A RESOURCE-BASED APPROACH TO MAC LAYER INDEPENDENT HIERARCHICAL LINK-SHARING IN WIRELESS LOCAL AREA NETWORKS

Lars Wischhof, John W. Lockwood
Department of Computer Science - Applied Research Lab - Washington University
Campus Box 1045 - 1 Brookings Drive - Saint Louis, MO 63103 - USA
Ph.: (314) 935-4163, Fax: (314) 935-7302, e-mail: wischhof@ieee.org, lockwood@arl.wustl.edu

ABSTRACT

The growing deployment of wireless local area networks (WLANs) in corporate environments and the increasing number of wireless Internet service providers create a need for controlled sharing of wireless bandwidth. In this paper, a method for using long-term channel-state information in order to control the sharing of wireless bandwidth is studied. The algorithm enables an operator of a WLAN to equalize the revenue/cost for each customer in the network and to control the link-sharing based on a combination of user-centric and operator-centric factors. As an example, we adapt one of the well-know wireline schedulers for hierarchical link-sharing, the Hierarchical Fair Service Curve Algorithm (H-FSC), for the wireless environment and demonstrate through simulations that the modified algorithm is able to improve the controlled sharing of the wireless link without requiring modifications to the MAC layer. In addition, we present first measurements of a prototype implementation.

1 INTRODUCTION

The increasing deployment of wireless local area networks and emergence of wireless Internet providers create a need for providing mechanisms for controlled hierarchical sharing of the wireless link. Recent research in the area of wireless scheduling has made much progress in the development of algorithms which are able to guarantee fair sharing of a wireless link and improve throughput by avoiding head-of-line blocking, which occurs when a packet has to be sent to a station experiencing a bad channel-state. Since wireless channel errors are location-dependent, most of the algorithms probe the channel before starting a transmission. If the channel is in a bad state, the transmission is deferred and a different flow is serviced. The skipped flow can be compensated later by using various wireless compensation mechanisms, e.g. by swapping its slot with the slot of the flow with a good channel-state or by using a credit/debit based scheme. Examples for this class of wireless scheduling algorithms are Wireless Packet Scheduling (WPS) [7], Channel-Condition Independent Packet Fair Queuing (CIF-Q) [9], Wireless Fair Service (WFS) [6] and Multi-class Priority Fair Queuing [8].

One reason why these approaches are hard to apply to current hardware is that they rely on accurate knowledge of the channel-state for each mobile destination obtained e.g. by using the RTS/CTS handshake. Therefore, they either need to be implemented as part of the MAC layer, or the MAC layer would need to pass this information to the higher layers in the network protocol stack.

While we assume that some kind of channel-state dependent scheduling will be incorporated in future wireless MAC layers, in this article we study a mechanism which is able to improve the controlled sharing of a wireless link without precise knowledge of the current channel-state. The assumed scenario is the following: An access point schedules downlink traffic with various rate/delay requirements for several organisations/customers sharing the wireless medium. Retransmissions and adaption of the transmission rate towards a specific mobile station are handled by the MAC layer as it is the case in current IEEE 802.11 hardware. Changes to the protocol stack of the mobile station should be avoided, so that a client is able to communicate with the access point without modifications (although the installation of a specific support application might be acceptable).

The remainder of the article is organized as follows: In Section 2, we give some background information on how hierarchical link-sharing architectures are implemented in a wired environment and the problems that arise if the same approach is used in a wireless network. A short problem description and motivation is given and related work is presented. Next, a brief overview of the Hierarchical Fair Service Curve Algorithm (H-FSC) covers the background information necessary in order understand the remainder of article. In Section 3, the wireless link-sharing model is introduced and modifications of H-FSC for a wireless environment are presented. The performance of the algorithm is studied through simulations in Section 4. First measurements of a prototype implementation are shown in Section 5, and conclusions are presented in Section 6.

2 BACKGROUND

In local area networks based on the IEEE LAN standards, mechanisms for hierarchical link-sharing and fluid fair queuing are implemented as part of the operating system or device driver, simply because the 802.X MAC and LLC do not provide the necessary mechanisms for a controlled sharing of the available bandwidth. An advantage, if a hierarchical link-sharing scheduler such as Class Based Queuing (CBQ) [4] or the Hierarchical Fair Service Curve Algorithm (H-FSC) [11] is part of the operating system, is its easy and flexible configurability and independence of specific networking hardware.

An important observation is that by implementing these mechanisms above the MAC layer, the scheduler does not see any MAC layer retransmissions and therefore schedules expected goodput.[1] Since the error probability in a wired LAN is low and all stations experi-

[1] In this article the term *goodput* is used for the amount of data seen above the MAC layer, whereas the term *raw throughput* is used for the capacity of the wireless link. Although this is an approximation since successful delivery via the link layer is not guaranteed it is a valid estimation because we assume a MAC layer correcting the majority of the wireless link errors e.g. by using retransmissions as in the case of IEEE 802.11.

ence the same link quality, the assumption that goodput is equal to throughput does not create unfairness. The opposite is true in case of a wireless LAN. The error probability on a wireless link is significantly higher, which results in a larger number of retransmissions. And since in this case the link quality is location-dependent, scheduling equal goodput to two mobile stations can lead to completely different shares of consumed raw throughput. While this is desirable in some cases (e.g. in order to be able to compensate a user for a bad link quality), it might be inefficient/unwanted in others.

