

EDCF-DM: A Novel Enhanced Distributed Coordination Function for Wireless Ad Hoc Networks

Hao Zhu, Guohong Cao, Aylin Yener and Allen D. Mathias
The Pennsylvania State University
University Park, PA 16802
E-mail: {hazhu, gcao}@cse.psu.edu, yener@ee.psu.edu

Abstract—A set of enhancements to the IEEE 802.11 standard, viz. the IEEE 802.11e have been proposed to meet the increasing demand for Quality of Service. The standard provides means for service differentiation by using multiple traffic categories at each node, where each traffic category has its own individual parameters such as priority, inter-frame space and contention window size. After each successful transmission, the contention window size will be decreased based on a static equation, which may result in poor channel utilization and decrease system throughput. In this paper, we propose a new protocol, called Enhanced Distributed Coordination Function with Dual-Measurement (EDCF-DM), to address this issue. EDCF-DM is based on the idea of reducing the number of idle slots by dynamically varying the contention window size according to the current traffic state of the traffic categories at each node. Meanwhile, it carefully adapts the contention window size based on the network condition of the system to avoid incurring extra collisions. Extensive simulations are performed to evaluate the proposed protocol. Simulation results demonstrate that EDCF-DM provides a good service differentiation and outperforms the standard 802.11e in terms of channel utilization, throughput and packet delay.

I. INTRODUCTION

With the tremendous growth of applications available over wireless networks, it is envisioned that wireless access will be considered as another hop of the communication path. Since many applications have quality-of-service (QoS) requirements such as delay and throughput, it is imperative that the wireless part of the communication should be able to support QoS similar to wired networks [1]. To achieve such goals, the medium access control (MAC) protocol should provide an efficient mechanism to share the limited spectrum among all mobile nodes, together with simplicity of operation, high system throughput and good service differentiation for flows with different priorities.

The IEEE 802.11 [6] for wireless local area networks (WLANs) is one of the most widely deployed wireless techniques [3], [10]. It allows people to implement a wireless network in one of two possible configurations: the *infrastructure* mode or the *ad hoc* mode. Under the infrastructure mode, all nodes reside in a particular region where all communication must go through the access point. If the connection between a

node and the access point is lost, the node cannot transmit any packets. Under the ad hoc mode, all nodes can form an ad hoc network spontaneously without any centralized control. Even if a node loses direct connections with some nodes, it is still possible for the node to communicate with others through multi-hop connections. This feature allows ad hoc networks to be flexibly deployed in scenarios such as battlefields, emergencies etc., where no pre-established infrastructure exists.

The IEEE 802.11 standard includes a set of protocols that are responsible for medium access control. The basic access mechanism for ad hoc networks is the *distributed coordination function* (DCF), which uses the carrier sense multiple access with collision avoidance (CSMA/CA). However, the legacy DCF does not support adequate degrees of service differentiation [1], [2], [11], which is crucial for home networking, video-on-demand and real-time voice-over-IP applications. To address this issue, many medium access protocols have been proposed to enhance QoS provisions under DCF [1], [2], [11]. The priority of each flow is controlled by different backoff time of CSMA/CA. As a result, high priority flows can have more chances to grab the medium than low priority flows, and get better QoS. Following the same principle, the IEEE task force group has proposed a set of QoS enhancements to IEEE 802.11 [4], [8]. Among them, the Enhanced DCF (EDCF) protocol, which is an extension to DCF and is completely distributed, adds many new and necessary features to the current IEEE 802.11 standard. EDCF provides a flexible and distributed solution to service differentiation by introducing the concept of prioritized traffic categories. By assigning different inter-frame space and contention window size to different priority, the high priority flow is granted faster access to the medium than the low priority flow.

Similar to DCF, EDCF is a contention-based MAC protocol. Thus, it is important to carefully control the backoff time of each flow to achieve good system throughput and channel utilization. As shown in [7], [9], aggressively decreasing the backoff time may increase the number of collisions a lot and significantly decrease the system throughput. On the other hand, changing the backoff time too conservatively could cause low channel utilization. In this paper, we study

This work was supported in part by the National Science Foundation (CAREER CCR-0092770 and ITR-0219711).

