# A Dynamic CBWFQ Scheme for Service Differentiation in WLANs

Vasilios A. Siris and George Stamatakis

Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH)
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
Tel.: +30 2810 391726, fax: +30 2810 391601, email: {vsiris,gstam}@ics.forth.gr

*Abstract*— **In this paper we address the issue of service differentiation in wireless networks based on the IEEE 802.11 standard. Wireless local area networks (WLANs) of this type are becoming increasingly prevalent, counting millions of deployed networks in homes, offices and hot spots. However, the medium access control (MAC) protocol and the unpredictability of the wireless channel prohibit the direct employment of classical wireline service differentiation algorithms. The main contribution of this paper is a network layer mechanism, capable of differentiating service, while improving fairness and aggregate throughput in the presence of both uplink (TCP) and downlink traffic. We exploit a class based weight fair queuing (CBWFQ) scheme, properly supplemented by two algorithms that perform periodical weight adjustments. The two algorithms deal effectively with important WLANs problems, such as, unfairness due to location dependent channel errors, uplink-downlink unfairness and decreased throughput for all nodes, when a station transmits at a small rate. The proposed solution has been implemented in a testbed and the conducted experiments confirmed its resultfulness in real life scenarios.**

**Keywords: WLAN, service differentiation, fairness, performance**

## I. INTRODUCTION

Wireless local area networks (WLANs) based on IEEE 802.11 technology are experiencing a widespread deployment in both indoor and outdoor environments. The prominent reason behind the explosive growth of this technology is the need for tetherless connectivity to Internet resources. As a consequence, WLANs are expected to support all the communication-intensive applications encountered in wired networks over the Internet. It is conceivable that a service differentiation mechanism, properly adapted to wireless network particularities, is a major prerequisite to achieve this goal; for both existing and future WLANs.

Provision of service differentiation has been studied at various levels in the protocol hierarchy. However, none of the proposed solutions provides a service differentiation mechanism that can be implemented solely at an *Access Router* and address, in a feasible and effective way, fairness and performance problems of WLANs. Mechanisms that require changes at the MAC layer of 802.11 cannot be applied to the large number of networks already installed. Furthermore, fairness and performance issues are often dealt with separately resulting in partial solutions. Such issues though, e.g. decreased throughput due to a node's small transmission rate and Uplink-Downlink unfairness, are major afflictions for WLANs and should be addressed in conjunction.

The purpose of this work is the design of a dynamic CBWFQ queuing mechanism that improves fairness and aggregate throughput, while supporting weight-based service differentiation in terms of throughput. The motive for this dynamic scheme selection is that a typical CBWFQ mechanism would yield unpredictable results in the presence of location-dependent bursty channel errors. This key feature of the wireless link brings up the need for a fair, resource redistribution algorithm. However, such resource sharing algorithms that do not consider the channel state, might offer even more resources to nodes that cannot transmit due to bad link conditions. This characteristic could have a negative effect on the overall network performance. By allocating resources for classes that can not exploit them, the system would offer more resources just to increase losses. To address this implication we propose an algorithm that considers the transmission rate used by a node, as an indication of the link quality. A prototype of the proposed scheme has been implemented on a wireless testbed and the experimental results are provided for evaluation.

The rest of the paper is organized as follows. In Section II, we describe our service differentiation mechanism. In Section III, we present the wireless testbed setup. In Section IV, we describe the evaluation scenarios and analyze the experimental results. In Section V, we review related work and finally, in Section VI, we present our conclusions and identify related ongoing and future work.

## II. DYNAMIC CBWFQ

The dynamic CBWFQ mechanism is composed of three parts. The service differentiation mechanism, the fairness algorithm and the performance improvement algorithm. The first one distinguishes between uplink and downlink traffic, and the latter two perform weight adjustments periodically.

Discriminating between uplink and downlink traffic is of critical importance since the base station and the mobile hosts have equal access to the wireless medium. Without some control over the uplink traffic no service differentiation mechanism can be applied successfully to the WLAN.

### A. Service Differentiation Mechanism

The service differentiation mechanism provides per node throughput differentiation based on a CBWFQ scheme. A class, in this link sharing structure, corresponds to an aggregation of traffic uniquely identified by information included in the packet's header and the packet's size. Each wireless node is related with two classes, the ACK and the DATA class. The ACK class matches the TCP acknowledgment packets destined to a host, while the DATA class matches the data packets (TCP or UDP) destined to it. Every class is assigned a separate queue for its packets and the bandwidth is shared among them according to their weights. In case there is unutilized bandwidth it will be shared among the nodes that generate traffic in proportion to their weights; thus a class could receive more resources than the minimum rate its weight guarantees, if demand for resources is low.