2.1 Motivation

In order to illustrate the problem, this section presents a simplified example scenario, which will also be used later on in order to show the benefits of our approach: An Internet Service Provider (ISP) offers services using currently available (e.g. IEEE 802.11) technology on a wireless link with a total capacity[2] of 6 Mbit/s. The ISP has installed an access point in order to service two different companies: Company A has bought 85% of the bandwidth for $85 per month and Company B the remaining 15% for $15. For simplicity, it is assumed that each company has only one mobile station using the access point, Mobile Station 1 (MS 1) belongs to Company A and Mobile Station 2 (MS 2) to Company B. The correct sharing of the bandwidth is to be enforced at the access point using a link-sharing scheduler.

One can now describe the effort needed to transmit data to a mobile destination by using a variable g corresponding to the ratio of the amount of data received at a destination to the necessary resources. Thus, if a mobile station would switch to a modulation enabling it only to receive data at rate of $\frac{1}{11}$th of the full speed, its value of g would change to $g = \frac{1}{11}$. For a link-sharing scheduler implemented above the MAC layer, the only visible effect is a decrease of the total link bandwidth. It would schedule the remaining capacity according the 85% to 15% weights for both companies. This leads to unfairness in terms of costs per used resource for each customer as soon as one assumes different values of g for each mobile station, e.g. if MS 1 is located next to the access point (has a perfect channel, $g_{ms1} \approx 1$) and MS 2 experiences a bad channel quality making retransmissions or adaptive modulation necessary and resulting in a low value of g_{ms2}. Figure 1(a) shows an example[3] where the value of g for MS 2 was varied while MS 1 had a constant value of $g_{ms1} = 1$. Figure 1(b) illustrates the influence of g_{ms2} on the amount paid per consumed resource for each mobile station.

This demonstrates two effects which are unwanted for the ISP: 1) The behavior of Company A drastically influences the bandwidth available to Company B. 2) Although Company A only pays for 15% of the link bandwidth, it is able to utilize a much larger share of the available resources leading to unfairness in terms of costs per unit resource consumption.

In order to avoid such a scenario, the scheduler needs to be able to take into account the amount of resources (in terms of raw bandwidth) which are needed to provide a specific service to a user. This would enable an ISP to make Service Level Agreements (SLAs) not only based on goodput but also specifying the conditions under which this goodput can be expected, e.g. a user will be guaranteed a rate of 64kbit/s as long as his signal to noise ratio (SNR) is above a specified value. After this point the scheduled rate for him will be reduced in order to limit his consumption of the network resources. Note that such a limitation makes sense only in a situation where not enough bandwidth is available.

2.2 Monitoring per Destination Goodput/Throughput Ratio

A simple way for a packet scheduler above the MAC layer to estimate the raw throughput consumption caused by transmitting a specific amount of data to a destination is the concept of monitoring the goodput to throughput ratio (GTR) per destination. The idea is that an entity, the *Wireless Channel Monitor*, which is either part of the MAC layer, the lower OS layers or even the scheduler itself (a possible implementation is presented in Section 5), keeps a windowed average of the goodput achieved per byte of consumed resource in terms of raw throughput. The GTR is then used by the scheduler in order to estimate used raw bandwidth when handing data for a specific destination to the MAC layer. The advantage of this approach is that it allows a unified consideration of the effects of various MAC layer techniques as retransmissions, forward error control (FEC), and adaptive modulation.

The model assumes that the raw bandwidth available is constant for a specific device, e.g. 11 MBit/s for a 802.11b WLAN NIC. If the card changes its modulation to transmit at 5.5 Mbit/s, this change is reflected by a corresponding change of the GTR.

2.3 Related Work

Apart from the various wireless scheduling algorithms [6] [7] [8] [9] already mentioned in the introduction, the controlled sharing of a wireless link has been addressed by several authors. Fragouli et. al. propose a modified version of CBQ [4] which uses channel-state dependent packet scheduling (CSDPS) in order to improve the throughput when applying CBQ to a wireless link [10]. The RTS/CTS mechanism is used to determine the channel-state, and the algorithm schedules based on MAC layer goodput. An CSDPS implementation based on the signal-level indicated by a current 802.11 MAC is presented in [3]. In [2], Choi develops a bandwidth management algorithm for adaptive QoS in a CDMA system which is based on the bandwidth usage efficiency of a connection, a property similar to the GTR used in our paper. However, he assumes that only a number of discrete values can occur for this property and uses a set of adaptation rules in order to react to changes.

