined traffic classes, this upcoming standard does not consider the problems introduced by varying channel conditions present in wireless networks.**

In this paper, we present a Channel-dependent Packet Level Tuning (ChaPLeT) mechanism to maintain service differentiation in the presence of channel variation. We propose the use of three parameters, namely *fragmentation threshold*, *persistence factor*, and *defer countdown* in enabling packet-level prioritization. Using a runtime adaptation policy, ChaPLeT dynamically configures the above parameters appropriately based on traffic class, as well as current channel conditions. We show the effectiveness of our approach by comparing our adaptation policy with the current IEEE 802.11e scheme.

I. INTRODUCTION

The increasing popularity of mobile connectivity coupled with the inexpensive access provided by Wireless LAN (WLAN) infrastructures have sparked a growing interest and usage of IEEE 802.11b as an edge access technology. With the increased use of wireless networks for high data rate access, there is a greater diversity in application requirements such as desired bandwidth and packet loss tolerance. Hence, WLAN networks have the challenge of providing support for diverse Quality of Service (QoS) requirements in a dynamic wireless medium.

The IEEE 802.11b legacy MAC layer uses Distributed Coordination Function (DCF) as a basic access mechanism and is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol [1]. The standard DCF does not provide any mechanism to enable differentiation between different traffic flows. A number of schemes have been proposed to enable differentiated access in the legacy DCF-based mechanisms. The initial studies focused on the effectiveness of different parameters in providing differentiation between traffic flows [2], [3]. Additionally, a few algorithms have been proposed to change the parameters dynamically [4], [5], [6]. While some of the above mechanisms are able to provide minimal service differentiation, the techniques target only one parameter at a time and assume perfect channel conditions.

Motivated by the above studies, recent standardization efforts are being pursued to enhance the channel access mechanism of the IEEE 802.11b protocol, resulting in a new standard [7]. One scheme proposed in the IEEE 802.11e standard is called Enhanced Distributed Coordination Function (EDCF). In EDCF, service differentiation is achieved with the introduction of different traffic classes and the association of different contention parameters to each of these classes. Although EDCF is able to provide a level of service differentiation, the mechanism does not consider the varying channel conditions present in WLAN networks. The variance can include changes in the Bit Error Rate (BER), caused by interference and propagation loss, and the possibly high collision rate due to the network load. Depending on the channel variation, the traffic class-based service differentiation mechanism may not be sufficient to provide the desired QoS and may lead to problems, such as *priority inversion* and *starvation*, which will be explained later in the paper.

Paper Contribution and Outline

In this paper, we present a Channel-dependent Packet Level Tuning (ChaPLeT), which is built on top of the service differentiation framework proposed in IEEE 802.11e. Specifically, we evaluate the potential for dynamic adaptations of link layer parameters to achieve differentiated services over varying channel conditions. We identify three packet-level parameters, namely *fragmentation threshold*, *persistence factor*, and *defer countdown*, and study the effects of these parameters in maintaining service differentiation in the presence of channel variations. We target these parameters with the goal of developing an adaptive IEEE 802.11e data link layer, which not only uses traffic classification, but also uses channel information and experienced performance to meet application QoS requirements.

Having studied the effects of the above parameters, we develop an adaptation framework that sets the parameters' values appropriately based on current channel conditions, such as experienced BER, and application requirements, such as expected throughput. Since the channel conditions experienced differ from node to node, we use a distributed approach, rather than a centralized one, for our service differentiation adaptations. In order to show the effectiveness of our approach, we compare our adaptation policy with the EDCF scheme, where parameters are statically assigned. We demonstrate that significant improvement can be achieved by enabling dynamic adaptation in the IEEE 802.11e data link layer.

The rest of the paper is organized as follows. We first present a brief background in Section II on IEEE 802.11e. In Section III, we provide some motivation for our work and present the effect of varying channel conditions on service differentiation. Next, we provide a description of the selected parameters and the ChaPLeT adaptation framework in Section IV. Experimental results follow next in Section V, demonstrating the effectiveness of our adaptation methodology. Finally, we conclude and describe future directions.

