

A Contention Window Based Differentiation Mechanism for providing QoS in Wireless LANs

Mayank Mishra and Anirudha Sahoo

Kanwal Rekhi School of Information Technology

Indian Institute of Technology, Bombay, Powai, Mumbai, India 400076.

email: {mayank, saho} @it.iitb.ac.in

Abstract

Running real time applications over wireless LANs is becoming common place. These applications require QoS. But the most widely used wireless LAN, IEEE 802.11, does not have QoS support. Hence, providing QoS in 802.11 WLANs is an important issue due to its large installation base. In this paper, we propose a priority based service differentiation mechanism at the MAC layer. The MAC assigns different contention windows to two priority classes to provide service differentiation. The MAC protocol was designed such that minimal change is required in 802.11 firmware and yet the performance is comparable to 802.11e MAC. When collision occurs, contention window is increased in a linear fashion and the new contention windows for high and low priority traffic become non-contiguous. This unique method of contention window management provides better relative performance between the two classes. We also propose an enhancement to our base protocol that further increases throughput at light load condition. We present our simulation experiment results that show that our protocol has performance comparable to 802.11e.

1. Introduction

Real time applications running over a network have stringent requirements in terms of delay, bandwidth and other QoS parameters. Hence, QoS should be provided by the underlying network to such applications. One way to achieve this is to provide QoS at the MAC layer. Since wireless LANs (WLAN) are very common nowadays, users expect these real time application to run over WLANs and get the required QoS. But IEEE 802.11, which is the most prevalent WLAN technology, does not have any inherent QoS support. So IEEE formed the 802.11e [3] task group which came up with a priority based CSMA/CA scheme to provide differentiated services across different types of applications.

1.1 IEEE 802.11 and Other Similar MAC to Provide QoS

The IEEE 802.11 [2] MAC uses DCF (Distributed Coordination Function) for media access among the participating

network nodes. But DCF alone is neither capable nor suitable for fulfilling the QoS requirements of realtime applications like voice and video. It does not provide any priority and there is no service differentiation between different flows. Generally, the QoS schemes proposed based on IEEE 802.11 try to improve DCF functionality. There are primarily three ways in which QoS is provided by modifying DCF based MAC:

- Prioritisation among different classes of data: Most of the techniques use different Inter Frame Space (IFSs) or different Contention Window (CWs) or both [12, 11, 8].
- Resource allocation to prioritized classes of data: This is achieved by some distributed variant of Weighted Fair Queuing (WFQ) [13, 9].
- Admission control: Measurement and model based admission control mechanisms are used to provide QoS [14, 7, 10].

1.2. IEEE 802.11e

The IEEE 802.11e MAC employs a channel access function, called Hybrid Coordination Function (HCF), which includes a contention based channel access known as Enhanced Distributed Channel Access (EDCA) and a contention free channel access mechanism. EDCA has four Access Categories (AC). Each AC obtains a differentiated channel access due to varying amount of time an AC would sense the channel to be idle and different length of the contention window size during backoff. EDCA supports eight different priorities, which are further mapped into four ACs. Access Categories are achieved by differentiating the arbitration interframe space ($AIFS$), the initial window size, and the maximum window size. For the AC $[i]$ ($i = 0; \dots; 3$), the initial backoff window size is $CW_{min}[i]$, the maximum backoff window size is $CW_{max}[i]$, and the arbitration interframe space is $AIFS[i]$. Each AC acts as an independent virtual MAC entity and performs the same DCF function, with a different interframe space ($AIFS[i]$), and a different Contention Window. Each AC has its own backoff counter ($BO[i]$), which is independent of others. If more than one

