Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

Providing Quality of Service in the IEEE 802.11 Distributed Coordination Function

A Kazmierczak
Computer Information Systems, Northwest Arkansas Community College, One College Drive, Bentonville, AR 72712
akazmierczak@nwacc.edu

Abstract - Wireless networks are becoming more and more common in today's networking environment. One of the earliest of the IEEE standards for wireless networks is the IEEE 802.11 standard. Most communications in an 802.11 network use the Distributed Coordination Function (DCF). Unfortunately, the 802.11 DCF was not designed with Quality of Service (QoS) in mind. However, as more and more applications tend to be run on the 802.11 network, there is a growing demand for QoS, even in the DCF. Recently researchers have emulated a shortest job first scheduling policy by using a new resetting backoff technique in DCF. In this paper, we take advantage of this new backoff technique to provide a new level of QoS to the DCF of the IEEE 802.11 wireless networks. To the authors knowledge this is the first work to address scheduling Diffserv traffic classes in a wireless network environment.

Keywords – Distributed Coordination Function, Quality of Service, Packet Schedulers

1. Introduction

Interest in wireless Local Area Network (LAN) technology has been growing rapidly in recent years. A number of standards have been proposed, primarily the IEEE 802.11 standard [1], the most widely implemented wireless technology. As 802.11 has been incorporated in a growing number of devices, there has been a broad range of new applications using a wireless medium. QoS is thus becoming a critical issue.

The Distributed Coordination Function (DCF) is the medium access control (MAC) protocol that is mandatory for all 802.11 implementations [1]. DCF is a distributed contention based MAC protocol which was designed without QoS requirements in mind. Though the 802.11 standard also calls for a centralized MAC protocol, the Point Coordination Function, it is preferable to use the DCF for scheduling for several reasons. First, the communications with a central controller adds significant overhead that is really not needed. Second, stations do not need to communicate with a central controller to maintain connectivity. Third, when using the distributed mode, there is no need for a coordinator. Thus, it is more beneficial to develop distributed protocols that can provide the necessary QoS.

The IEEE 802.11 working group was formed to address support for QoS. The Enhanced DCF (EDCF) was developed with features that support some QoS. This is accomplished by assigning priorities to different traffic classes, high priority traffic accesses the channel before low priority traffic [2]. EDCF performance is comparable to a centrally controlled MAC [3].

The motivation for our work is to provide a means to support multiple traffic classes, giving higher priority to certain classes and lower priority to other classes and giving better access to higher priority classes. We use the Diffserv standard [4] to categorize the classes of traffic. Diffserv categorizes traffic into three priority classes: Expedited Forwarding (EF) is high priority, Assured Forwarding (AF) is medium priority, and Best Effort (BE) is low priority.

DCF does not provide support for these traffic classes, therefore all three classes have to contend for access to the medium. If all nodes use the same algorithm and mechanism for accessing the medium, there is no way to give priority access to any class of traffic. Thus, the delays seen by EF traffic is in part cause by AF and BE traffic gaining access to the medium, creating contention unfairness.

Very little research has appeared addressing the contention unfairness of DCF. Most of the research has centered around the size of packets. The packet size is of considerable concern because large packets, once they have accessed the medium can force undue transmission delay for small packets [5]. Distributed Far Queuing (DFQ) [6] has been proposed to give flows a fair share of the bandwidth, as far as packet size is concerned. This work does not address the aspect of QoS that we are concerned with in this paper.

However, this line of research did result in the implementation of a Shortest Job First (SJF) scheduling algorithm [7]. In that work, the authors realize SJF scheduling by proposing a new resetting backoff scheme. It is this new resetting backoff scheme that lends itself to a QoS scheduling scheme.

The rest of this paper is organized as follows. Section 2 describes the DCF MAC protocol for IEEE 802.11 and describes the new resetting backoff scheme as it was used to implement SJF scheduling. In section 3 we describe how to provide different QoS to different priority traffic classes by manipulating the resetting backoff scheme. Section 4 presents a simplified analytical performance measure. Section 5 presents numerical results. Section 6 includes our conclusions.

2. DCF In 802.11

In this section we briefly describe the 802.11 DCF MAC protocol and the new resetting backoff scheme.