2.4 Hierarchical Fair Service Curve Algorithm

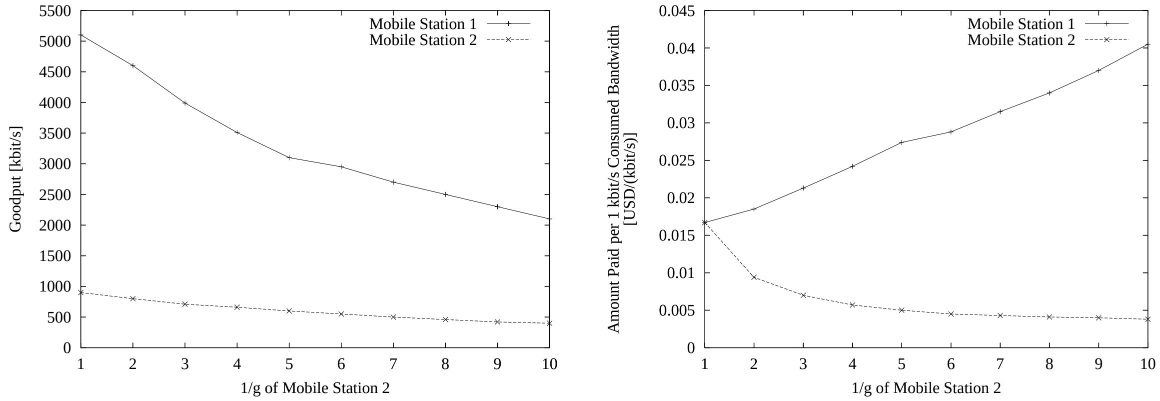
This section gives a brief introduction to the Hierarchical Fair Service Curve Algorithm (H-FSC) [11], which will be modified in the next sections in order to take the consumption of wireless resources into account. H-FSC is based on the concept of a *service curve* which defines the QoS requirements of a traffic class or single session in terms of bandwidth and priority (delay). A session i is guaranteed a service curve $S_i(t)$ if the following equation holds

$$S_i(t_2 - t_1) \leq w_i(t_1, t_2); \;\; t_1 < t_2 \qquad (1)$$

in which t_2 is an arbitrary packet departure time at which the session is backlogged, t_1 is the start of this backlogged period, and $w_i(t_1, t_2)$ is the amount of service received by session i in the interval $(t_1, t_2]$. Usually, only linear or piece-wise linear service curves are used for simplic-

[2] Although this is an arbitrary value, it is close to the maximal amount of goodput one usually achieves using 11 MBit/s 802.11b WLAN cards.

[3] The results were obtained using a simulation of Class-Based Queuing[4] with a separate class for each company. For more details on the simulation environment used refer to Section 4.



(a) The goodput of *both* stations is reduced since MS 2 consumes more than its share of the available resources.

(b) The amount paid per consumed resource by the customer with a perfect link (MS 1, $g_{ms1} \approx 1$) and the customer with decreasing link quality (MS 2) diverge.

Figure 1: Effect of decreasing link quality of MS 2 ($\frac{1}{g_{ms2}}$ increases) on both mobile stations.

ity. A concave[4] service curve results in a lower average and worst case delay than a convex curve with the same asymptotic rate. However, a server can only guarantee all service curves $S_i(t)$ if the service $S(t)$ it provides is always larger than or equal to their sum:

$$S(t) \geq \sum_i S_i(t) \tag{2}$$

In H-FSC, each class in the hierarchy has a service curve associated with it. Excess bandwidth is distributed according to the service curves, and a service which received excess service will not be punished later (fairness properties). However, in their paper [11] Stoica et. al. prove that with non-linear service curves (which are necessary if bandwidth and delay properties are to be decoupled) it is impossible to avoid periods in which the scheduler is unable to guarantee the service curves of all classes or it is not able to guarantee both the service curves and fairness properties. Therefore, the H-FSC algorithm only guarantees the service curves of leaf classes and tries to minimize the difference between the service an interior class receives and the amount it is allocated according to the fairness properties.

This is achieved by using two different criteria for selecting packets: A *real-time criterion*, which is only used when the danger exists that the service curve of a leaf class is violated, and a *link-sharing criterion*, which determines the next packet otherwise. Therefore, each leaf class has a triplet (e_i, d_i, v_i) associated with it which represents the eligible time, the deadline and the virtual time. An interior class only maintains its virtual time v_i. If, at time t, a leaf class with $e_i < t$ exists, there is danger of violating a service curve and the real-time criterion is used to select the packet with the minimal eligible time. Otherwise, the link-sharing criterion searches the packet with the minimum virtual time, starting at the root class. Since the virtual time of a class represents the normalized amount of work received by a class, the goal of the link-sharing criterion is to minimize the difference between the virtual time v_i of a class and that of any sibling. The virtual time v_i^s of the system is computed as the mean of the minimal and maximal virtual times of any class by $v_i^s = (v_{i,\min} + v_{i,\max})/2$. For each leaf class, the algorithm also maintains an eligible curve $E_i(a_i^k;\cdot)$ and a deadline curve $D_i(a_i^k;\cdot)$ where a_i^k is the start of active period number k of class i and a variable c_i, the amount of service a class received under the real-time criterion.