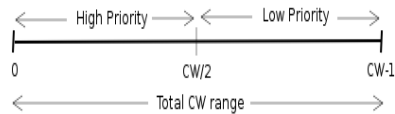


Figure 1: Different CW Ranges for Different Priorities

AC finishes the backoff at the same time, the highest priority AC frame is chosen for transmission by the virtual collision handler. Other lower priority AC frames go to the next round of backoff. Obviously, IEEE 802.11e is a much better protocol than IEEE 802.11 in terms of providing QoS. But it is a new standard. Almost all of the current WLAN infrastructure is 802.11 based. Thus, QoS mechanisms based on IEEE 802.11 have to be looked at until 802.11e is available. But such solution should require minimal change to the 802.11 MAC and yet provide good performance for real time applications. We propose one such MAC protocol. Our scheme provides priority to real time flows by using contention window based service differentiation method. In keeping with minimal change philosophy, we do not propose to have differentiation using different IFSs. But it has been found that providing priority through IFS based techniques is more effective than Contention Window (CW) based techniques [4]. This is because such protocols get deterministic differentiation in access times of different priority classes, whereas in CW based schemes, differentiation is probabilistic. Since 802.11e employs both IFS based and CW based priority it is quite effective in providing service differentiation. Hence, we did not expect our scheme to outperform 802.11e, but our protocol is carefully designed to make the performance comparable to 802.11e. Simulation experiments show that our protocol is quite comparable to 802.11e when only voice traffic is supported in the WLAN. When a mix of voice and video is supported, our protocol performs better than 802.11e in some scenarios.

2 Description of our protocol

2.1. Basic Mechanism

Our protocol provides two levels of priorities. High priority can be used by real time applications like voice and video, whereas low priority would be used by regular best effort based application like email, FTP etc. Our protocol uses contention window based differentiation mechanism to provide priorities to traffic flows. Basically, it specifies two different contention window (CW) ranges for two priority levels. As shown in Figure 1, the high priority class occupies the lower half of the Contention Window range, whereas the low priority occupies the upper half.

Higher priority class chooses its backoff from the lower half of the Contention Window range. Hence higher priority traffic has a smaller backoff interval than the lower priority

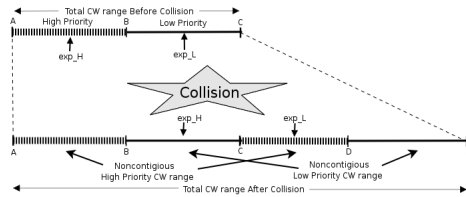


Figure 2: Effect of Collision on the Contention Window Ranges for Different Priority Classes

traffic. Thus, the average delay of high priority traffic should be less than that of low priority traffic. Moreover, since the delay is low for higher priority class, it receives relatively higher throughput. Thus, this MAC protocol basically provides better quality of service to the higher priority class at the cost of service of the lower priority class.

2.2. Collision Handling

In IEEE 802.11 DCF MAC protocol, when a collision occurs, the contention window range is doubled. The stations involved in collision then choose the backoff value from the larger range, which lowers the probability of collision. In our scheme, however, the CW range is increased in a linear fashion. Thus, after every unsuccessful transmission attempt the current CW range is increased by CW_{min} . Further, while the CW range is increased, the individual CW ranges of priority classes become noncontiguous as shown in Figure 2. This Figure shows the effect of collision on the contention window of individual priority classes. Before collision, the contention window of high priority flow is from A to B and of low priority is from B to C. After collision, the contention window of both the priority classes become noncontiguous i.e., high priority class gets CW range from A to B and from C to D, whereas low priority class gets CW range from B to C and from D to E. Now we provide the rationale behind two major aspects of our protocol.

2.2.1. Linear Increase

In IEEE 802.11 and IEEE 802.11e the increase in CW size after collision is exponential. This decreases the probability of further collision between the same stations. But in our scheme, we increase the CW size linearly. It has been reported that the probability of stations going through four or more successive collision is negligible [5]. Even the probability of having three successive collision is quite low. Moreover, the first two rounds of backoff in exponential and linear increase scheme will have the exact same contention window size. Hence, the performance difference between the two schemes may not be that significant. Linear increase in contention window size helps reducing the delay difference between packets sent from different rounds of backoff, while reducing the probability of collision in subsequent round.