2.1 IEEE 802.11 DCF MAC

The 802.11 DCF relies on sensing the wireless medium. A node with a packet to send senses the channel for activity. If the channel is idle for a time interval equal to a distributed inter frame space (DIFS), then the node transmits its packet. If the channel is sensed busy, the node continues to monitor the channel until it is idle for more than a DIFS. The node generates a random backoff interval before transmitting. This backoff procedure is intended to decrease the probability of a collision with packets transmitted by another node. A node chooses a new backoff time after each successful transmission. Though the backoff procedure is intended to avoid collisions, it can also be used to schedule transmissions in a distributed manner.

The backoff counter is decremented every slot after the channel is sensed idle for more than a DIFS. The backoff counter is frozen when the channel is busy. After each packet transmission, the backoff time is uniformly chosen in the range [0, w-1], where w is the contention window. At the first transmission attempt, w is set equal to the value of the minimum contention window, CW_{min}. At each unsuccessful transmission attempt, the contention window is doubled, up to a maximum size CW_{max} to help reduce the probability of another collision. This scheme is called binary exponential backoff.

2.2 The Virtual Clock in 802.11

Centralized schedulers maintain a virtual clock at the coordinator. Using this virtual clock, start and finish tags are allocated to each packet. Schedulers differ in how they update the virtual clock and how they order packets for service, either by the start tag or the finish tag. In fully distributed systems, each node contains its own virtual clock. We briefly describe the distributed virtual clock mechanism in DCF. For simplicity of explanation, we assume that no collisions occur and no packets are lost.

The virtual times of packets are computed when they reach the head of the queue. Let $V_i(t)$ denote the virtual time at actual time *t* for node *i*. Let S_i^k denote the virtual start time of the k-th packet of flow I, and let F_i^k denote its finish time. A_i^k is the real time the packet arrives. Define B_i^k to be the number of slots lost to channel contention and D_i^k be the transmission time of the packet. Per the 802.11 standard, each packet is sent after a backoff period starting right after the previous packet is sent. Start and finish times are computed as

$$S_i^k = \max\{ V_i(A_i^k), (F_i^{k-1} + B_i^k) \}$$

$$F_i^k = S_i^k + D_i^k$$

The virtual clock at a node is initialized to 0, $V_i(0) = 0$. Every time a packet from node I starts transmission, the node sets its virtual clock to the finish tag of the packet. The virtual clock is incremented every time the channel is sensed idle after a DIFS interval. If the packet of another node is transmitted, this node slows down its virtual clock by the transmission time of the packet. If the virtual start time of a packet is reached, the packet is transmitted.

2.3 Emulated Distributed Sjf Mechanism

In this section we briefly describe the mechanism used to emulate a SJF protocol [7]. This is the work most relevant to ours because of the novel resetting backoff mechanism that comes into play for our Priority QoS scheduler.

The author's propose that the backoff of each packet transmitted be reset each time a packet was transmitted. With this mechanism, the virtual clocks of all nodes run using the same absolute time steps. To satisfy SJF, all small packets access the channel before larger packets. This is accomplished using a non overlappingbackoff window for different packet sizes. It is proposed to assign the window range [0, W_S] for small packets and a range [W_S, W_L] for large packets. The mechanism that will make QoS possible for traffic classes is the assignment of the window range.

3. QOS BASED ON TRAFFIC CLASS

As mentioned earlier, we use the three traffic classes defined in Diffserv [4]: Expedited Forwarding (EF) for high priority, Assured Forwarding (AF) for medium priority, and Best Effort (BE) for low priority. To accommodate three traffic classes we propose to use three non overlappingbackoff window ranges, one range for each priority class. We propose a window range [0, W_H] for high priority traffic, a window range of [W_H, W_M] for medium priority traffic and a window range [W_M, W_L] for low priority traffic, where $W_H < W_M < W_L$. High priority traffic attempts to access the channel during the range [0, W_H], thus having the first opportunity to transmit. In the absence of high priority traffic, medium priority traffic attempts to access the channel during the range [W_H, W_M], thus getting access to the channel after high priority traffic but before low priority traffic. Low priority traffic can access the channel during the range [W_M, W_L], thus only getting

Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

access to the channel after high and medium priority traffic.

It is worth noting that once the transmission of a lower priority packet has begun, it cannot be preempted by higher priority traffic, but due to the choice of non overlapping window ranges, the high priority packet will get first opportunity to access the medium immediately after the transmission of the lower priority packet.

