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 mechanism capable of differentiating service, in the presence of both uplink (TCP) and downlink traffic, that demands no alternations in the existing protocol stack. We exploit a class based weight fair queuing (CBWFQ) scheme, properly supplemented by two algorithms that perform periodical weight adjustments in order to improve fairness and performance respectively. The two algorithms deal effectively with major WLAN's 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 was implemented in a testbed and the conducted experiments confirmed its resultfulness in real life scenarios.

Keywords: WLAN, service differentiation, fairness, performance

1 Introduction

Wireless local area networks (WLANs) based on IEEE 802.11b/g 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. At the MAC layer, 802.11e, the new IEEE draft, defines the MAC procedures to support LAN applications with QoS requirements. A large number of scheduling algorithms have also been

proposed for wireless networks that consider channel state, losses etc. At the network level, protocols such as IntServ and DiffServ, properly modified to fit the wireless channel needs, provide the means to support service differentiation in WLANs.

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 802.11b/g should be considered impractical due 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 class based weight fair queuing mechanism that differentiates service based on traffic monitoring so as to improve fairness and aggregate 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. Such algorithms on the other hand tend to favor classes that have lost service as they attempt to restore fairness. This characteristic can be proved devastating for the overall performance of the network. By allocating resources for classes that can not exploit them the system offers more resources just to increase losses. To address this implication we propose an algorithm that considers the modulation scheme used by a node, as an indication of the link quality. A prototype of the proposed scheme is implemented on a wireless testbed and the experimental results are provided for evaluation.

The rest of the paper is organized as follows. In Section 2, we describe our service differentiation mechanism. In Section 3, we present the wireless testbed setup. In Section 4, we describe the evaluation scenarios and analyze the experimental results. In Section 5, we review related work and finally, in Section 6, we present our conclusions and identify related ongoing and future work.

2 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 services them according to their weights, 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 share the wireless medium and have equal access to it. It is obvious that without some control over the uplink traffic no service differentiation mechanism can be applied successfully to the WLAN.

2.1 Service Differentiation Mechanism

The service differentiation mechanism is based on CBWFQ. A class, in this link-sharing structure, corresponds to an aggregation of traffic. Each node associated to the Access Router is related with two classes of packets, the ACK and the DATA classes. 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. The service provided to each host, on a congested link, is differentiated according to the weights of its ACK and DATA classes. 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. The function we use for rate controlling ACK classes is based on experimental results and is linear:

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

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. The experiments were conducted on our testbed and a host located in a remote LAN. The experiments included scenarios with one or more TCP flows and various packet sizes. 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 the remote LAN.

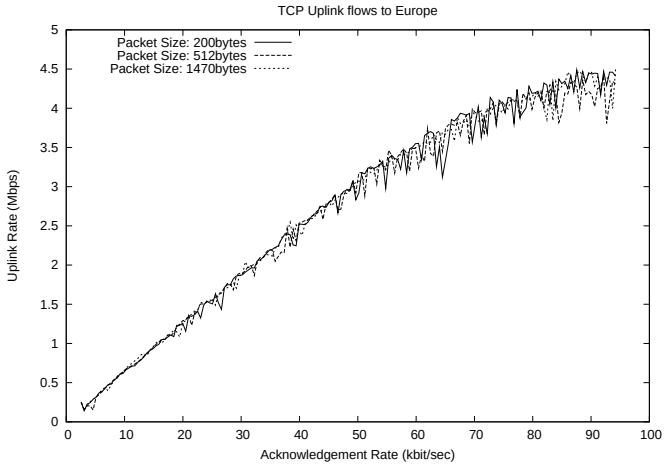

Figure 1: Uplink transmission rate as a function of acknowledgment transmission rate.

Equation (1) provides the 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 limi-

tations 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 DCCP protocol.

2.2 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 with a previously error session or to a class previously absent. Accounting absence as loss of service improves performance in the presence of bursty data traffic. Compensation can be achieved by degrading the services of other nodes.