This research is supported by the Center for Wireless Communications (CWC), University of California, San Diego.

II. IEEE 802.11E BACKGROUND

Before presenting our proposed adaptation, we provide a brief description of the improvements made in IEEE 802.11e to support QoS. In this paper, we consider only the infrastructure mode of IEEE 802.11 WLAN and we focus specifically on distributed contention-based medium access of IEEE 802.11e's EDCF.

In legacy DCF, a node wanting to send a packet waits for the wireless medium to be idle for a fixed period called DCF Inter Frame Spacing (DIFS) before attempting to transmit. In order to avoid collisions among nodes, the node chooses a random number of slots to wait before accessing the medium. The backoff period is chosen to fall in a range predefined by each node, called the Contention Window (CW). If the channel is idle for the backoff period, the node is able to transmit the packet and waits for an acknowledgment frame (ACK) from the receiving node. If an ACK is not received within a timeout period, the packet will be retransmitted and the maximum backoff period is doubled to decrease the probability of collision between contending nodes.

Based on the legacy DCF functionality, the IEEE 802.11e standard makes improvements for QoS and proposes a new channel access mechanism called EDCF. In this scheme, service differentiation is achieved with the introduction of different traffic classes (TCs) and the association of different contention parameters for each of these classes. The parameters used to enable service differentiation are (1) Arbitration Interframe Spacing ($AIFS$), (2) Contention Window Minimum (CW_{min}), and (3) Persistence Factor (PF).

The basic access mechanism in EDCF is shown in Figure 1. As with the legacy DCF, each node starts backoff after detecting channel to be idle for a fixed duration called $AIFS$. In this case, flows belonging to different traffic classes use different $AIFS$ periods. Similarly, contention window sizes differ depending on the traffic classes. Traffic classes with higher priority use lower values for AIFS and CW, and thus, will experience a lower waiting period on average than other classes.

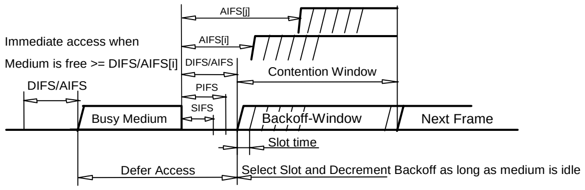


Fig. 1. Enhanced Distributed Coordination Function

An additional modification made to the standard is the Persistence Factor (PF). After an unsuccessful transmission attempt, rather than simply doubling the CW, a new CW is calculated by scaling the previous CW by a PF factor. By associating different PF values for traffic classes, it is ensured that traffic classes with higher priority have a higher probability of accessing the channel after an unsuccessful attempt. Based on the above parameters, the backoff time for a traffic class, TC, in IEEE 802.11e is calculated as follows, where $CW_i = PF[TC]^i * CW_{min}[TC]$

$$Backoff = AIFS[TC] + rand(0, CW_i) * SlotTime \quad (1)$$

III. MOTIVATION

Having provided the background of IEEE 802.11e, we now motivate the need for packet level tuning by presenting the problems that can occur due to channel variance.

IEEE 802.11e provides a means of service differentiation in the scenario where all nodes are experiencing the same channel conditions, *i.e.* signal strength, interference, *etc*. However, in the more realistic case, where nodes are experiencing variable channel conditions depending on location and interference, traffic class-based PFs are insufficient in providing adequate QoS. Two problems that can occur are *priority inversion* and *starvation*.

Priority inversion occurs in situations when poor channel conditions cause high priority nodes to achieve lower throughput than low priority nodes. In these scenarios, because of poor channel conditions, the higher priority nodes will need to retransmit more than the low priority nodes. Despite the reduced initial contention window, EDCF's parameter settings have little or no effect. Retransmissions cause the high priority nodes to magnify their contention window, leading to an average channel access time greater than the lower priority nodes. Figure 4, which will be further explained in the Section V, illustrates that priority inversions can occur in poor channel conditions. Note that the average throughput of a high priority node (n4_H) is less than a lower priority node (n6_M1) due to the channel conditions faced.