Discrimination between acknowledgment and data packets destined to a host provides the means to control its uplink traffic. By controlling the transmission rate of acknowledgment packets we can affect a node's aggregate uplink transmission rate. Experiments suggest that the TCP uplink rate is approximately proportional to the rate of ACKs.

$$T_{uplink,j} \simeq k \cdot T_{ACK,j} \tag{1}$$

where $T_{uplink,j}$ is the uplink rate achievable by node $j$ and $T_{ACK,j}$ is the shaped rate of node's $j$ ACK class. Fig. 1 presents the aggregate transmission rate for three TCP flows as a function of the acknowledgments aggregate transmission rate. Each point of the plot is a mean value calculated over a period of 60 seconds. The flows originated at a mobile host and were destined to fixed host.

Equation (1) provides an upper bound for uplink traffic. The rate that a node actually achieves depends on many other parameters as well, e.g. link quality and congestion. The fact that limitations can be imposed on the uplink TCP traffic generated by mobile hosts permits the effective deployment of our service differentiation policy. Although there is no corresponding method for controlling uplink UDP traffic, our method can be applied when some rate control over UDP is used, as in video streaming and the DCCP protocol.

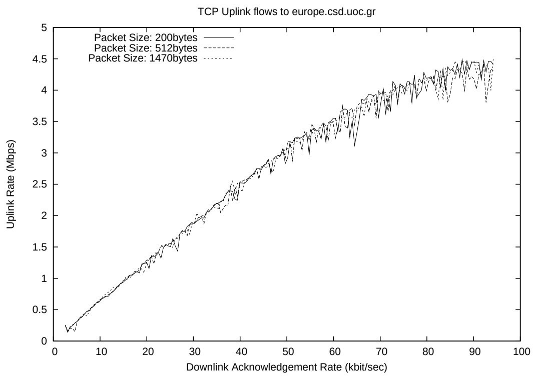

Fig. 1. Uplink transmission rate as a function of acknowledgment transmission rate.

### B. Fairness Algorithm

The effectiveness of the CBWFQ scheme is limited due to the existence of location-dependent bursty channel errors. It is conceivable though to achieve long term fairness by giving more service to a class that previously experienced an erroneous channel or to a class previously absent. Accounting absence as loss of service improves performance in the case of bursty data traffic.

The proposed fairness algorithm performs this resource management on top of the CBWFQ mechanism based entirely on weight manipulation. The algorithm takes feedback periodically by monitoring the achieved throughput of the classes. The duration of the period affects how fast the algorithm responds to changes in the wireless network. The amount of service lost or gained by class $i$ up to period $n$ is:

$$D_i[n] = D_i[n-1] + (achieved\_r_i[n] - target\_r_i[n]) \tag{2}$$

where $achieved\_r_i[n]$ is the rate achieved by class $i$ during period $n$, $D_i[0]$ is zero and $target\_r_i[n]$ is the rate class $i$ was expected to achieve in the same period:

$$target\_r_i[n] = \frac{w_i[n]}{\sum_j w_j[n]} \cdot C_{cck11} \tag{3}$$

where $w_i[n]$ is the weight of class $i$ and $C_{cck11}$ is the 802.11 protocol capacity when complementary code keying at 11Mbps is used. Because of the overheads introduced by the physical and MAC layers the value of $C_{cck11}$ can't be larger than 6Mbps. The exact value of this parameter will not affect the functionality of the algorithm though it will affect the speed of convergence. Equation (2) designates that $D_i[n]$ will be negative if node $i$ has lost service and it will be positive if node $i$ has received excess service.