The deadline curve is initialized to the corresponding service curve of the class. At each point in time a_i^k when the class becomes active, the deadline curve is updated according to Equation 3, which by taking into account only the service received under real-time criterion does not punish a session for using excess service.

$$D_i(a_i^k;t) = \min(D_i(a_i^{k-1};t), S_i(t - a_i^k) + c_i(a_i^k)); \quad t \geq a_i^k \tag{3}$$

If a_i is the last time that session i has become active, then the eligible curve represents the maximum amount of service a class can receive under real-time criterion if it is continuously backlogged in the interval $(a_i, t]$ and is defined as

$$E_i(a_i;t) = D_i(a_i;t) + [\max_{\acute{t}>t}(D_i(a_i;\acute{t}) - D_i(a_i;t) - S_i(\acute{t} - t))]^+; \quad t \geq a_i \tag{4}$$

For a detailed description and analysis of the algorithm the reader is referred to [11].

3 Wireless Link-Sharing

In order to illustrate our approach of goodput to throughput ratio based scheduling, we integrate it in the H-FSC algorithm. However, since the H-FSC algorithm was developed for fixed rate links, using it in a variable rate wireless environment makes other modifications necessary. To summarize, the identified issues are:

- The scheduling does not take into account the raw throughput needed to transmit a packet.
- An active session can aquire unlimited backlog leading to the starvation of new sessions.
- If sessions are serviced at a reduced rate, their delay guarantees cannot be kept. In the worst case, for a real-time session this can lead to a situation where only worthless, outdated packets are sent.

The example (Figure 2) used to illustrate the behavior of the modified H-FSC algorithm is an extended version of the scenario presented in Section 2.1. Here, the wireless link is shared between two agencies, each of them using Voice over IP, WWW and FTP services. It is also

[4] A service curve is *concave* if its second derivative is non-positive and not the constant function zero. Analogously, it is *convex* if its second derivative is positive and not the constant function zero.

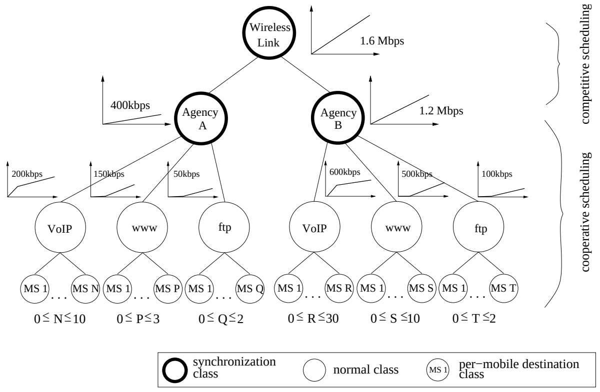


Figure 2: Link-sharing hierarchy used in the simulated example scenario.

Agency	Traffic Type	m_1 [Mbit/s]	d [ms]	m_2 [Mbit/s]
A	VoIP	0.030	20	0.020
	WWW	0.000	20	0.050
	FTP	0.000	20	0.025
B	VoIP	0.030	20	0.020
	WWW	0.000	20	0.050
	FTP	0.000	20	0.050

Table 1: Service curve parameters for leaf classes in the example scenario shown in Figure 2.

assumed that a leaf class has been created for every mobile destination using a specific service. Table 1 lists the service curve parameters for the leaf classes. Two piecewise linear service curves [11] are used: m_1 is the slope of the first segment, d is the x-coordinate where the two pieces intersect, and m_2 is the slope of the second segment.

3.1 Competitive vs. Cooperative Scheduling

A simple solution to avoid the unfairness caused by different goodput to throughput ratios of mobile stations would be to base the scheduling completely on the resource consumption of a mobile, i.e. by defining the service curve in the raw throughput domain. However, this would lead to an unwanted behavior when scheduling traffic for different traffic classes within the subtree of each agency. A simple example would be that of two different mobile stations, each with a specified resource share of 20 kbit/s for VoIP. If MS 1 has a goodput to throughput ratio of $g_{ms1} = 1$, MS 2 has $g_{ms2} = \frac{1}{10}$, and the scheduler has a share of 220 kbit/s of raw throughput available, in a purely resource-consumption based model it will schedule a fraction of 110 kbit/s of the available resources for each mobile, leading to excess service for MS 1 and only 11 kbit/s MAC layer goodput for MS 2 causing it to drop its VoIP connection, whereas the desired behavior would be to guarantee the minimum needed service for each mobile station before scheduling excess bandwidth.

Therefore, a wireless link-sharing model has to be able to take resource consumption based constraints *and* expected goodput based constraints into account. Thus, one can define two types of scheduling which need to be supported by the algorithm:

- *Competitive Wireless Scheduling:* Resource-consumption based scheduling between competing entities (e.g. two different customers). Service curves specify the amount of resources in terms of raw bandwidth which the scheduler needs to provide for a class if it has demand. (In the example, this corresponds to the scheduling between Agencies A and B.)
- *Cooperative Wireless Scheduling:* Goodput based scheduling between two classes (e.g. within one company). Service curves specify the amount of MAC layer goodput needed by a class. No excess bandwidth is scheduled to any class as long as another class in the subtree is unsatisfied. Excess bandwidth is scheduled independently of the GTR of a class.

3.2 Synchronization Classes

In order to integrate both scheduling types in one model, a class which synchronizes both approaches is needed. This type of class is called *synchronization class* in the remainder of the paper. Its algorithmic representation will be explained in the next section, but usually cooperative scheduling is performed in the subtree below the synchronization class, and the subtree is integrated in a competitive scheduling environment by using the synchronization class. The root class of the link-sharing tree is always a synchronization class since the raw throughput of the wireless device is assumed to be constant. In the example shown in Figure 2, the classes for Agency A and Agency B are both synchronization classes.