Another problem that can occur due to channel conditions is starvation. Achieved throughput of lower priority nodes in EDCF can be heavily affected by the static settings of the class-based parameters, the number of higher priority nodes present, and channel condition variances. Unable to access the medium, due to a combination of its magnification of its contention window and excessive deferrals to allow higher priority nodes, low priority nodes may be handicapped to a point of starvation. For example, the minimal achieved throughput of the low priority node (n1_L) in Figure 4 reveals a starvation scenario.

IV. CHANNEL-DEPENDENT PACKET LEVEL TUNING

In this section, we now describe Channel-dependent Packet Level Tuning (ChaPLeT) for IEEE 802.11e. First, we describe and present the effects of tunable parameters, which includes *fragmentation threshold*, *persistence factor*, and *defer countdown*. We then present runtime adaptation policies using the presented parameters to maintain packet level service differentiation.

A. Tunable Parameters

We present three tunable parameters that can be used to alleviate the problems caused by varying channel conditions. We motivate the use of each of the parameters and describe their effects.

1) Fragmentation Threshold: In order to address poor channel conditions, IEEE 802.11-based networks provide the ability to partition higher level packets into smaller units. This is made possible through a parameter, called the *fragmentation threshold*, which indicates the maximum packet size before fragmentation is necessary.

Fragmentation can be used to respond to dynamic erroneous channel conditions. Because the probability of a successful transmission increases with a decrease in packet size, fragmentation can improve the packet transmission reliability. However, partitioning large data packets into smaller fragments can lead to additional data transmission overhead. Once a packet is partitioned, each fragment is augmented with individual headers, sent as an independent transmission, and acknowledged individually.

Hence, under good channel conditions, fragmentation can lead to suboptimal performance.

In the case of ChaPLeT, fragmentation threshold can be used to improve throughput and enable nodes to meet their traffic requirements despite channel condition variations. In the case where EDCF prioritization is insufficient for nodes to meet their required throughput, the fragmentation threshold can be chosen appropriately to improve throughput and meet the node requirements. For our analysis, we estimate the optimum value setting for this parameter. We use similar techniques in our previous work [8], but evaluate the optimal value to achieve maximum throughput rather than to minimize energy. Due to space constraints, we include here only a brief description of our analysis.

The throughput per packet can be calculated as the ratio of the packet size, X, to the expected total transmit time, t_{pkt}. Given that n_{frag} is the number of fragments after a packet has been partitioned, a_{ret} is the average number of attempts before a successful fragment transmission, and t_{frag} is the transmission time for each fragment, the expected total transmit time, t_{pkt} can be written as $t_{pkt} = (n_{frag} * a_{ret} * t_{frag})$.

In order to find the optimum fragmentation threshold, we compute the gradient (dT/dF) and equate it to zero. The optimum fragmentation threshold can be calculated as follows, with C representing average overhead time per fragment, which includes inter-frame spacing, medium access, and acknowledgment time, and R representing the data rate (bits/sec).

$$F_{opt\_thrpt} = -\frac{C * R}{2} + \sqrt{\frac{C * R}{BER}} \tag{2}$$

2) Persistence Factor: The next parameter we target is the persistence factor. As described in the background section, one of the schemes used in the IEEE 802.11b to avoid collision is the magnification of the CW after every unsuccessful transmission, so as to minimize the probability of collision on the next attempt. Note that the magnification of CW is wasteful if the retransmission is caused by channel related errors in the packet rather than collision. In legacy IEEE 802.11, the magnification factor was statically fixed such that the CW doubled for every retransmission. As described earlier, EDCF introduces a magnification factor, PF, based on traffic classification. The goal of this improvement is to provide service differentiation even during retransmissions caused by collisions.