how to improve the performance of EDCF in terms of the channel utilization and service differentiation in ad hoc wireless networks. We design a new protocol, called EDCF with Dual-Measurement (EDCF-DM), to achieve this goal. EDCF-DM can significantly increase the channel utilization by dynamically modifying the contention window size according to the collision rate and the current state of each traffic category. When the collision rate is high, the contention window size of each flow will be changed slowly to avoid further collisions. When the higher priority queue is empty, the contention window size of the low priority flow will be decreased faster than that when the high priority flow has packets to send, so that the wasted idle time slots can be reduced. Meanwhile, the service differentiation between traffic categories is maintained by monitoring the higher priority traffic and retaining the arbitration inter-frame space (AIFS) [4] of each traffic category. We evaluate the proposed protocol through extensive simulations. The simulation results show that, compared to the standard EDCF and the adaptive EDCF (AEDCF) [9], EDCF-DM achieves a marked improvement in throughput and channel utilization with a good service differentiation.

The remaining of the paper is organized as follows. Section II provides an overview of the legacy EDCF proposed in the 802.11e draft and the motivation of our work. Section III gives a detailed description of our protocol. The simulation results are given in Section IV. Section V concludes the paper.

II. BACKGROUND AND MOTIVATIONS

A. The Enhanced DCF of IEEE 802.11e

The EDCF is the contention-based channel access protocol. The proposed scheme provides capability for up to eight types of Traffic Classes or Categories (TC). It assigns smaller contention window (CW) sizes for higher priority classes and larger sizes to low priority classes, giving high priority classes small lower bound of CW sizes. IEEE 802.11e also proposes the use of different inter frame space (IFS) according to the priority of each TC. Instead of DIFS used in DCF, an *arbitration inter-frame space* (AIFS) would be used for each TC. The TC with the smallest AIFS will have the highest priority. Each AIFS is equal to the DIFS time plus some (possibly zero) time slots. A big difference between DCF and EDCF is that when the medium is detected as idle for a period of AIFS, the backoff counter is reduced by one at the beginning of the last slot interval of the AIFS period. While in the legacy DCF, the backoff counter is reduced by one at the beginning of the first slot interval after the DIFS period [8]. Each TC within every node behaves like a virtual node and independently contends for access to the medium. The backoff process is also carried out individually after detecting if the medium is idle for a time equal to its AIFS. Thus each TC_i is individually parameterized as: $AIFS_i, CW_{i,min}, CW_{i,max}$ and PF_i , where PF_i is the persistence factor of TC_i . Conflicts between virtual nodes within each node are resolved by granting access to the higher priority transmission. With EDCF, after each successful transmission of TC_i , the corresponding CW_i will be set to $CW_{i,min}$. Once a transmission fails, CW_i

will be calculated as follows:

$$CW_i = \min\{CW_{i,max}, CW_i * PF_i\} \quad (1)$$

After waiting for $AIFS_i$, each backoff timer is set to a random number from $[1, CW_i + 1]$ with the unit of time slot.

B. Motivations

For contention-based medium access mechanisms (i.e. DCF of IEEE 802.11), there are two major factors affecting the system throughput: transmission failure due to collisions and the wasted idle slots due to backoffs at each contention period. These problems exist in EDCF of IEEE 802.11e since it is contention-based. However, these two problems are inherently conflicted, which means reducing the backoff time could increase the number of collisions and vice versa. As a result, it is desirable to carefully control the backoff time at each contention period to achieve a good throughput.

Since the backoff time is directly related to the contention window size, we target at designing a good scheme to manage contention window under EDCF. Generally, the management of contention window needs to be improved from two aspects: First, as stated in [9], the window size should be adjusted according to the network condition. Specifically, when the system is heavily loaded, the value of CW_i should be adjusted more slowly rather than setting to $CW_{i,min}$ upon a successful transmission. Second, one new feature of EDCF is that different TCs are assigned different value of parameters and then have different priority. The performance obtained is not optimal since the parameters do not adapt to the traffic state of each TC. For example, suppose there are two TCs: TC_i and TC_j , where TC_i has higher priority than TC_j . When TC_i does not have data to send, CW_j can be decreased faster after each successful transmission. As a result, the number of wasted idle slots during backoff can be reduced, which increases system throughput and channel utilization. Based on the observations, it is important to design a new scheme that adapts the contention window size to both the traffic state of each TC at each node and the network condition of the system.

III. THE PROPOSED EDCF-DM PROTOCOL

In order to take into account the network condition and the traffic state for contention window adaptation, we design a dual measurement scheme to get the related online information. In this section, we describe the scheme and give the algorithm used by the EDCF-DM protocol.