The fairness algorithm proposed performs this resource management on top of the CBWFQ mechanism based entirely on weight manipulation. The algorithm takes feedback periodically by monitoring the actual performance of each node's classes. The duration of the period affects how fast the algorithm responds to changes in the wireless network, and should be as small as the accuracy of the monitoring mechanism permits. The amount of service lost or gained by class i up to period n is:

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

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:

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

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 starting from time interval $n = 0$.

In case class i has not sent any packets (absence), $achieved\_r_i = 0$ and the whole amount of $target\_r_i$ will be accounted as lost service. In a real world scenario it is expected that all nodes will have prolonged periods of absence. It is obvious that this can result in severe losses (very small D_i) for every class i! However, the limited resources of a WLAN allow for small compensations only, which will be inadequate to change significantly the value of D_i for all i. In order to address this problem we require that:

$$D_i[n] \geq -(w_{i,initial} \cdot S_l), \qquad \text{for all } i \tag{4}$$

where S_l expresses the maximum aggregate loss we will account for in our WLAN and $w_{i,initial}$ is the initially assigned weight to class i. Each class, according to its weight will be compensated for losses up to the value given by equation (4). The S_l value is determined by considering the available bandwidth, so that our algorithm can compensate the amount of losses it accounts for.

Similarly, an upper bound is necessary for $D_i[n]$. A class that utilizes excess bandwidth for an extended period of time will face a similar problem. A scenario that corresponds to this case is when a single class generates traffic, e.g. ftp, while all other classes remain absent. Since the class will be exploiting all the available bandwidth, the value of $D_i[n]$ will grow large (leading state). If the other classes begin to serve traffic at some point, the class in the leading state will suffer severe resource reduction for a prolonged period of time. This is unacceptable since it violates the initially agreed bandwidth distribution scheme for a large period of time.

The upper bound for $D_i[n]$ will be:

$$D_i[n] \leq w_{i,initial} \cdot S_g, \qquad \text{for all } i \tag{5}$$

where S_g expresses the maximum aggregate gain we will account for in our WLAN. Each class, according to its weight will be charged for the excess service it received up to the value given by inequality (5). A consequence of (5) is that small weight classes will not be punished as much as big weight classes, since that would cause their starvation.

The weight of class i will be updated at the end of each time interval according to:

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

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. A function that possess the above properties and has been used successfully in our experiments is:

$$f_i[n] = \begin{cases} \ln\left(\frac{|D_i[n]|}{C_{cck11}} + 1\right), & \text{if} \quad \text{-}w_{i,init} \cdot S_l \leq D_i[n] \leq 0 \\ \ln\left(\frac{|D_i[n]|}{C_{cck11}} + 1\right), & \text{if} \quad 0 < D_i[n] \leq w_{i,init} \cdot S_g \end{cases} \tag{7}$$

The form of this function indicates a non-linear compensation method.

2.3 Performance Improvement Algorithm

The main concept 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. MAC layer retransmissions, contention window increase, transmission rate reduction and head of line blocking affect the overall network performance.

In order to estimate the resources that should be assigned to each host it is necessary to obtain information concerning 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 (see Fig. 3). 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 specify the algorithm for dynamic rate switching allowing vendors to design their own mechanisms so as to *optimize* their products' behavior. Hence, although the exact algorithm used by each wireless card is not known, we can reach useful conclusions by exploiting the transmission rate information.

Fig. (2), which was taken from [7] shows the simulated performance, in terms of throughput, for each modulation scheme available in IEEE 802.11b versus the SNR (assuming an optimum link state prediction algorithm). It is conceivable that a node using a modulation other than CCK-11 will possess the wireless medium longer than a node using CCK-11 for the transmission of the same amount of data. This fact causes a significant degradation in aggregate throughput of WLANs and is called the *Performance Anomaly* phenomenon of 802.11 [5].