The choice of traffic class-based PFs provides a clear means of service differentiation in the case where all nodes are experiencing the same channel conditions. However, varying channel conditions can hamper EDCF mechanisms and can lead to priority inversion. In order to address this problem using ChaPLeT, we consider providing nodes, based on traffic class and channel conditions, with the ability to decrease their PF beyond the initial static setting of EDCF to improve their situation. Rather than assume every retransmission is caused by collision, nodes can estimate the probability that the retransmission is caused by poor channel conditions and use this estimate to determine what PF value is appropriate.

3) Defer Count: In addition to the above factors, another mechanism we plan to target is the deferral process. In the IEEE 802.11b standard, a node defers access to the medium if another node transmits before its backoff period is completed. In this case, the node stops the backoff process, and waits for the channel to be idle to resume countdown of the residual backoff period. The waiting time introduced by the other nodes accessing the medium can lead to significant delay in the packet transmission, and hence decrease the average throughput. In addition to the possibility of starvation, priority inversions may occur since deferral periods are not taken into account by IEEE 802.11e provisions for service differentiation.

ChaPLeT considers the scaling of the countdown rate, called *defer countdown*, in order to prioritize nodes who have experienced a significant number of deferrals. This scaling can aid in preventing starvation, by providing higher throughput to nodes obtaining less than their required throughput and experiencing excessive deferrals.

B. Adaptation Policy

This section describes the overall ChaPLeT adaptation policy in greater detail. ChaPLeT takes a distributed approach of achieving QoS differentiation with each node monitoring and tuning their parameters to the current channel conditions. The policy uses the fragmentation threshold parameter to generally improve the throughput of nodes in poor channel conditions, the persistence factor parameter to combat priority inversion, and the defer countdown to decrease the possibility of starvation. As seen in Figure 2, the ChaPLeT framework consists of two major components, a *monitor* and a *tuner*.

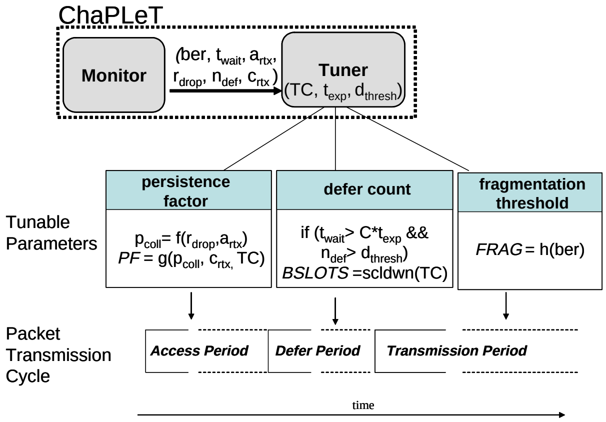


Fig. 2. ChaPLeT Adaptation

The first component, the *monitor*, records the experienced channel conditions, such as BER, dropped packet ratio, r_{drop}, and number of retransmissions per packet, a_{rtx}. It also tracks packet-specific information, such as the time spent on the current packet, t_{wait}, number of deferrals n_{def}, and current retransmission count, c_{rtx}. The *monitor* uses these measurements to estimate uplink channel conditions.

The second component of ChaPLeT is the *tuner*, which makes decisions on the value of the targeted parameters. The *tuner* keeps track of node/flow-specific information, such as traffic class, TC, the expected waiting time for the current packet, t_{exp}, and the tolerable number of deferrals, d_{thresh}.

As shown in Figure 2, the ChaPLeT *tuner* affects three phases of the packet transmission cycle: (1) *Access period* which is the minimum time (number of backoff slots) a node has to wait before attempting to transmit, (2) *Defer Period* which is the time consumed by other nodes' transmissions prior to accessing the medium, and (3) *Transmission Period* which is the time consumed in sending the packet/fragments successfully. Each of the individual parameters affects different periods in the packet transmission cycle, as will be described later. Hence, the policies are run in parallel since they have very minimal dependencies between each other. Next, we focus on presenting each of the adaptation policies.