3.3 Modifications to H-FSC

In the following, the modifications to the H-FSC algorithm are presented. First, the cooperative scheduling within a subtree is described both in situations where enough bandwidth is available and under overload conditions, then competitive scheduling in the modified H-FSC is shown. Finally, a simple way to support dropping of outdated packets for real-time classes is shown.

3.3.1 Cooperative Scheduling

As defined in Section 3.1, each class for which cooperative scheduling is performed is part of a subtree which has a synchronization class as its root. Cooperative scheduling within this subtree (e.g. the subtree of which the “Agency A” class is root in Figure 2) in situations where enough resources are available (Equation 2 is valid for the subtree) is similar to the behavior of an unmodified H-FSC scheduler: Service is provided without taking destination specific resource-consumption into account. The virtual time of a class within the subtree, which determines the distribution of bandwidth using the link-sharing criterion, corresponds to the amount of data handed down to the MAC layer. Implicitly, stations experiencing a bad link are allowed to use a larger share of the raw bandwidth in order to be able to compensate the low link quality e.g. by doing FEC, retransmissions or using a lower transmission rate.[5] Therefore, compared to an unmodified H-FSC scheduler implemented above the MAC layer, the only difference is that the goodput scheduled is based on the amount of resources available for this subtree.

In an overload state, the amount of available resources available for a subtree s is insufficient to guarantee all service curves since it is less than the sum of the eligible curves of all active sessions in the subtree $\sum_{i\in\mathcal{A}_s(t)} E_i(a_i;t)$. Since the root of a cooperative scheduling subtree is a synchronization class, the service curve is

[5]The scheduler is only able to guarantee delay constraints with an accuracy of the time needed to transmit one maximum length packet via the MAC layer (including retransmissions, FEC, back-off etc). Various MAC layer mechanisms are proposed in order to provide differentiated services in the MAC layer, these are out of the scope of this paper but could be used in order to be able to give tighter delay guarantees.

resource-based, and the total amount of resources available for the subtree within an arbitrary interval $(t_1, t_2]$ is $S_s(t_2) - S_s(t_1)$. Under overload, these resources are completely consumed, and the following equation must hold for all active leaf classes $\mathcal{A}_s(t)$:

$$S_s(t_2) - S_s(t_1) = \sum_{i \in \mathcal{A}_s(t_1)} \frac{1}{g_i \cdot d_i} \cdot (E_i(a_i, t_2) - E_i(a_i, t_1)), \quad \mathcal{A}_s(t_1) = \mathcal{A}_s(t_2) \tag{5}$$

Here, d_i is a class specific factor by which the service for this class is reduced, and the goodput to throughput ratio $g_i(t)$ is approximated to the constant g_i within the interval $(t_1, t_2]$. If, analogously to the approach in the goodput domain, the raw bandwidth share of each class is to be degraded by a constant factor, then d_i has to be determined in a way that $\frac{1}{g_i \cdot d_i} = const$. The solution is to determine the d_i for each class as the fraction of the required GTR over GTR of the specific leaf class:

$$d_i = \frac{\left(\frac{K}{S_s(t_2) - S_s(t_1)}\right)}{g_i} \quad \text{where} \quad K = \sum_{i \in \mathcal{A}_s(t_1)} (E_i(a_i, t_2) - E_i(a_i, t_1)) \tag{6}$$

Whereby $\frac{1}{g_i \cdot d_i} = \frac{S_s(t_2) - S_s(t_1)}{K}$ and the condition given in Equation 5 is satisfied.

Roughly speaking, this approach reduces the (goodput) service of each class in a way that the raw bandwidth available to each class is distributed proportional to the service curves and guarantees that the subtree consumes only the amount of resources available to the synchronization class which is its root.

3.3.2 *Competitive Scheduling*

The modified H-FSC scheduler performs competitive scheduling based on raw throughput (resource-consumption) among synchronization classes. Since the algorithm defined in the previous section guarantees that a synchronization class is never allocated more than the amount of resource specified by its service curve by using the real-time criterion, only the sharing of bandwidth using the link-sharing criterion needs to be considered. In contrast to cooperative scheduling, now the virtual time of a class is based on the resource consumption by incrementing it by the product of $\frac{1}{g_i}$ and the packet size whenever a packet is transmitted for a class. The following example illustrates the incrementation of virtual times: If one assumes that subclass "MS 1" of Agency B in Figure 2 transmits a packet of size s_p using the link-sharing criterion on a link with $g_{ms1} = \frac{1}{10}$, the virtual time of the class itself and of the VoIP class will be incremented by s_p and the virtual time of Agency B and of the wireless link root class will be incremented by $\frac{1}{g_{ms1}} \cdot s_p = 10 \cdot s_p$.