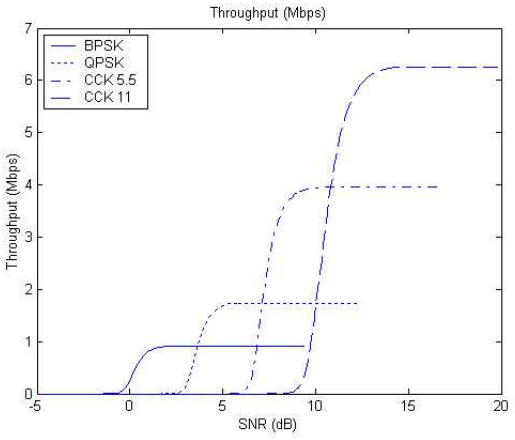

Figure 2: IEEE 802.11b throughput vs. SNR

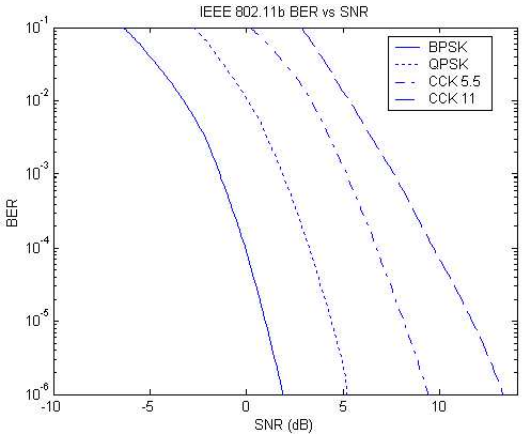

Figure 3: IEEE 802.11b BER vs. SNR

To address this, so called, performance anomaly problem of WLANs in conjunction with the performance improvement one we propose the weight reduction of all nodes that use a lower transmission rate 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 (1) we present the link quality coefficient values assuming that the maximum rates are approximately those of Fig. 2.

With this method we achieve to reduce the weight of those classes that have bad link quality

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

Table 1: l_i for the each modulation scheme.

according to the *encumberment* they cause to the WLAN. Furthermore the estimation of channel state is done by the network card driver in a way that adjusts to channel state variations efficiently without requiring an alternation of the MAC layer of 802.11. The equation for updating of unnormalized weight of class i at the end of each period (equation 6) becomes:

$$w_i[n] = (w_{i,initial} + f_i[n]) \cdot l_i[n] \tag{8}$$

where l_i is the link quality coefficient. The pseudo-code for the estimation of the value of l_i for the next period is presented in the next section.

3 Implementation

The dynamic class based scheme proposed is implemented on an experimental testbed. For the evaluation of the WLAN's performance a typical network topology is used that appears if Fig. 4. The mobile hosts used for the experiments run the Windows XP operating system without any modifications. For the Access Router machine a Linux box is used running the Redhat 9.0 operating system with an updated kernel version (2.4.27). The Linux box is equipped with a wireless card, based on Prism 2.5 chipset, which is driven by the HostAP driver (version 0.2.5) in Master mode (Access Point). Packets are forwarded by the driver to upper layers for the proper routing decisions to be taken. The Linux kernel has been configured so as to act as a software router forwarding packets from the wired to the wireless network and vice versa (ip_forwarding and ip_masquerading).

The service differentiation mechanism is implemented using the hierarchical token bucket discipline [13], [12] and the Linux `tc` tool. A sample HTB configuration for two wireless nodes appears in Fig. 5. The size of the queue attached to each class is the ratio of the Linux default output queue size (100 packets) per the number of classes. A proper set of functions is provided to control the behavior of each HTB class according to its weight. Furthermore if there is excess bandwidth it will be shared only among the competing DATA classes, in proportion to their weights. The HTB mechanism, handling excess bandwidth sharing, is not capable of handling small packets accurately, and therefore can't be used for ACK classes.

The fairness and performance improvement algorithms are based on a set of parsers which are

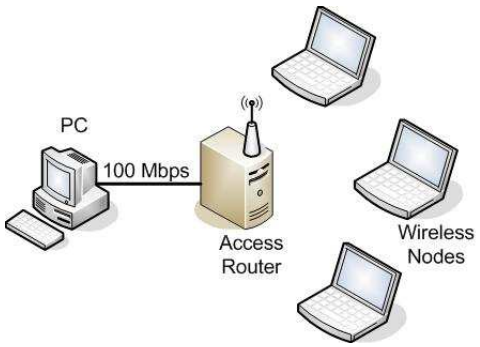

Figure 4: Network Topology

executed periodically. They gather all the statistics from system files and the `tc` tool output that are necessary to perform the calculations presented in Section 2. In particular transmission rate information is taken from HostAP output files located at the `proc` file system. A separate file is provided for each host associated with the base station. The driver exports detailed information for the last packet serviced and aggregate statistics for previous packets. Utilization of this information permits a more accurate estimation of a node's link quality coefficient for the next period.