A. Network Condition Measurement

It is well known that, with contention-based MAC protocol, the number of contentions could significantly increase as the system is heavily loaded. Similar to [9], we use the number of recent collisions as the indicator of the network condition. Specifically, the time domain is divided into continuous measurement windows (MWs) with the specified window size. When the j th measurement window, denoted by MW_j expires, the node summarizes the network condition indicator $\alpha(j)$ during MW_j by calculating:

$$\alpha(j) = \frac{n_j(\text{collisions})}{n_j(\text{packets successfully sent})} \quad (2)$$

where $n_j(event)$ is the number of the events occurred during MW_j . Since $\alpha(j)$ itself cannot precisely represent the long-term network condition, we also apply the exponentially

Initialization:

Set the parameters of each TC_i :
 $AIFS_i, CW_{i,min}, CW_{i,max}, PF_i$;

When the j th measurement window expires:

get $n_j(packet)$ and $n_j(collision)$;
 calculate $\alpha(j)$ according to Eq (2);
 update α_{avg} according to Eq (3);
for (each TC_i)
 calculate $\beta_i(j)$ according to Eq (4);
if ($n_j(collision) == 0 \ \&\& \ \beta_i(j) == 0$)
 update σ_i according to Eq (5);
else
 update σ_i according to Eq (6);

After a successful transmission in TC_i :

adjust CW_i according to Eq (7);

After a collision in TC_i :

adjust CW_i according to Eq (1);

Fig. 1. The EDCF-DM Protocol

weighted average method to smooth the measurement result of each measurement window. In particular, the average network condition indicator after MW_j , denoted by $\alpha_{avg}(j)$ is calculated by:

$$\alpha_{avg}(j) = \phi * \alpha_{avg}(j-1) + (1 - \phi) * \alpha(j) \quad (3)$$

where $0 < \phi < 1$. In this paper, we set ϕ to be 0.8. At a time, the network condition is reflected by the value of α_{avg} . A small α_{avg} means the network condition is good.

B. Traffic State Measurement

As stated in Section II-B, when the traffic density of each high priority TC is low, reducing the contention window size of the low priority TC could improve the system throughput and channel utilization. Similar to the principle of network condition measurement, each node maintains a traffic state indicator for each TC_i , denoted by β_i . During the j th measurement window, $\beta_i(j)$ is equal to the number of transmitted or received packets of each high priority TC. As one optimization, TC_i does not need to compute the number of transmitted packets of all higher TCs. Instead, suppose TC_i has lower priority than TC_j if and only if $i > j$, $\beta_i(j)$ can be computed by:

$$\beta_i(j) = n_j(\text{packets of } TC_{i-1}) + \sum_k \beta_k(j), k < i-1 \quad (4)$$

Since the traffic state of each TC may change quite fast (e.g., tens of milliseconds), unlike the computation of α_{avg} , we use the most recent value of β_i as the measurement result to capture the chance to adjust the contention window in time,

C. The EDCF-DM Protocol

With the measurements of network condition and traffic state, we propose a new dual-measurement based EDCF protocol, called EDCF-DM. The key idea of EDCF is that: Based on the measurement results, the extent of changing

CW_i is adapted. When the workload of the system is high, CW_i is changed slowly to avoid further collisions. For TC_i , if other high priority TCs have low traffic density, CW_i can be decreased faster to reduce the number of wasted idle slots. The speed is controlled by the control factor of TC_i , denoted by σ_i , which is dynamically assigned different values according to the measurement results. In particular, when there is no collision during the last measurement window and no higher TCs have packets to transmit or receive, σ_i is calculated by:

$$\sigma_i = \min\{(1 + (i * 2)) * \alpha_{avg}, \sigma_{min}\} \quad (5)$$

where σ_{min} is the specified system parameter. Otherwise, σ_i is calculated by:

$$\sigma_i = \min\{(1 + (i * 2)) * \alpha_{avg}, \sigma_{max}\} \quad (6)$$

where σ_{max} is the specified system parameter. After a successful transmission, according to σ_i , CW_i is changed as follows:

$$CW_i = \max\{CW_{i,min}, CW_i * \sigma_i\} \quad (7)$$

When a transmission fails due to collision, similar to standard EDCF, CW_i is updated according to Eq (1). The formal description of EDCF-DM is shown in Figure 1.