3.3.3 *Packet Dropping*

For a real-time class the reduction of its rate in an overload situation can lead to an unwanted accumulation of packets in its queue increasing the delay of each packet. For this traffic class, a mechanism is needed which allows the dropping of outdated packets allowing the usage of the remaining capacity for packets which have not exceeded their maximal delay. A simple way to integrate this behavior in the H-FSC scheduler is the concept of a packet drop service curve. It is used to compute a hard deadline after which a packet is outdated and will be dropped from the queue in the same way that the transmission deadline is calculated using the regular service curve.

Traffic Type	Packet Size [byte]
Voice over IP	64
WWW	512
FTP	1024

Table 2: Packet sizes (at network layer, including IP header) of traffic types used the in simulations.

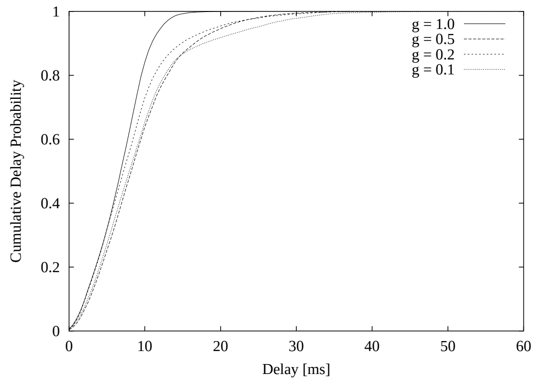


Figure 3: Effect of g_{ms2} (MS 2, Agency A) on the packet delay of VoIP traffic of a mobile in Agency B. The delay of VoIP packets of the competing agency remains almost unchanged although the link quality for MS 2 in the other agency decreases dramatically.

4 SIMULATION

The described modifications of H-FSC were implemented and simulated in the Linux TCSIM[1] environment, which was extended with a wireless component to be able to simulate the destination specific behavior of the wireless link. Adaptive modulation with a specific goodput to throughput ratio g_i is simulated by increasing the time needed to transmit a packet accordingly. The implementation of the wireless channel monitor is done as a separate module, which estimates the factor g_i by keeping an exponentially weighted average of the (goodput) transmission rate for every destination mobile. Because of space limitations and since only minor changes were made influencing the behavior in a state where enough bandwidth is available, the two simulated scenarios presented in this article concentrate on the behavior of the scheduler in an overload situation.[6]

SCENARIO 1: In the first scenario, all classes present in Figure 2 are transmitting at their maximum rate. The VoIP traffic is enqueued at a constant rate of 20 kbit/s (similar to VoIP using a G.728 codec) and a packet a the head of line is dropped if it is more than 8 ms over its deadline. FTP/WWW traffic is enqueued at the maximum rate following a Poisson distribution. Table 2 lists the used packet sizes. All mobiles have a perfect channel (g_i=1) except for MS 2 of Agency A whose GTR is varied.

Figure 3 shows the influence of the transmission condition for a mobile in Agency A on the cumulative delay probability for VoIP packets transmitted to a mobile station in Agency B. For $g_{ms2} = 1$ the system is not in an overload condition and the scheduler is able to guarantee a delay below 20 ms. With the decrease of g_{ms2}

[6] More detailed simulations and measurements have been published in a technical report [12].

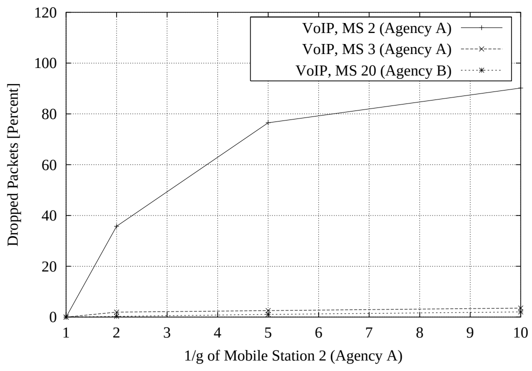


Figure 4: Effect of g_{ms2} on the percentage of packets dropped for the mobile experiencing the bad link (MS 2), a mobile of the same Agency (MS 3), and a mobile of Agency B (MS 20) in an overload situation. MS 20 is not affected because of the competitive scheduling among the two agencies, and the service for MS 3 also is not reduced below its minimal share of resources, with which in this case ($g_{ms3} = 1$) it is still able to send most packets on time.

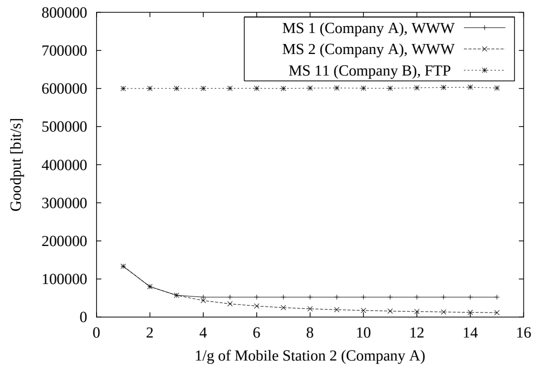


Figure 5: Goodput available for selected mobile stations in Scenario 2. As the amount of resources needed to transmit to MS 2 (and MS 3) of Agency A increases, the goodput available for the competing Agency B is not affected. As long as excess bandwidth is available ($\frac{1}{g} < 4$) within Agency A, it is used to compensate the low link quality of MS 1 (and MS 3).