4 Experimental Evaluation

In this section we evaluate the performance of the proposed dynamic CBWFQ scheme and demonstrate its effectiveness in resolving common WLAN's issues. The testbed for carrying out the experiments was placed in a typical office environment. All channel distortions, fades and interferences occurred in a random way as people moved inside the office. The channel state of each node was very good when there were no intervening obstacles. The S_l and S_g system parameters were set to 25Mbit for all five experiments of the evaluation.

Uplink Service Differentiation. The aim of this experiment is to exhibit the capability of our mechanism to differentiate uplink traffic. The scenario included two mobile hosts. Each host

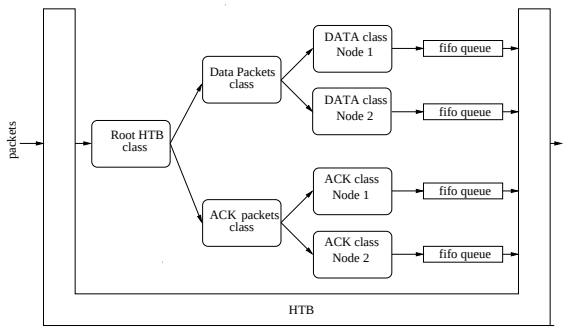

Figure 5: HTB class diagram

Algorithm 1 Deriving l_i value based on transmission rate

if $(var\_rx1 \geqslant 2 \text{ and } rate_{lp} = 1Mbps)$ **then**
    $l_i = 1/6$;
else if $(var\_rx2 \geqslant 2 \text{ and } rate_{lp} = 2Mbps)$ **then**
    $l_i = 2/6$;
else if $(var\_rx5.5 \geqslant 2 \text{ and } rate_{lp} = 5.5Mbps)$ **then**
    $l_i = 4/6$;
else if $(var\_rx11 \geqslant 2 \text{ and } rate_{lp} = 11Mbps)$ **then**
    $l_i = 1$;
else
    $l_i = l_i$;
end if

generated an uplink TCP flow destined to the wired host. Traffic generation was based on `Iperf` client-server pairs. The two ACK classes were initially assigned equal bandwidth shares. Every 60 seconds the weights of the two classes were changed. The throughput of each uplink flow was measured at the server side (receiver) of Iperf and the results can be seen in Fig. 6. There we present the ratio of throughputs for the uplink flows as a function of the ratio of weights of the ACK classes. The linearity of the function verifies the ability of our mechanism to differentiate uplink TCP traffic.

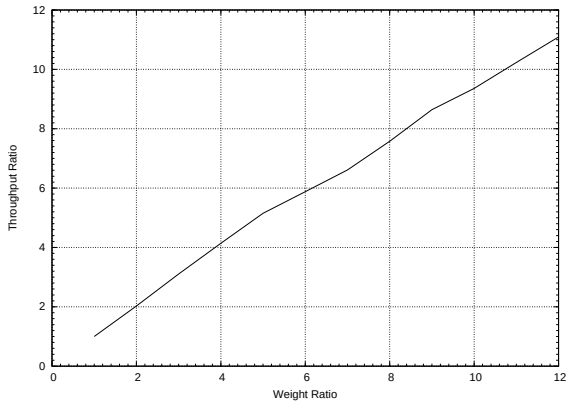

Figure 6: Uplink service differentiation.

Fair bandwidth distribution. 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 of them was receiving a persistent TCP flow. The available bandwidth was initially allocated equitably among the three DATA classes. In Fig. 7 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 important losses, resulted in an unfair bandwidth distribution. Fig. 8 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 follows. Furthermore it can be seen that during the declination of a node's performance other nodes were permitted to utilize the bandwidth assigned to it. In Fig. 9 we present the amount of lost service for each node in Mbits for the same experiment. All nodes started in a lagging state since they were associated to the access point before any traffi c was generated, hence they all had lost service due to absence. When TCP flows started generating traffi c the magnitude of lost service for each node decreased. One would expect, though, that losses would remain unchanged since all the available bandwidth was utilized by the three flows and nothing was left for loss compensation. The decrease observed is a result of our conservative estimation of the available bandwidth. We assumed it to be approximately 5Mbps (considering protocol overhead) when the 11Mbps transmission rate was used. At the 26th and 44th seconds we can see the loss of service for nodes 2 and 3 respectively. The lost service was compensated by the throughput overshoots so that its magnitude is equal for all nodes at the end of the experiment.