The weight of class $i$ will be updated at the end of each period according to:

$$w_i[n+1] = w_{i,initial} + f_i[n], \quad \forall i \tag{4}$$

where $f_i[n]$ is a function of $D_i[n]$. Function $f_i[n]$ must be bounded so as to avoid an excess increase of a class's weight which will cause starvation of other traffic classes. It should also result in a smooth decrease or increase of a class's weight. To achieve the above we propose the following function that has been used successfully in our experiments:

$$f_i[n] = \begin{cases} \ln\left(\frac{|D_i[n]|}{C_{cck11}} + 1\right), & \text{if} \quad \text{-}w_{i,init} \cdot S_l \leq D_i[n] \leq 0 \\ -\ln\left(\frac{|D_i[n]|}{C_{cck11}} + 1\right), & \text{if} \quad 0 < D_i[n] \leq w_{i,init} \cdot S_g \end{cases} \tag{5}$$

where $S_l$ and $S_g$ are, respectively, the maximum aggregate loss and gain we will account for in our WLAN; $w_{i,init}$ is the initial weight of class $i$. Values of $D_i[n]$ larger than $w_{i,nit} \cdot Sg$ or smaller than $-w_{i,init} \cdot S_l$ won't be accounted for. The value of $S_l$ is determined by considering the available resources so that our algorithm can indeed compensate the amount of losses it accounts for. The value of $S_g$ should be such that a class that has received excess service in the past won't starve or suffer weight reduction for a prolonged period. Furthermore, the maximum amount of loss or gain of a class in any period is analogous to its initial weight and so is the compensation or service reduction it will receive.

### C. *Performance Improvement Algorithm*

The main idea in improving performance of WLANs is to assign resources to each wireless node according to its capability to exploit them. A node with transmission and receive failures not only wastes its resources but also prohibits other nodes from using their own (head of line blocking, etc.).

In order to estimate the resources that should be assigned to each host it is necessary to obtain information concerning the channel state between the base station and the end hosts. For 802.11b networks such information is available through the data transmission rate. The 802.11b physical layer defines four possible transmission rates, 1, 2, 5.5 and 11 Mbps, each one corresponding to a different modulation scheme. The type of modulation specifies the bit error rate (BER) as a function of signal to noise (SNR) ratio. Each node will change its modulation scheme, degrading its transmission rate, when channel errors occur, in order to increase the probability of correct future transmissions. The 802.11 standard specification doesn't define the algorithm for dynamic rate switching allowing vendors to design their own mechanisms so as to *optimize* their products' behavior.

A node transmitting with a *lower rate* than 11 Mbps (5.5, 2, 1) will acquire the channel longer than a node transmitting at 11 Mbps for transmission of the same amount of data. This fact causes a significant degradation in aggregate throughput and is called the *Performance Anomaly* phenomenon of 802.11 [1].

To address this, so called, Performance Anomaly problem of WLANs we propose the weight reduction for nodes that use a transmission rate lower than 11Mbps. This reduction will be done by multiplying their classes' weights with a weight coefficient (link quality coefficient). The value of this coefficient will be the ratio of the maximum transmission rate achievable by the modulation used, per the maximum transmission rate of CCK-11 modulation (including 802.11 protocol overhead). In Table (I) we present the link quality coefficient values. Information on the transmission rate is provided by the network card driver periodically.

To take into account the channel state in terms of the transmission rate, the weight of class $i$ at the end of each period is updated according to:

$$w_i[n+1] = (w_{i,initial} + f_i[n]) \cdot l_i[n] \tag{6}$$

TABLE I

$l_i$ FOR THE EACH MODULATION SCHEME.

| Modulation | Transmission Rate (Mbps) | Link Quality Coefficient |
|---|---|---|
| DBPSK | 1 | 1/6 |
| DQPSK | 2 | 2/6 |
| CCK-5.5 Mbps | 4 | 4/6 |
| CCK-11 Mbps | 6 | 1 |

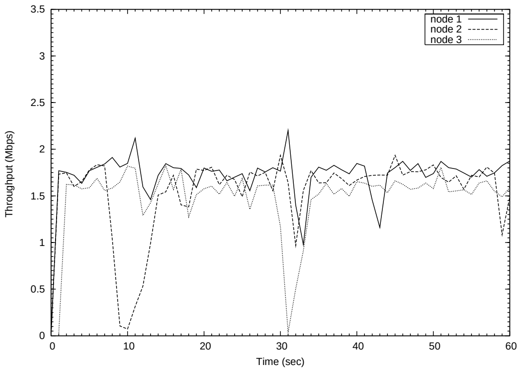

Fig. 2. Unfairness due to channel errors.

where $l_i$ is the link quality coefficient, $f_i[n]$ is computed from (5) and $w_{i,init}$ is the initial weight of class $i$. Hence, if a station has low transmission rate, it will receive a low weight, compared to a station that transmits at 11Mbps.