(the mobile station of Agency A needs more bandwidth in order to achieve its desired service) the scheduler has not enough resources available to guarantee all service curves. However, the simulation illustrates that the algorithm it is able to limit the impact on a mobile in the competing agency. Figure 4 compares the number of dropped packets for the two mobile stations. Since the scheduler is able to guarantee Agency B its share of the available resources, the number of dropped packets for a mobile station of Agency B does not significantly increase. In addition, since the scheduler in an overload state reduces the amount of service for subclasses within a cooperative scheduling subtree based on resource consumption, the service for a mobile in Agency A experiencing a perfect channel is also not reduced. (The way the hierarchical link-sharing is configured in this example assumes that it is better to significantly reduce the service for the customer experiencing the bad link (i.e. drop his connection) than to reduce the VoIP share of the other mobiles below their minimal service level.)

SCENARIO 2: This scenario demonstrates the effects of competitive and cooperative scheduling. For Agency A only three WWW traffic sources are transmitting (Poisson distributed, 160 kbit/s), Agency B has two customers transmitting FTP traffic (Poisson distributed, 650 kbit/s). All other classes are idle. The mobile stations of Agency A and the first mobile station of Agency B have a GTR of $g_i = 1$, whereas for MS 2 and MS 3 of Agency A the long-term channel quality is varied. In Figure 5, the variation of the bandwidth available to one of the FTP customers and two WWW customers, of which MS 1 is experiencing a constantly good and MS 2 is experiencing a varying channel quality, is shown. The competitive scheduling among the two agencies guarantees Agency B a share of 1200 kbit/s of raw bandwidth regardless of the channel quality which a mobile in Agency A experiences. Since both FTP customers of Agency B have perfect channels, each is able to achieve a goodput of 600 kbit/s.

As long as $\frac{1}{g}$ of MS 2 (and MS 3, which is not shown in the graph since it receives the same goodput) is less than 4, excess bandwidth is available to the WWW customers of Agency A and because the scheduler is configured to use cooperative scheduling within an agency, it is distributed goodput-based and MS 1 and MS 2 achieve the same rate of goodput. (Therefore, MS 2 and MS 3 are allowed to consume a larger share of resources than MS 1 in order to be able to compensate for the low link quality.) When $\frac{1}{g}$ becomes larger than 3, not enough bandwidth is available to guarantee the service curves of all WWW customers (50 kbit/s). Therefore, the goodput of MS 2 is degraded based on its resource-consumption in order to avoid violating the service curve of MS 1 because of the bad channel conditions of MS 2 and MS 3.

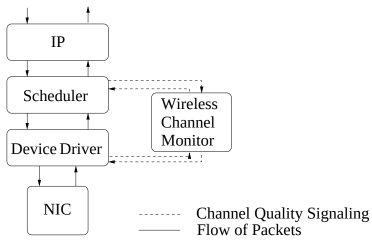


Figure 6: Structure of prototype implementation.

5 PROTOTYPE IMPLEMENTATION

This section presents first results of a prototype implementation of the modified algorithm based on the NetBSD implementation of H-FSC [11] in a Linux kernel (V. 2.4.5). Figure 6 illustrates the structure of the implementation.

For the implementation of a wireless channel monitor there are several possible sources of information which can be used to determine the goodput to throughput ratio g_i for a given destination i: monitoring the time between dequeue events, the status interrupt of the wireless card, the signal quality status byte [3], and the currently used adaptive modulation of the MAC. The current prototype implementation uses the signal strength reported by the MAC and maps it to an expected value for g_i. Because the MAC of the used wireless cards updates the signal strength information only upon reception of a packet, the current system forces the mobile stations to send packets in regular intervals by sending *ICMP echo request* packets every second.

The testbed consisted of a desktop PC (Pentium 120 Mhz) serving as the access point and two mobile stations (laptops, Celeron 550 Mhz processor). All three stations were using the PCMCIA card version of the *Raylink* 2 MBit/s FHSS WLAN cards [5] conforming to the IEEE 802.11 standard. The testbed scenario is similar to the one presented in Section 2.1: The scheduler was con-

Pos. MS 2	MS	Sched.	Avg. [kbit/s]	Min. [kbit/s]	s [kbit/s]
good	1	mHFSC	1022	849	42
		CBQ	949	870	18
	2	mHFSC	154	80	30
		CBQ	228	198	10
med.	1	mHFSC	877	634	80
		CBQ	580	384	58
	2	mHFSC	68	30	14
		CBQ	138	46	20
bad	1	mHFSC	1048	329	108
		CBQ	270	215	19
	2	mHFSC	0	0	-
		CBQ	0	0	-

Table 3: Comparison of first results obtained using an early prototype implementation of the modified H-FSC algorithm (mHFSC) with those of a not wireless aware CBQ scheduler.

figured to share the link on a resource-based criterion by using a separate synchronization class for each mobile/company: 80% of the link were assigned to MS 1 and 20% to MS 2. As a comparison, the same setup was used with the CBQ scheduler included in the kernel, configured to share the link in the same manner. The ARP cache on the access point was set up statically in order to avoid side effects on the measurements. The access point was connected to a 10 Mbit/s Ethernet in which a fixed host generated 100 byte UDP packets to each of the mobile destinations: constant bit-rate traffic of 2 Mbit/s for MS 1 and 0.48 Mbit/s for MS 2 – therfore guaranteeing that the classes of both mobile stations where always backlogged. MS 1 was constantly in a good position (avg. signal level indicated by MAC $\approx$ 170), and the position of MS 2 was varied from good, medium (avg. signal level $\approx$ 110) to bad (no signal). Eleven measurements of 300 seconds were done, the bit-rate was calculated over a window of one second.