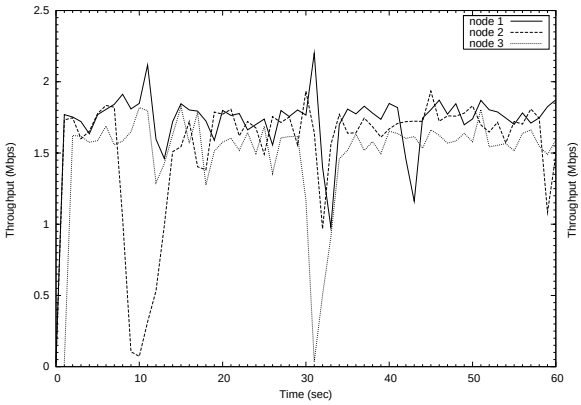

Figure 7: Unfairness due to channel errors.

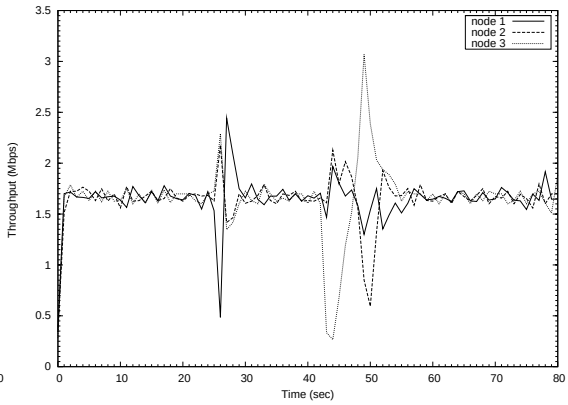

Figure 8: Fairness in an error channel.

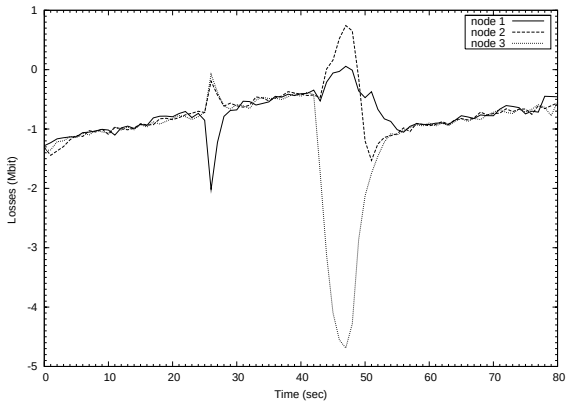

Figure 9: Amount of service lost by each node.

Uplink-Downlink Unfairness. This experiment shows that our mechanism resolves the uplink-downlink unfairness problem of WLAN's. As stated in [11] the main reasons causing this unfairness issue are buffer availability at the base station and the concept of 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 three TCP data flows and received three TCP data flows. Since for each TCP data flow there is a corresponding acknowledgment flow on the opposite direction each node served a total of 12 flows. The Access Router on the other hand served a total of 26 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 up link-downlink unfairness phenomenon. Fig. 10 and 11 show snapshots taken from Windows performance monitor that present this phenomenon as well as its fade out when our mechanism was activated at the 70th second of the experiment. DATA and ACK classes were assigned, initially, equal shares of bandwidth. The figures present the performance for two of the three nodes participating in the experiment. 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.