2.2.2. Noncontiguous Contention Windows

In our protocol, contention windows of the two priority classes become noncontiguous after collision. This has the

following advantages:

- Under high load condition, if a high priority and a low priority frame have collided, then they will not collide again.
- The difference between expected values of backoff for high and low priority remains same irrespective of the round of retransmission. This has the nice property that the relative delay performance of the two classes still remain the same in subsequent backoff rounds. This is illustrated in Figure 2 by exp_H and exp_L .

2.3. Effect of Backoff Countdown on Low priority class

In the proposed scheme, the frames belonging to low priority class always choose a higher backoff value than the high priority frames. This may seem unfair for the low priority flows. Although the low priority frames have to wait longer than the high priority ones, they eventually get a chance to transmit. Figure 3 shows the process of countdown of backoff counter of low priority traffic. The low priority frames initially choose a backoff value from $CW/2$ to $CW-1$, but after countdown of $CW/2$ backoff slots, the backoff counter takes a value between 0 to $(CW/2) - 1$, which is the contention window range for high priority class. Thus, the lower priority frame effectively becomes equivalent to a high priority frame when its backoff counter counts down to a value which is in the range of high priority traffic. This dynamic change in priority helps maintain fairness to the low priority traffic.

2.4. Management of Contention Window

As mentioned earlier, we provide service differentiation based on contention window assigned to two priority classes. Apart from this, our protocol is very similar to IEEE 802.11 DCF MAC protocol. Let CW_i denote the total contention window size in a backoff round i . When $i = 0$, $CW_i = CW_0$ is the minimum total contention window size. For this study, we have taken it as 32, which is the default for IEEE 802.11 DCF. Since we increase total contention window linearly, CW_i is given by

$$CW_i = (i + 1) * CW_0 \quad (1)$$

Note that this CW_i is the size of the total CW range. This is divided into individual ranges of the two priority classes. Now consider the i^{th} backoff round. The total CW range is CW_i and the non-contiguous contention window for the high priority class, denoted by CW_i^H is given by

$$CW_i^H = \begin{cases} 0 & \text{to } \frac{1}{3}CW_0 - 1 \\ CW_0 & \text{to } \frac{2}{3}CW_0 - 1 \\ 2CW_0 & \text{to } \frac{5}{3}CW_0 - 1 \\ \dots & \\ iCW_0 & \text{to } \frac{2i+1}{2}CW_0 - 1 \end{cases} \quad (2)$$

Similarly, the non-contiguous range of contention window for low priority class CW_i^L is given by

$$CW_i^L = \begin{cases} \frac{1}{3}CW_0 & \text{to } CW_0 - 1 \\ \frac{2}{3}CW_0 & \text{to } 2CW_0 - 1 \\ \frac{5}{3}CW_0 & \text{to } 3CW_0 - 1 \\ \dots & \\ \frac{2i+1}{2}CW_0 & \text{to } (i + 1)CW_0 - 1 \end{cases} \quad (3)$$

2.5. An Enhancement to Improve Resource Utilization

Keeping the contention window size completely non-overlapping between the high and low priority class may not be always good. For example, at low load condition, the low priority traffic will choose high backoff value due to the nonoverlapping backoff window. This leads to higher delay and lower throughput. At low load condition, allowing the low priority class to encroach into the CW of high priority class would improve its performance and may not hamper the performance of high priority class significantly. This is the main idea behind the enhancement proposed to our base protocol. Since high priority flow is delay sensitive, we do not allow the same encroachment of high priority CW into low priority CW. Thus, the total CW range remains fixed, but the individual CW range of low priority can step into or come back to its original boundary based on the network load. This dynamic nature of CW of low priority class is shown in Figure 4. The increase in traffic load in the network causes increase in the number of collisions. In other words, the number of collisions per unit time is an indicator of the load on the network. The dynamic size of the Contention Window range of low priority class depends on the number of collision experienced, normalized by number of transmission attempts in an observation period. Thus, we can have a case where, due to negligible number of collisions, the Contention Window range of low priority class stretches fully from 0 to $CW - 1$. In high load condition, however, the two CW ranges should not overlap. It may appear that decreasing the size of CW of lower priority in case of increasing load will make the situation worse as there will be lesser number of slots from where the lower priority stations can choose backoff. But it is essential so as to keep the higher priority class unaffected due to the lower priority. Let the overlap factor be denoted by Δ . This Δ can be defined as the amount of CW which the low priority class encroaches into the region of high priority. Thus we have,