Figure 7 shows the performance of the algorithm in the three tested situations, and Table 3 lists the average bit-rates, the minimum bit-rates and the standard deviation *s*. Because the modified HFSC scheduler was configured to enforce competitive scheduling among the two mobile stations, it is able to decrease the influence of the position of MS 2 on the amount of resources available to MS 1. Even in the case where MS 2 is in a bad position, the algorithm provides MS 1 its full share of the network resources, whereas under CBQ it only receives one fourth of it. Furthermore, the results indicate some minor problems of the current prototype implementation: Since the GTR of a station is estimated only based on the currently reported channel quality and then used to modulate the amount of data scheduled for a station, the bit-rate has high variance. In addition, the currently used method seems to favor higher rate classes because they are able to update their channel status information more often. Both observations illustrate the need for a more accurate channel monitor. Therefore, an improved version currently being developed will take more sources of information into account, namely transmission interrupts and the currently used MAC modulation.

6 Conclusions

This article presented an approach to use long-term information on a mobile stations channel-state in order to perform controlled sharing of a wireless link. A modified version of the Hierarchical Fair Service Curve Algorithm was introduced which enables an operator of a WLAN to specify link-sharing constraints based on a combination of cooperative (goodput based) and competitive (resource-consumption based) scheduling. Simulations and early prototype results indicate that the algorithm is able to guarantee a specific amount of resources to a customer and therefore can be used to equalize the cost per revenue. The prototype implementation of the algorithm in a Linux kernel demonstrates that the proposed link-sharing model can be used on currently available WLAN hardware conforming to the IEEE 802.11 standard.

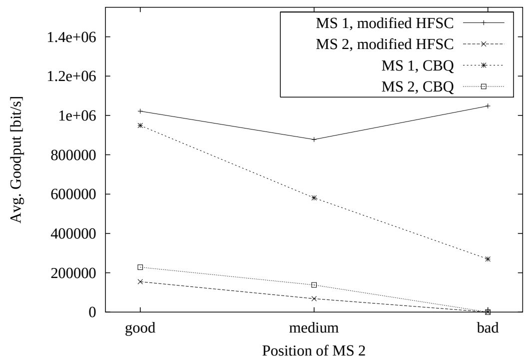


Figure 7: Influence of the position of MS 2 on average goodput of both mobile stations. The modified HFSC scheduler is configured to perform competitive scheduling among the mobile stations and successfully decreases the influence of the link quality of MS 2 on the performance experienced by MS 1.

References

[1] Werner Almesberger. Linux traffic control - implementation overview. Technical report, EPFL, Nov 1998. http://tcng.sourceforge.net.

[2] Sunghyun Choi. *QoS Guarantees in Wireless/Mobile Networks*. PhD thesis, University of Michigan, 1999.

[3] Saman Desilva and Samir R. Das. Experimental evaluation of channel state dependent scheduling in an in-building wireless lan. In *Proceedings of the 7th International Conference on Computer Communications and Networks (IC3N)*, pages 414–421, Lafayette, LA, October 1998.

[4] Sally Floyd and Van Jacobsen. Link and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4), 1995.

[5] Raylink Inc. Raylink user manual. http://www.raylink.com/pdf/pccard.pdf.

[6] S. Lu, T. Nandagopal, and V. Bharghavan. Design and analysis of an algorithm for fair service in error-prone wireless channels. *Wireless Networks Journal*, Feb. 1999.

[7] Songwu Lu, Vaduvur Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *Proceedings of ACM SIGCOMM 1997*, 1997.

[8] Jay R. Moorman and John W. Lockwood. Multiclass priority fair queuing for hybrid wired/wireless quality of service support. *WoWMOM 99 - Proceedings of The Second ACM International Workshop on Wireless Mobile Multimedia*, pages 43–50, Aug. 1999.

[9] T. S. Eugene Ng, Ion Stoica, and Hui Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. *Proceedings of IEEE INFOCOMM*, page 1103, 1998.

[10] M. Srivastava, C. Fragouli, and V. Sivaraman. Controlled multimedia wireless link sharing via enhanced class-based queueing with channel-state-dependent packet scheduling. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 572–, April 1998.

[11] Ion Stoica, Hui Zhang, and T.S. Eugene Ng. A hierarchical fair service curve algorithm for link-sharing, real-time and priority service. In *Proceedings of SIGCOMM '97*, 1997. http://www.cs.cmu.edu/∼zhang/HFSC/.

[12] Lars Wischhof and John Lockwood. Packet scheduling for link-sharing and quality of service support in wireless local area networks. Technical Report WUCS-01-35, Applied Research Laboratory, Washington University in St. Louis, November 2001.