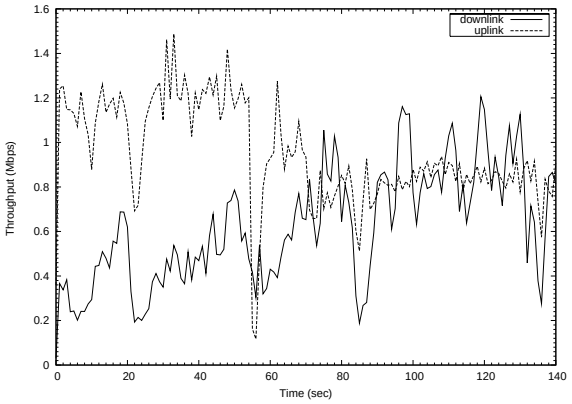

Figure 10: Resolving uplink-downlink unfairness.

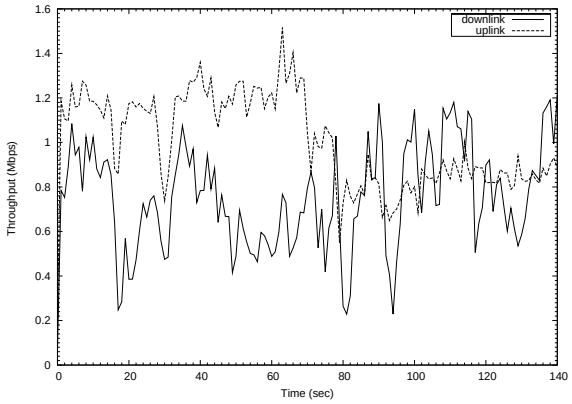

Figure 11: Resolving uplink-downlink unfairness.

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 [9]. 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. remaining 2% was shared equitably among the two ACK classes. The experiment was comprised of 20 HTTP requests for each object and the mean response times can be seen in Fig. 12. Each response time was measured using `gettimeofday()` just before issuing the `wget` command and immediately after `wget` returned. The performance improvement is signifi cant for small objects while it reduces for larger ones. This stems from the limitation imposed to the amount of service that a class can be compensated for, which is controlled by the S_l parameter.

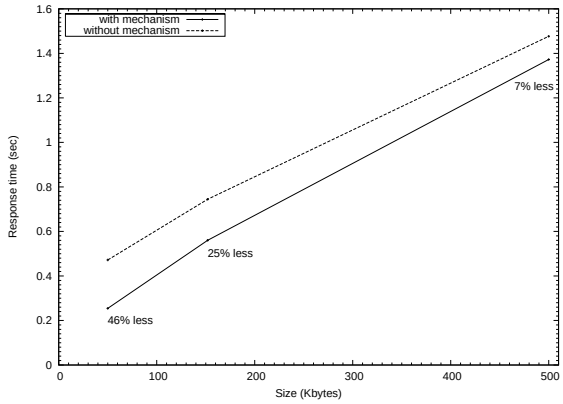

Figure 12: Improving response time for Web traffi c.

Performance anomaly of 802.11. This fi nal experiment presents the behavior of our mechanism in a performance anomaly scenario. We used the setup of Fig. 4 with each node generating an uplink TCP flow destined to the wired host. The ACK class 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 confi guration program. In Fig. 13 we present a typical 802.11 performance anomaly scenario. Node's 1 transmission rate oscillated between 11Mbps and the rest three possible rates, and forced an analogous performance reduction for all nodes. Fig. 14 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 rest nodes. Finally, Fig. 15 is a plot of the quality coeffi cient over time for the same experiment. It presents clearly the node's 1 quality coeffi cient reduction that corresponds to the transmission rate changes and also reveals that the other nodes faced such reductions as well. These, short duration decrements in quality coeffi cient, can be attributed to transmission rate changes caused by packet losses.