Fragmentation Threshold: The ChaPLeT *tuner* estimates the optimum fragmentation threshold value to maximize throughput under current channel conditions (BER), using the values given by Equation 2. Note that the tuning of the fragmentation threshold is primarily used to combat poor channel conditions and is performed periodically. Since an IEEE 802.11e node can send smaller fragments sequentially without having to re-access the medium, fragmentation only affects the *Transmission Period*.

Persistence Factor: In order to set the persistence factor appropriately, the *tuner* first evaluates the current channel conditions and estimates whether the experienced retransmissions are mainly caused by collisions or by a poor channel. The probability of collision, p_{coll}, is determined using the average dropped packet ratio, r_{drop}, experienced BER, and average number of retransmissions per packet, a_{rtx}, over a specified window. Since the probability of a collision decreases with each retransmission attempt, high observed values of r_{drop} and a_{rtx} indicate that retransmissions are likely to be caused by poor channel conditions, rather than collisions.

With the p_{coll}, information about the current retransmission count, c_{rtx}, and the node's traffic class, TC, the policy then determines how aggressively the PF should be decreased. For example, if p_{coll} is estimated to be very minimal, the persistence factor is decreased to the minimum value at an early retransmission attempt. On the other hand, if the p_{coll} is estimated to be significant, the PF is decreased slightly. Note that this parameter has a direct impact on the *Access Period*, since the calculation for the backoff slots uses the PF after every retransmission, as explained in Equation 1. Because we target this adaptation to avoid priority inversions, the PF reduction is dependent on traffic class and is more aggressive in the case of a high priority node.

Defer Countdown: The final parameter determined by the ChaPLeT *tuner* is the scaling of the defer countdown. In order to avoid starvation, ChaPLeT *tuner* factors in the waiting time introduced by others accessing the medium. For each packet, using the timestamp of the arrival of the packet, the *tuner* calculates the current wait time, t_{wait}, and compares this value with the expected wait time, t_{exp}. The t_{exp} is determined using the packet size, expected throughput, and data rate currently being used.

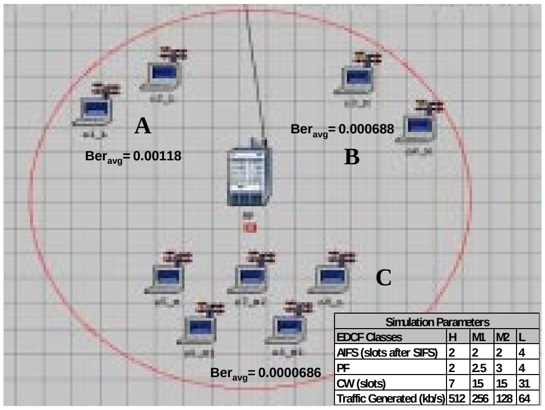
The packet is flagged when t_{wait} exceeds t_{exp} significantly and the node has experienced a large number of deferrals n_{def}. When a packet has been flagged, the adaptation algorithm speeds up the countdown process by changing the rate at which the countdown continues. This will expedite the channel access when the waiting period caused by deferrals exceeds tolerable limit for the node. Note that this policy mainly affects the *Defer Period* since the t_{exp} is assumed to have factored in the time spent in the *Access Period*. Since this policy is only triggered when starvation occurs, this adaptation is more likely to be active in a low priority node.

V. EXPERIMENTAL RESULTS

In this section, we present the experimental evaluation of our adaptation policies and the proposed framework. We briefly describe the experimental setup and system configuration. Next, we present the results demonstrating the effectiveness of the ChaPLeT framework.

A. Setup and System Configuration

For our experiments, we use the *Opnet* simulation environment [9]. In order to simulate the IEEE 802.11e standard, we modified the IEEE 802.11b MAC layer models to incorporate EDCF service differentiation. We utilized the *Opnet* Direct Sequence Spread Spectrum (DSSS) physical layer with a data rate of 11 Mb/s. The *Opnet* tool models channel error and channel variation using a pipeline model, which evaluates characteristics such as propagation loss and interference.