It is also worth noting that this scheduling mechanism does not eliminate the probability of collision of packets in the same priority class. It does eliminate the probability of collision of packets from different classes. With fewer flows in a particular priority class there should still be a reduced probability of collisions.

4. Analytic Performance

In this section we provide a simplified analysis of the transmission delay for EF, AF, and BE packets. Let us establish notations used. Let N_E be the mean number of nodes with EF traffic and let M_E be the mean number of EF packets per flow. The mean number of EF packets is $E = N_E \times M_E$. Let T_E be the mean transmission time for an EF packet and let W_E be the mean backoff time and DIFS is the idle time.

Let N_A be the mean number of nodes with AF traffic and let M_A be the mean number of AF packets per flow. The mean number of AF packets is $A = N_A \times M_A$. Let T_A be the mean transmission time for an AF packet and let W_A be the mean backoff time and DIFS is the idle time. Let N_B be the mean number of nodes with BE traffic and let M_B be the mean number of BE packets per flow. The mean number of BE packets is $B = N_B \times M_B$. Let T_B be the mean transmission time for a BE packet and let W_B be the mean backoff time and DIFS is the idle time.

At this point we make a major simplifying assumption to ease the analysis. We assume that no EF traffic arrives during the transmission of AF or BE traffic and we also assume that no AF traffic arrives during the transmission of BE traffic. We also assume no collisions. These assumptions do not reflect a realistic environment and are made simplify the analysis.

For an arbitrary EF packet for an arbitrary EF flow E_j, the transmission delay is

$$D_E = \sum_{i=1}^{j-1} (T_E \times E_i \times (W_E + \mathrm{DIFS}))$$

For an arbitrary AF packet for an arbitrary AF flow A_j, the transmission delay is

$$D_A = \sum_{i=1}^{j-1} (T_A \times A_i \times (W_A + \mathrm{DIFS})) + D_E$$

For an arbitrary BE packet for an arbitrary BE flow B_j, the transmission delay is

$$D_B = \sum_{i=1}^{j-1} (T_B \times B_i \times (W_B + \mathrm{DIFS})) + D_E + D_A$$

5. Numerical Results

In this section we provide numerical results for the transmission delay of EF, AF, and BE packets at positions 10, 20, 30, 40, and 50 out of 50 packets per class. For ease of calculation we let $T_E = T_A = T_B = 2$ time units. Also for ease of calculation we make the assumptions about the backoff range. Let DIFS = 3, $W_E = 1$, $W_A = 2$, $W_B = 3$ time units respectively. These numbers are not realistic but used for comparison purposes. Our results are shown in table 1.

Table 1 – Transmission Delays

Traffic Class	Packet Number	Transmission Delay
EF	10	72
AF	10	490
BE	10	618
EF	20	152
AF	20	570
BE	20	728
EF	30	232
AF	30	690
BE	30	848
EF	40	312
AF	40	790
BE	40	968
EF	50	392
AF	50	890
BE	50	1118

6. Conclusion

In this paper we presented a QoS scheduling mechanism for Diffserv traffic for IEEE 802.11 wireless networks. It is clearly shown that higher priority traffic sees significantly less transmission delay than lower priority traffic.

Reference

[1] IEEE, Inc., IEEE Standar 802.11 – Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1999

[2] D. Chen, et al, “Supporting VoIP traffic in IEEE 802.11 WLAN with enhanced medium access control for QoS”, Technical Report, Araya Labs, 2002

[3] A. Lindgren, A. Almquist, O. Shelen, “Quality of service schemes for IEEE 802.11 wireless LANs – an evaluation” Kluwer Mobile Networks and Applications, Vol 8, pp 223 – 235, 2003

[4] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, W. Weiss, “An architecture for differentiated services”, IETF RFC 2479

Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

[5] S. Choi, et al, "IEEE 802.11 Contention based channel access (EDCF) performance evaluation", IEEE ICC, 2003
[6] N. H. Vaidya, P. Bahl, S. Gupta, "Distributed fair scheduling in a wireless LAN", Sixth Annual International Conference on Mobile Computing and Networking, Aug 2000
[7] S. Pollin, A. Motamedi, A. Bahai, F. Catthoor, L. Vander Perre, "Delay Improvement of IEEE 802.11 Distributed Coordination Function Using Size based scheduling", IEEE, pp 3484 – 3488, 2005