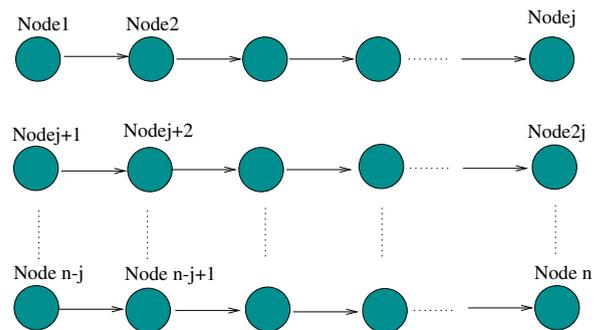
IV. PERFORMANCE EVALUATION**A. Simulation Setup**

Fig. 2. The simulation topology

We evaluate the performance of EDCF-DM through simulations by using ns-2 [5]. The simulation duration is 500 seconds, the channel capacity is set to be $36Mbps$, the slot time is $9\mu s$ and the SIFS is equal to $16\mu s$. The simulation topology is shown in Figure 2. In particular, there are n nodes in the area, and each node is assumed to stay within the transmission range of other nodes. Each node generates *three* different flows representing three uniquely prioritized traffic categories viz. high, medium and low. We use two constant bit rate (CBR) sources for the high and medium priority flows, and a best effort source for low priority traffic. The values of the parameters used for each traffic category are listed in Table I. As can be seen, the data rate for each node is at least $1088Kbps$, which is a sum of the CBR traffic categories at each node. Following our simulation setup, as the number of nodes increase, the number of flows keeps increasing by a factor of 3. As the number of nodes increases from 5 to 50, the workload of CBR flows changes approximately from 15% to 150% of the channel capacity.

We first determine the proper value of each parameter used in EDCF-DM. Then, we compare EDCF-UM with legacy

Parameters	High	Medium	Low
CW_{min}	5	15	31
CW_{max}	200	500	1023
AIFS(μs)	34	43	52
PF	2	4	5
Packet Size (bytes)	160	1280	1000
Packet Interval (ms)	20	10	n/a
Data Rate (Kbps)	64	1024	n/a

TABLE I
THE CONTROL OVERHEAD OF r PCF

EDCF [4] and the Adaptive EDCF [9]. The main difference between EDCF-DM and EDCF is that EDCF-DM adapts CW_i according to the network condition. The main difference between EDCF-DM and AEDCF is that EDCF-DM adapts CW_i with the consideration of the traffic condition. In addition to throughput and channel utilization, we use packet delay as one of the performance metrics. In the following subsections, we show the simulation results of throughput, packet delay and channel utilization respectively.

There are three important parameters of the EDCF-DM protocol: the measurement window size, σ_{max} and σ_{min} . With the results of [9], a good value of σ_{max} is 0.8. With extensive simulations, we choose the measurement window size and σ_{min} to be $1000 * \text{time slot}$ and 0.6 respectively.

B. Throughput Comparisons

In this subsection, we show the throughput of these three different schemes. For the purpose of clearance, we give the results of each traffic category separately. The simulation results are shown in Figure 3. As shown in Figure 3 (a), there is only a small change in throughput for the high priority traffic. This is because we avoid any modification of parameters for the high priority class, which acts as a reference to all the other lower priority classes. Any visible improvements on the highest priority class can be attributed to the adaptation mechanism according to the network condition, which decreases the probability of collision when the workload is high. For example, when the number of nodes is 50, both AEDCF and EDCF-DM achieve better throughput than EDCF.

From Figure 3 (b), we can see that, for the medium priority traffic, the throughput of EDCF-DM is better than that of EDCF and AEDCF when the number of nodes is greater than or equal to 15. Especially, when the number of nodes varies between 15 and 35, which means the workload is medium, the throughput improvement of EDCF-DM is significant. Under EDCF-DM, when the workload of the high priority traffic is low, the medium priority flows can exploit more chances of medium access than AEDCF since the CW_i can be decreased faster. As a result, the throughput of medium priority traffic has better throughput. On the other hand, as the workload of high priority traffic increases, the opportunity of fast decreasing CW_i reduces, and then the throughput gain of EDCF-DM becomes less.

The throughput of the low priority traffic is shown in Figure 3 (c). Surprisingly, AEDCF achieves the best throughput for the low priority traffic. Compared with EDCF, when the workload of the system is not too low (i.e., the number of nodes is greater than 5), the number of collisions is reduced under

AEDCF by adapting CW_i according to the network condition. The reason for less throughput under EDCF-DM is that: Under EDCF-DM, with the consideration of the traffic state of higher priority traffics, CW_i of the low priority traffic is changed more slowly than that under AEDCF. However, this indicates that EDCF-DM can provide better service differentiation than AEDCF. For example, when the number of nodes is 20, the throughput difference between the medium priority traffic and the low priority traffic under EDCF-DM is much larger than that under AEDCF.