## III. IMPLEMENTATION

Our algorithm is implemented on a Linux Box equipped with a Prism 2.5 wireless card that is driven by the HostAP driver. The CBWFQ mechanism was based on hierarchical token bucket queuing discipline (HTB). A set of parsers gather the throughput and transmission rate statistics from the Linux `tc` tool and the HostAP driver respectively.

## IV. EXPERIMENTAL EVALUATION

For the experimental evaluation of the dynamic CBWFQ scheme a typical network topology was used that included a wired host directly connected to the access router and two or three mobile hosts. The testbed was placed in a typical office environment, in which signal fades occurred as people moved. The aggregate output queue of all HTB classes was equal to the Linux default output queue. Parameters $S_l$ and $S_g$ were set to 25Mbit for all experiments.

*Fair bandwidth distribution in case of bursty errors.* The target of this experiment is to demonstrate the effectiveness of the fairness algorithm in redistributing bandwidth fairly. For this experiment three wireless hosts were used. Each one was receiving a persistent TCP flow. The available bandwidth was initially shared equally among the three DATA classes. In Fig. 2 we present the throughput achieved by each flow, over time, without use of the dynamic CBWFQ mechanism. Nodes 2 and 3 suffered severe losses at the 10th and 30th second respectively, which, combined with other less severe losses, resulted in an unfair bandwidth distribution. Fig. 3 on the other hand presents the throughput achieved by each flow when the dynamic CBWFQ mechanism was activated. In this case, although nodes 1 and 3 lost service at the 26th and 44th second, they were compensated for their lost service as indicated by the throughput overshoot that followed each loss. Furthermore, it can be seen that when a node experienced reduced channel quality, other nodes were allowed to use more resources.

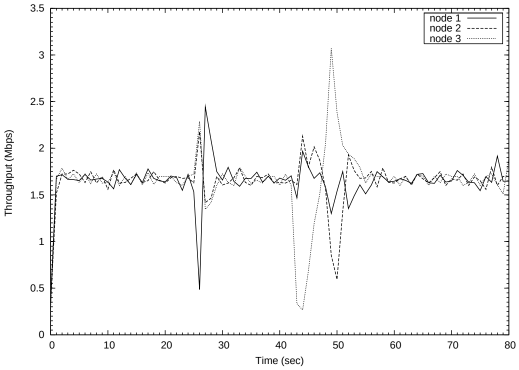

Fig. 3. Fairness in an error channel.

*Uplink-Downlink Unfairness.* This experiment shows that our mechanism resolves the uplink-downlink unfairness problem of WLAN's. As stated in [2] the main reasons causing such unfairness is buffer availability at the base station and cumulative acknowledgments. Our mechanism separates data packets from acknowledgment ones and schedules them through different queues. This characteristic makes the Access Router immune to the problem of uplink-downlink unfairness. For this experiment we used three nodes; each one sent 3 TCP data flows and received 3 TCP data flows. Additionally the total buffer of the wireless interface was reduced to 60 packets from the default value of 100 packets. This experimental setup caused the output buffer of the Access Router to overflow and exhibit the uplink-downlink unfairness phenomenon. Fig. 4 is a snapshot taken from the Windows performance monitor of a node and presents this phenomenon as well the uplink-downlink throughput when our mechanism was activated at the 70th second of the experiment. DATA and ACK classes were assigned, initially, equal shares of bandwidth. The roughness of the aggregate downlink throughput seen in the figures occurs because of the large number of flows serviced by the access point in a heavily congested link.

*Response time measurements for web traffic.* This experiment points out the performance improvement that results from compensating service loss due to absence. Since this aspect focuses on bursty data traffic, Web browsing is used as a case study. An Apache web server was installed on the wired host and three objects of sizes 50, 152, and 500 KBytes were available for download. Two wireless stations were used for this experiment; the first one made an ftp transfer of a large file while the second one performed HTTP requests using the `wget()` utility. An exponentially distributed think time, with a mean of 20 seconds, was assumed. The wireless host that performed the HTTP requests was a Linux Box with the same setup as the Access Router. Each DATA class was initially assigned half of the available bandwidth. The experiment was comprised of 20 HTTP requests for each object and the mean response times can be seen in Fig. 5. The performance improvement is significant for small objects while it reduces for larger ones. This is due to the limitation imposed to the amount of service that a class can be compensated for, which is controlled by the $S_l$ parameter.