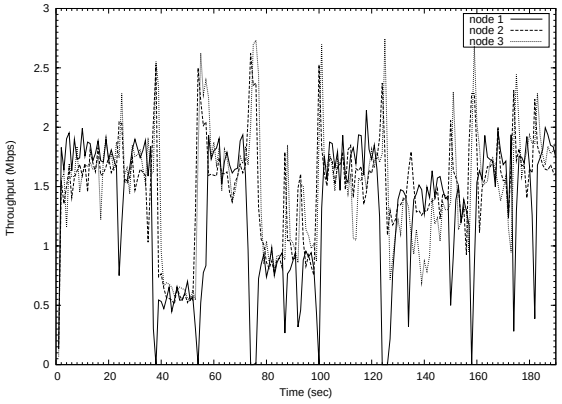

Figure 13: Performance anomaly

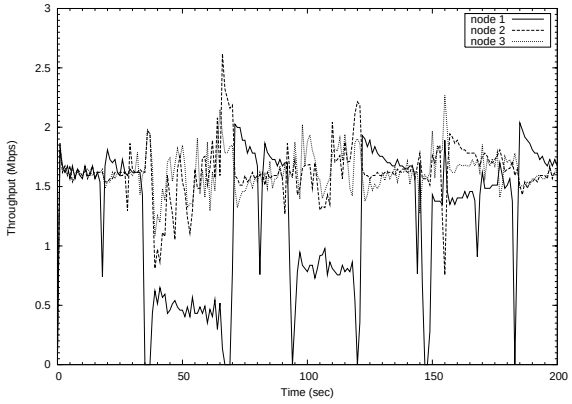

Figure 14: Resolving performance anomaly

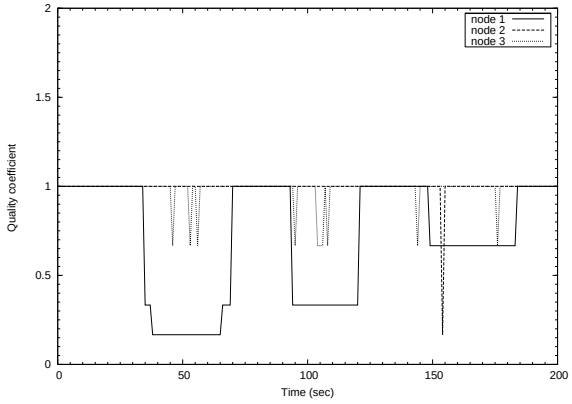

Figure 15: Quality coeffi cient value over time.

5 Related Work

Service differentiation in wireless networks has been studied by several researchers. In [4] 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 traffi c 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. 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 [6], [8], [10] and [3] presenting fair scheduling algorithms. In [6], 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 wireless link-state estimation problem has also been addressed by many researchers. In [1], 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 [2] suggests to employ MAC level acknowledgment failures as a criterion for link quality and change the retransmission attempts accordingly. These algorithms, however, suffer from the same limitations as those presented for the fairness algorithms. Our approach, on the other hand, utilizes the transmission rate as an indication of a link's quality which is readily available at the Access Router.

Recently, in [11], Pilosof *et al.*, proposed the dynamic adjustment of 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 [14] 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.

6 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 802.11g networks are parts of an ongoing work. Furthermore, a single Access Router could service more than one access points by provision of a separate dynamic CBWFQ mechanism for each one.

References

[1] H. Aida, Y. Tobe, M. Saito, and H. Tokuda. A software approach to channel-state dependent scheduling for wireless lans. *WOWMOM*, 2001.

[2] 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.

[3] M. Bottiglienco, C. Casetti, C. Chiasserini, and M. Meo. Smart traffi c scheduling in 802.11 wlans with access point. *VTC IEEE*, 2003.

[4] 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.

[5] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. *IEEE INFOCOM 2003*, 2003.

[6] H.Zhu and G. Cao. On improving service differentiation under bursty data traffi c in wireless networks. *IEEE INFOCOM*, 2004.

[7] J.Pavon and S.Choi. Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement.

[8] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on networking*, 7, August 1999.

[9] B. Mah. An empirical model of http netwrok traffi c. *IEEE*, 1997.

[10] T.S. Ng, I.Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location dependent errors. *IEEE INFOCOM'98*.

[11] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha. Understanding tcp fairness over wireless lan. *IEEE INFOCOM*, 2003.

[12] O. Rusu, M. Subredu, I. Sparlea, and V. Vraciu. Implementing real time packet forwarding policies using htb. *http://luxic.cdi.cz/ devik/qos/htb*.

[13] `http://luxic.cdi.cz/~dvik/qos/htb`. Htb for linux.

[14] Y. Yuan, D. Gu, W. Arbaugh, and J. Zhang. Achieving packet-level quality of service through scheduling in multirate wlans. *VTC IEEE*, 2004.