Simulation Parameters				
EDCF Classes	H	M1	M2	L
AIFS (slots after SIFS)	2	2	2	4
PF	2	2.5	3	4
CW (slots)	7	15	15	31
Traffic Generated (kb/s)	512	256	128	64

Fig. 3. Simulation Topology

To evaluate the effectiveness of the ChaPLeT adaptation policy, we used the topology of the simulated network illustrated in Figure 3. The topology can be segmented into three different regions *(A, B, C)*. There are two low priority nodes (n1_L, n2_L) in region *A*, two high priority nodes (n3_H, n4_H) in region *B*, and five nodes with varying priorities (n5_H, n6_M1, n7_M2, n8_M1, n9_L) in region *C*. The default characteristics of the simulated wireless network and parameters have been chosen similar to those that presented in [10], which provides a table of typical EDCF settings.

With the above topology, starvation occurs because the low priority nodes suffer from poor channel conditions and a large number of deferrals in region *A*. Priority inversion occurs with high priority nodes facing poor channel conditions in region *B*. Region *C* consists of nodes with varying priorities that are experiencing good channel conditions. Figure 4 provides the average achieved throughput for a high priority node (n4_H), a low priority node (n1_L) and a medium node (n6_M1). It can be noted that the high priority node achieves a lower throughput than the medium priority node and the low priority node suffers from starvation.

B. Adaptation Policies

We now evaluate the effectiveness of the ChaPLeT adaptation policy on improving QoS and service differentiation in varying channel conditions. For our first set of experiments, we focus on

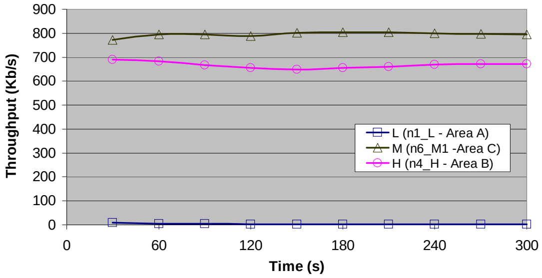


Fig. 4. Effect of Channel Variation on Achieved Throughput

the effectiveness of each of the tunable parameters. By activating one parameter at a time, we observe a gradual improvement in the stations' achieved throughput. Figure 5 shows the average throughput of one of the high priority nodes (n4_H) suffering from poor channel conditions over the simulated period with various levels of adaptation. In this simulation, the improvement of fragmentation adaptation alone is approximately 15%, fragmentation and persistence factor is 20%, and with all parameters, the improvement is 24%.

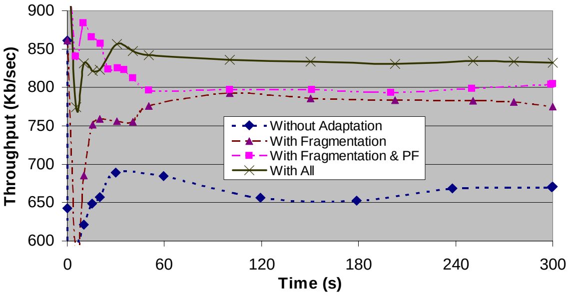


Fig. 5. Effect of Tunable Parameters on Achieved Throughput

Having observed a positive effect by combining the individual parameter adaptations, we next evaluate the effectiveness of ChaPLeT over the entire network. Figure 6 illustrates the average achieved throughput with ChaPLeT adaptation for the same nodes (n1_L, n6_M1, and n4_H) observed in Figure 4. It can be seen that with the ChaPLeT adaptation, the achieved throughput for the high priority node is greater than that of the medium priority node, indicating that priority inversion has been avoided. Furthermore, the figure illustrates that the low priority node is able to avoid starvation.