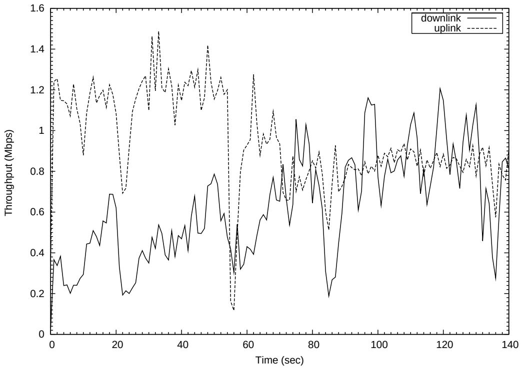

Fig. 4. Resolving uplink-downlink unfairness.

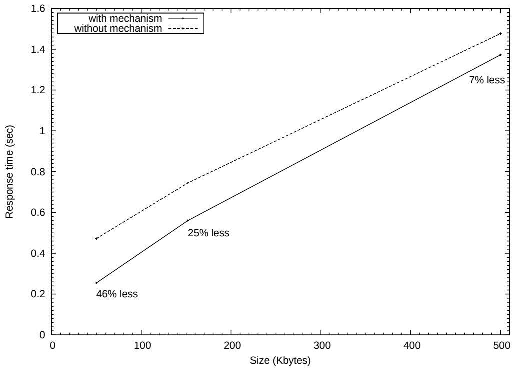

Fig. 5. Improving response time for Web traffic.

*Performance anomaly of 802.11.* This final experiment presents the behavior of our mechanism in a performance anomaly scenario. We used three wireless hosts, each was generating an uplink TCP flow destined to the wired host. The ACK classes were assigned initially equal shares of the available bandwidth. One of the wireless hosts was changing its transmission rate over time by using the wireless card's configuration program. In Fig. 6 node's 1 transmission rate oscillated between 11Mbps and the other three possible rates, and forced an analogous performance reduction for all nodes. Fig. 7 shows a similar scenario with the dynamic CBWFQ mechanism activated. It can be seen that, by reducing the weight of the node transmitting with a lower rate than 11Mbps, we prevented the performance degradation of the other nodes.

## V. RELATED WORK

Service differentiation in wireless networks has been studied by several researchers. In [3] the authors combined the class-based queuing (CBQ) mechanism with a channel state dependent packet scheduler and a compensation algorithm in a manner that is conceptually similar to ours. However, they don't address the problem of uplink traffic control and there is no accounting for losses due to absence. Furthermore, they propose the design of a special purpose scheduler, whereas our algorithms are independent modules communicating with the scheduler only through the class weight parameter. Their algorithm prohibits the transmission of a class that has received excess service in the presence of a class that has suffered losses. Hence there is no provision for starvation avoidance.

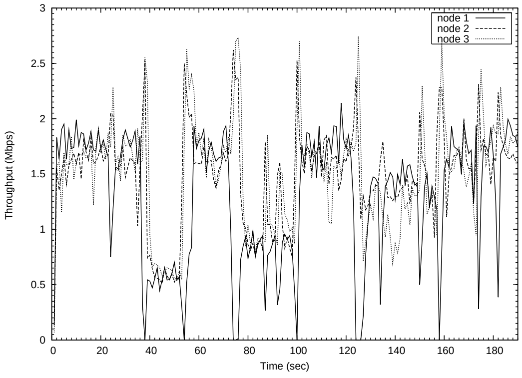

Fig. 6. Performance anomaly

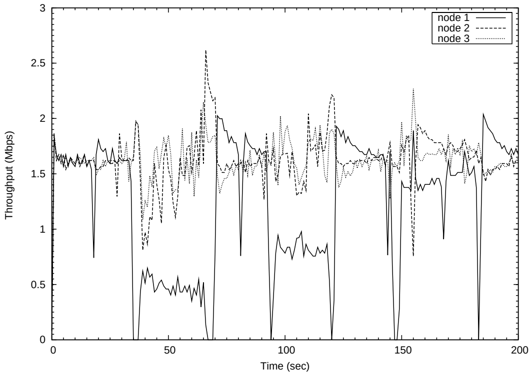

Fig. 7. Resolving performance anomaly

Finally the link state estimation algorithm proposed, is based on RTS/CTS messages and alteration of the maximum retransmission attempts.