$$0 \leq \Delta \leq \frac{CW_0}{2} \quad (4)$$

The non-contiguous range for low priority class CW_i^L with this enhancement is given by

$$CW_i^L = \begin{cases} \frac{1}{3}CW_i - \Delta & \text{to } \frac{CW_i}{3} - 1 \\ \frac{2}{3}CW_i - \Delta & \text{to } 2 * \frac{CW_i}{3} - 1 \\ \frac{5}{3}CW_i - \Delta & \text{to } 3 * \frac{CW_i}{3} - 1 \\ \dots & \\ \frac{2i+1}{2}CW_i - \Delta & \text{to } (i + 1) * \frac{CW_i}{3} - 1 \end{cases} \quad (5)$$

2.5.1. Calculation of Overlapping Factor

The overlapping factor Δ depends on the number of collisions in the network. Let k be the number of transmission attempts, c be the number of collisions in last k attempts and t be the collision threshold. The overlapping factor depends inversely on the value of f , which is defined as c/k . When $f = 0$, there is complete overlap, i.e. $\Delta = CW_0/2$. As f increases, Δ decreases and finally Δ becomes 0 when $f \geq t$.

3 Performance Evaluation

In this Section, we present performance results of our proposed protocol in different application scenarios. We have used Opnet [1] simulation software for our experiment. Since 802.11 does not provide any priority, it is obvious that

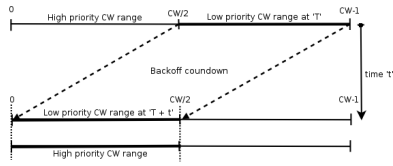


Figure 3: Transition of Low Priority Traffic to High Priority Due to Countdown

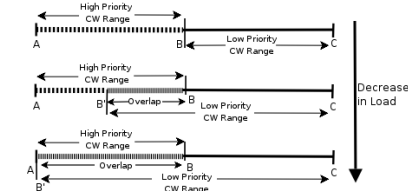


Figure 4: Effect of Network Load on Contention Window Range of Low Priority

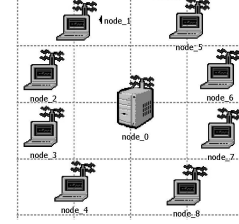


Figure 5: Simulation Topology in SQS Mode

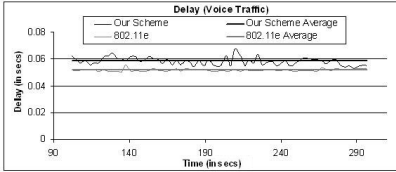


Figure 6: Avg. Delay of Voice Traffic (Scenario A)

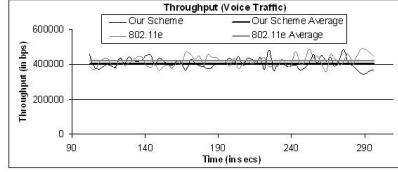


Figure 7: Avg. Throughput of Voice Traffic (Scenario A)

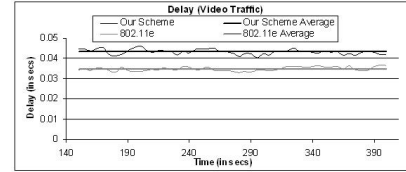


Figure 8: Avg. Delay of Video Traffic (Scenario B)