Table I summarizes the percentage improvement of the average throughput of selected nodes. For each node, the table provides information about its priority, BER, and average throughput delivered with and without adaptation. Note that the throughput achieved by the high priority nodes, n3_H and n4_H, is improved by 22% and 20% respectively. Furthermore, the low priority nodes, n1_L and n2_L, achieve a significant improvement (43x) in throughput and avoid starvation. This is possible through the use of the fragmentation threshold and defer count adaptation. For the medium priority nodes, we observe a slight decrease in average throughput by 7% and 2% for n6_M1 and n8_M1 respectively. This is an artifact of the ChaPLeT adaptation and the fair allocation of resources among other nodes in the network.

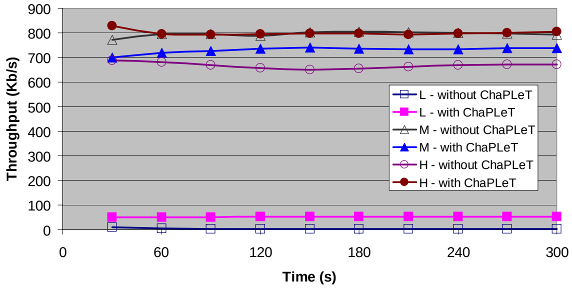


Fig. 6. Achieved Throughput with ChaPLeT

TABLE I

PERCENTAGE IMPROVEMENT WITH CHAPLET

		Avg Throughput (Kb/s)		
Priority (node)	**BER**	**Without ChaPLeT**	**With ChaPLeT**	**Percentage Change**
H (n3_H)	6.79E-04	680.065	830.506	22.12%
H (n4_H)	6.97E-04	670.443	804.793	20.04%
M1 (n6_M1)	6.89E-05	793.945	738.337	-7.00%
M1 (n8_M1)	6.86E-05	813.006	797.297	-1.93%
L (n1_L)	1.16E-03	1.175	51.301	4266.04%
L (n2_L)	1.20E-03	1.231	52.954	4201.71%

VI. CONCLUSION AND FUTURE WORK

In this paper, we showed the impact of tuning selected parameters to provide and maintain service differentiation in the presence of channel variation. We described the problems caused by channel variance, such as priority inversion and starvation. We then presented an adaptation framework, ChaPLeT, to dynamically configure three parameters, fragmentation threshold, persistence factor, and defer count, to overcome these problems. Through our experimental results, we demonstrated that ChaPLeT outperforms IEEE 802.11e and improves service differentiation in scenarios with varying channel conditions. Our future directions include exploration of other parameters in IEEE 802.11e and providing QoS in a centralized approach using Hybrid Coordination Function (HCF).

REFERENCES

[1] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard, June 1999.

[2] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 209–18.

[3] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 582–590.

[4] Daji Qiao and K. G. Shin, "Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 wlan under the DCF," in *Proceedings of IEEE International Workshop on Quality of Service*, Miami Beach, FL, May 2002, pp. 227–36.

[5] A. Banchs and X. Perez, "Providing throughput guarantees in IEEE 802.11 wireless lan," in *Proceedings of ACM WCNC*, Orlando, FL, Mar. 2002, pp. 130–8.

[6] Daji Qiao and K. G. Shin, "UMAV: A simple enhancement to the IEEE 802.11 DCF," in *Hawaii International Conference on System Sciences*, Hawaii, Hawaii, january 2003.

[7] IEEE 802.11 WG, "Draft supplement to standard for telecommunications and information exchange between systems - LAN/MAN specific requirements - part 11: MAC enhancements for quality of service (QoS)," IEEE 802.11e Standard Draft/D2.0, 2001.

[8] Naomi Ramos, Debashis Panigrahi, and Sujit Dey, "Energy-efficient link adaptations in ieee 802.11b wireless lan," in *Proceeding of IASTED Intl. Conf. on Wireless and Optical Communications*, 2003, pp. 578–583.

[9] "Opnet simulation framework," `http://www.opnet.com`.

[10] Dajing Gu and Jinyun Zhang, "Qos enhancement in ieee 802.11 wireless local area networks," in *Proceeding of World Wireless Congress (3G Wireless)*, 2003.