With regard to fairness, of particular interest are the works in [4], [5], [6] and [7] presenting fair scheduling algorithms. In [4], Zhu *et al.*, introduced the notion of accounting absence as loss of service but they didn't correlate it with weight adjustment as a mean to improve fairness and performance. Furthermore, deployment of all the above algorithms in 802.11 networks demands the construction of a specialized scheduler, significant changes at the MAC layer and modifications of the end systems.

The upcoming IEEE *802.11e* protocol provides MAC layer functions for per node service differentiation. However, unlike our mechanism, only static service differentiation is supported and there is neither compensation for losses nor anticipation for throughput degradation due to the multirate physical layer. Moreover, our dynamic CBWFQ mechanism, implemented at the network layer, is compatible with all existing 802.11b/g networks, whereas 802.11e requires new equipment.

The wireless link-state estimation problem has also been addressed by many researchers. In [8], Aida *et al.*, make use of the mean SNR in order to characterize the state of the wireless links. The major drawback of this method is the lack of knowledge of the SNR value at the region of the mobile host, which must be transmitted explicitly to the access point. Another line of research [9] suggests to employ MAC level acknowledgment failures as a criterion for link quality and change the retransmission attempts accordingly. These algorithms, however, require changes at the MAC layer and modifications of the end systems, whereas our approach, utilizes the transmission rate as an indication of a link's quality. This information is readily available at the Access Router.

Recently, in [2], Pilosof *et al.*, proposed the dynamic adjustment of the TCP window in order to control uplink traffic. This method, however, could not coexist with security protocols, e.g IPsec, that encrypt packets content. It also requires exact knowledge of the active TCP flows number in order to share bandwidth fairly, which is difficult to obtain in the presence of idle TCP flows. In contrast to this method, our approach for controlling uplink traffic requires only the number of hosts that are associated with the Access Router.

Finally, in [10] the authors redefine fairness in terms of time shares. By granting temporal fair share of the channel to each traffic source they improve fairness and performance even in multi-rate capable physical layers. However, their algorithm does not provide the means to differentiate service, contrariwise to our mechanism that is based on a CBWFQ service differentiation mechanism.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we addressed the problem of supporting service differentiation in commercial WLANs based on 802.11 protocol. In such networks the direct employment of wireline service differentiation algorithms is not effective due to the peculiarities of the wireless channel. However, utilization of existing mechanisms, properly supported with algorithms that deal with fairness and performance issues proved to be an efficient and practical solution in real life scenarios.

Nevertheless, there are still open issues that we are currently pursuing. Construction of an admission control module and experimentation with voice and video traffic on top of 802.11g networks are parts of an ongoing work. Furthermore, we plan to implement the dynamic CBWFQ mechanism on a control station that serves more than one access points.

## REFERENCES

[1] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," *IEEE INFOCOM 2003*, 2003.
[2] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding tcp fairness over wireless lan," *IEEE INFOCOM*, 2003.
[3] C. Fragouli, M. Srivastava, and V. Sivaraman, "Controlled multimedia wireless link sharing via enhanced class-based queueing with channel state dependent packet scheduling," *IEEE INFOCOM'98*, March 1998.
[4] H.Zhu and G. Cao, "On improving service differentiation under bursty data traffic in wireless networks," *IEEE INFOCOM*, 2004.
[5] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on networking*, vol. 7, August 1999.
[6] T. Ng, I.Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location dependent errors," *IEEE INFOCOM'98*.
[7] M. Bottiglienco, C. Casetti, C. Chiasserini, and M. Meo, "Smart traffic scheduling in 802.11 wlans with access point," *VTC IEEE*, 2003.
[8] H. Aida, Y. Tobe, M. Saito, and H. Tokuda, "A software approach to channel-state dependent scheduling for wireless lans," *WOWMOM*, 2001.
[9] P. Bhagwat, P. Bhattacharya, A. Krishma, and S. Tripathi, "Enhancing throughput over wireless lans using channel state dependent packet scheduling," *IEEE INFOCOM'97*, April 1996.
[10] Y. Yuan, D. Gu, W. Arbaugh, and J. Zhang, "Achieving packet-level quality of service through scheduling in multirate wlans," *VTC IEEE*, 2004.