our protocol will be better than 802.11. Hence, we have compared our scheme with 802.11e only. As our scheme is based on 802.11, we modified the existing implementation of 802.11 in Opnet to build our protocol. The amount of code change to realize our protocol in Opnet was very small (less than 10 lines of code). This implies that our protocol can easily be implemented in real 802.11 hardware with a small amount of change in the MAC firmware. In our experiments we have designated some nodes as high priority nodes and others as low priority nodes. High priority nodes send out high priority traffic (e.g., voice), whereas low priority nodes transmit low priority traffic (e.g., best effort). To load the network with high priority traffic, we selectively assign real time applications like Voice and Video to high priority nodes. In 802.11e we have not used AC[0], the least priority class. Instead, we have used AC[1] as the lowest possible priority. This implies that our protocol can easily be implemented in real 802.11 hardware with a small amount of change in the MAC firmware. The topology used in our simulation is shown in Figure 5. It consists of 8 wireless nodes and 1 access point. The parameters like CW_{min} , CW_{max} and IFS are configurable in 802.11e, but for our simulation, we have taken their default values as shown in Table 1. Table 2 shows the parameter values used in our protocol. The specifications of applications and their associated priorities are given in Table 3. We ran our experiment in different scenarios as explained below.

3.1. Scenario A

This scenario involves

- A mix of low priority applications (see Table 3) to each of the four low priority nodes
- Four voice applications assigned to four different high priority nodes. The load in the network was around 3.5 Mbps.

The average end-to-end delay experienced by voice applications was measured in high and low load situations (Figure

6). Note that this delay is measured at network aggregate level and not on per node basis. Under this condition, it is clear that 802.11e performs better than our protocol. But the delay is only about 8% less than our scheme. This is due to the priority mechanism of 802.11e through IFS differentiation. The average delay in our scheme is about 58msec, which is pretty good for voice applications, considering that upto 150msec one way delay provides good voice quality. Figure 7 shows the performance of voice application in both 802.11e and our scheme in terms of throughput. Throughputs of the two protocols are almost equal. Since the offered load is within the capacity of the system and the difference of average delay between the two protocols is not significant, in a given observation interval, there is no perceived difference between the throughputs.

We also ran the same experiment in low load (1Mbps) condition. We observed that our protocol and 802.11e performed almost equally in terms of delay and throughput. This was because, at low load, there were very few collisions and most of the time the medium was found to be free. Hence the stations did not go into backoff. The Throughput of Best Effort (low priority) traffic in both protocols were also almost equal in both high and low load condition. We have not shown the graphs due to space limitation.

3.2. Scenario B

This scenario involves

- One Video streaming application assigned to one high priority node.
- A mix of low priority applications to each of four low priority nodes.

The total load on the network was approximately 6 Mbps and average end-to-end delay experienced by Video application was measured (Figure 8). The delay in our scheme is nearly 15% higher than that of 802.11e. In this situation AIFS based service differentiation of 802.11e is responsible for its better performance. Throughput of Video traffic (Figure 9) under the two protocols is almost the same.

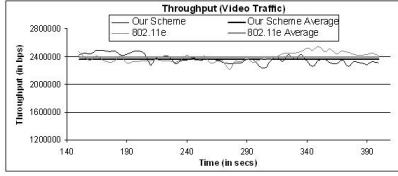


Figure 9: Avg. Throughput of Video Traffic (Scenario B)

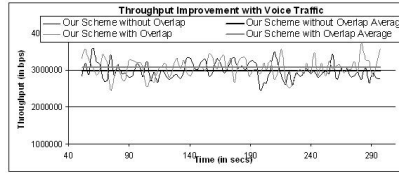


Figure 10: Improvement in Resource Utilization of Best Effort With Voice Traffic

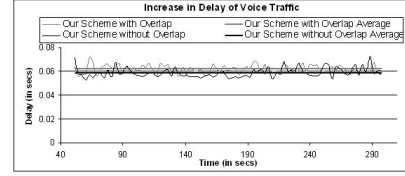


Figure 11: Slight Increase in Delay of Voice Traffic Due to Overlap

Type	AC	IFS	CW_{min}	CW_{max}
Voice	AC[3]	2	8	16
Video	AC[2]	2	16	32
BE	AC[1]	3	32	1024
BG	AC[0]	7	32	1024

Table 1: 802.11e parameters

Type	Priority	IFS	CW_{min} (noncontiguous)	CW_{max}
Voice	High	2	16	1016
Video	High	2	16	1016
other	Low	2	32	1024
			(16-31)	