In terms of system throughput, by simply adding the throughput of each traffic category together, it is easy to see that EDCF-DM has higher system throughput than EDCF and AEDCF.

C. Packet Delay Comparisons

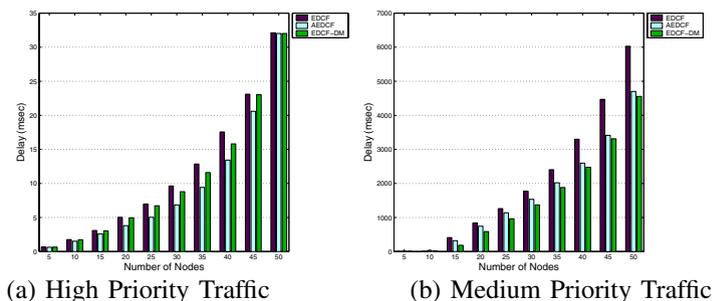


Fig. 4. The Packet Delay Comparison of EDCF, AEDCF and EDCF-DM

In this subsection, we compare the average packet delay under these three schemes. Since the low priority traffic is best effort traffic, each packet delay is dominated by the queuing delay. Thus, we do not show the packet delay of the low priority traffic. As shown in Figure 4 (a), under most workloads, AEDCF achieves the best packet delay for high priority traffic, and the delays under EDCF and EDCF-DM are almost the same. The reason why AEDCF outperforms EDCF has been explained in the previous subsection and is still valid here. For EDCF-DM, the longer packet delay is mainly due to the aggressively changing CW_i for medium priority traffic. Medium priority flows under EDCF-DM have more chances to transmit data than under AEDCF. As a result, the high priority packet will experience longer delay under EDCF-DM.

The packet delay of medium priority traffic is shown in Figure 4 (b). We can see that EDCF-DM has the best performance. Since both EDCF-DM and AEDCF control CW_i according to the network condition, as the workload increases, the packet delays of medium priority traffic under EDCF-DM and AEDCF are shorter than that under EDCF. According to the traffic state of high priority traffic, EDCF-DM adjusts CW_i of medium priority traffic more aggressively than AEDCF. Consequently, under EDCF-DM, a medium priority flow could grab the channel faster, which results in shorter packet delay.

D. Channel Utilization Comparisons

In addition to throughput and packet delay comparisons, we further evaluate channel utilization of these three schemes. As shown in Figure 5 (a), under most workloads, EDCF-DM and AEDCF have much better channel utilization than EDCF. This

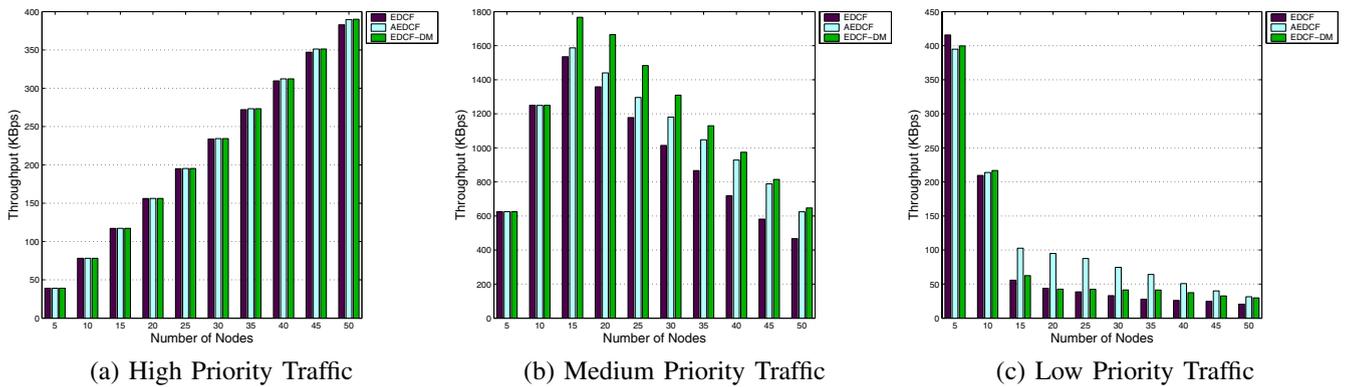


Fig. 3. The Throughput Comparison of EDCF, AEDCF and EDCF-DM

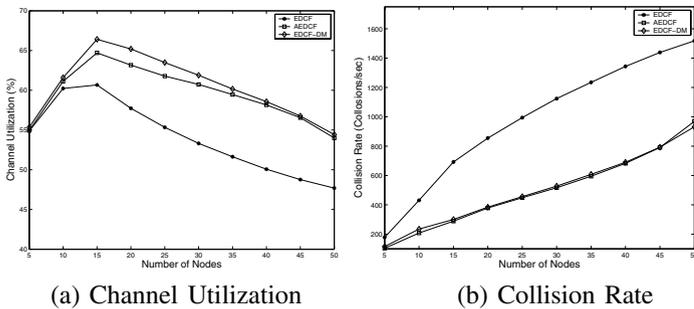


Fig. 5. The Channel Utilization Comparison of EDCF, AEDCF and EDCF-DM

is because that both EDCF-DM and AEDCF adjust CW_i upon a successful transmission according to the network condition. However, EDCF just blindly resets CW_i to $CW_{i,min}$ after a successful transmission. As a result, when the workload is not very low, the numbers of collisions under EDCF is much greater than that under EDCF-DM and AEDCF. To prove this claim, we show the collision rate of each scheme in Figure 5 (b). The collision rate is calculated as the average number of collisions per second. From the Figure, we can see that the number of collisions is significantly reduced under EDCF-DM and AEDCF.

From Figure 5 (a), we can also see that EDCF-DM achieves better channel utilization than AEDCF. As explained before, EDCF-DM adapts CW_i more aggressively than AEDCF when high priority traffics are low. As the workload of high priority traffics increases, which is correspondent to the number of nodes, the difference of channel utilization between EDCF-DM and AEDCF decreases. When higher priority traffic is not high, EDCF-DM can decrease CW_i more aggressively than AEDCF and reduce the number of wasted time slots due to backoffs. When higher priority traffics is high, EDCF-DM has little chances to fast adjust CW_i , and then acts similar to AEDCF. Since EDCF-DM adapts CW_i quite carefully, it controls the number of collisions as well as AEDCF, which is proven in Figure 5 (b).

V. CONCLUSION

In this paper, we focus on studying and improving EDCF through easily extensible techniques. We proposed a new dual-measurement based EDCF scheme, called EDCF-DM,

which has following features: First, it retains the service differentiation mechanism proposed in IEEE 802.11e; Second, it dynamically varies the size of the contention window by using an adaptive approach based on both the traffic state of each traffic category at each active node and the network condition of the system. As a result, EDCF-DM can reduce the number of collisions and maintain a good service differentiation; Third, EDCF-DM improves the system throughput and channel utilization by granting better access to the lower priority flows when the higher priority flows have no data to send. We evaluate the performance of EDCF-DM through extensive simulations. Simulation results show that, compared with EDCF and AEDCF, EDCF-DM not only improves the system throughput and channel utilization, but also maintains a good service differentiation. As future work, we will further balance the tradeoff between channel utilization and priority by adapting the contention window more intelligently.

REFERENCES

- [1] I. Ada and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," *IEEE INFOCOM'01*, 2001.
- [2] M. Barry, A. T. Campbell and A. Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," *IEEE INFOCOM'01*, 2001.
- [3] Jim Geier, "Wireless LANs, 2nd Edition," *Sams Publishing*, 2002.
- [4] IEEE Work Group, "Draft Supplement to Standard for Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," *IEEE 802.11e Draft 3.1*, May 2002.
- [5] VINT group, "UCB/LBNL/VINT Network Simulator – ns (Version 2)," <http://mash.cs.berkeley.edu/ns>.
- [6] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," *IEEE 802.11 standard*, 1999.
- [7] Y. Kwon, Y. Fang and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," *IEEE INFOCOM'03*, 2003.
- [8] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," *Proceedings of European Wireless*, Feb. 2002.
- [9] L. Romdhani, Q. Ni and T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks," *IEEE WCNC'03*, 2003.
- [10] J. Walrand and P. Varaiya, "High-Performance Communication Networks, Second Edition," *Morgan Kaufmann Publishers*, 2000.
- [11] X. Yang and N. H. Vaidya, "Priority Scheduling in Wireless Ad Hoc Networks," *ACM MobiHoc'02*, Jun. 2003.