Table 2: parameters used in our protocol

Application	Priority	Specification
Voice over IP	High	64Kbps, G.711
Video Stream	..	10fr/sec, 1.38Mbps
FTP	Low	varying
Telnet	..	varying
Database
Email, Web

Table 3: Specification of Applications

3.3. Improvement in Resource Utilization by using Overlapping CWs

As discussed in Section 2.5 there is an improvement in the throughput of the BestEffort or Low Priority traffic when the overlapping enhancement is utilized. Although, due to improved performance of low Priority, high Priority suffers, but that is negligible as the load in the system is low. Figure 10 shows the performance improvement of Best Effort traffic and Figure 11 shows the slight increase in delay for Voice traffic due to overlap. The threshold is set to be $t = 0.32$.

4. Conclusion and Future Work

We have presented a 802.11 based MAC protocol that provides QoS to real time traffic like voice and video applications. Our MAC protocol provides priority to real time traffic by using contention window based service differentiation method. This has been achieved by making minimal changes to 802.11 MAC. We also presented enhancement over our base protocol to increase throughput in light load situation. Our simulation experiments show that performance of our protocol is comparable to 802.11e when voice applications are running in the network. For video applications, performance of our protocol is almost equal to that of 802.11e.

In this study, we have assumed that a priority is assigned at a node level i.e., a high (low) priority node sends out only high (low) priority traffic. In future, we would like to relax this constraint and allow a node to carry traffic of both the priorities. We are also planning to carry out analytical modeling of our protocol and evaluate its performance using the model. For the enhanced version of our protocol, we would also like to explore other methods of determining the overlap factor as the offered load changes in the network.

References

[1] Opnet network simulator. <http://www.opnet.com>.
 [2] Wireless lan medium access control(mac) and physical layer (phy) specifications. IEEE 802.11 Standard, 1997.

[3] Wireless lan medium access control(mac) and physical layer (phy) specifications. IEEE 802.11e Standard, 2005.
 [4] G. Bianchi, I. Tinnirello, and L. Scalia. Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. *IEEE Network*, 19(4):28–34, July/Aug 2005.
 [5] M. Bottigliengo, C. Casetti, C. F. Chiasserini, and M. Meo. Short-term fairness for tcp flows in 802.11b wlans. *IEEE Infocom*, 2004.
 [6] J. Deng and R. Chang. A priority scheme for IEEE 802.11 DCF access method. pages 96–102. *IEICE Transactions in Communications*, 1999.
 [7] S. Garg and M. Kappes. Call admission control for 802.11. In *Proc of. GLOBECOM*, 2003.
 [8] A. Nafaa, A. Ksentini, and A. Mehaoua. Scw: Sliding contention window for efficient service differentiation in 802.11 networks. In *Proc of. IEEE WCNC 2005, The IEEE Wireless Communication and Networking Conference.*, 2005.
 [9] W. Pattara-atokom, P. Krishnamurthy, and S. Banerjee. Distributed mechanisms for quality of service in wireless lans. pages 26–34. *IEEE Wireless Communications*, June, 2003.
 [10] D. Pong and T. Moores. Admission control for voip traffic in ieee 802.11 contention access mechanism. pages 174–178. In *Proc of. GLOBECOM*, 2003.
 [11] L. Romdhani, Q. Ni, and T. Turletti. Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks. *IEEE WCNC'03(Wireless Communications and Networking Conference)*, March 16-20 2003.
 [12] L. Romdhani, Q. Ni, and T. Turletti. A survey of qos enhancements for ieee 802.11 wireless lan. *Journal of Mobile Communications and Mobile Computing*, 4(5):547–566, 2004.
 [13] N. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless lan. Boston,USA, August 2000. *Annual International Conference on Mobile Computing and Networking (Mobicom 2000)*.
 [14] Y. Xiao and H. Li. Evaluation of distributed admission control for the 802.11e edca. *IEEE Radio Communications*, pages S20–S24, September 2004.