Fair Bandwidth Allocation of a Wireless Base Station

György Miklós
Traffic Analysis and Network Performance Laboratory
Ericsson Telecommunications Hungary
Email: Gyorgy.Miklos@eth.ericsson.se
H-1037 Budapest Laborc u. 1., Hungary
Tel: +36 1 437 7633, Fax: +36 1 437 7219

Sándor Molnár
High Speed Networks Laboratory,
Dept. of Telecommunications and Telematics,
Technical University of Budapest
Email: molnar@bme-tel.ttt.bme.hu
H-1117 Budapest Pázmány Péter sétány 1/D, Hungary

March 31, 1999

Abstract

Fair allocation of the capacity of a wireless base station is becoming important due to the increasing use of wireless access technologies which often become the bottleneck of the network. In this extended abstract we present a fair allocation mechanism and extend it by a simple loss compensation scheme. We do not assume any knowledge or prediction of the channel quality ahead of transmission. We present some initial simulation results of the presented mechanisms.

1 Introduction

Cellular wireless systems are becoming increasingly common as access systems with the development of several new technologies (e.g. GPRS, UMTS, Wireless LAN systems). With the deployment and increasing use of these access technologies, the capacity of a radio cell can frequently become the bottleneck point of the network. Therefore, the efficient sharing of this capacity is vital.

Sharing the bandwidth of a wired link fairly has been an important area of research [1, 6]. In a wireless environment however a number of specific problems arise: there can be considerable location-dependent packet loss due to bit errors and the number of users in a radio cell changes due to mobility of users. Furthermore the MAC protocol of the wireless link has to cope with a number of problems: e.g. the base station (BS)

may not have complete knowledge of the transmission needs of the mobile terminals (MTs) yet it has to allocate the uplink and downlink transmission capacity.

There has been recent work [4, 5] which addresses wireless fair scheduling. These papers however assume knowledge or prediction of the state of a wireless link before transmission. This assumption may not be realistic: in many cases we have knowledge about the success of a transmission only after the transmission.

The focus of this abstract is the fair allocation of the capacity of a wireless base station. The abstract does not address the question of providing delay guarantees to flows; these can be provided by other priority mechanisms. Our focus here is on flows that are elastic, i.e. can adapt to the available capacity. We present a basic allocation mechanism (section 3) and address briefly the question of how losses can be accommodated in the fairness model (section 4).

2 Architecture

Figure 1 illustrates the architecture considered in this abstract. An example to the architecture is the HiperLAN wireless LAN system [2], but the architecture itself is general.

Each MT corresponds to a user. The BS is represented by many users, each interfacing a MT. The BS contains a master scheduler that is responsible for the distribution of the capacity between the users.

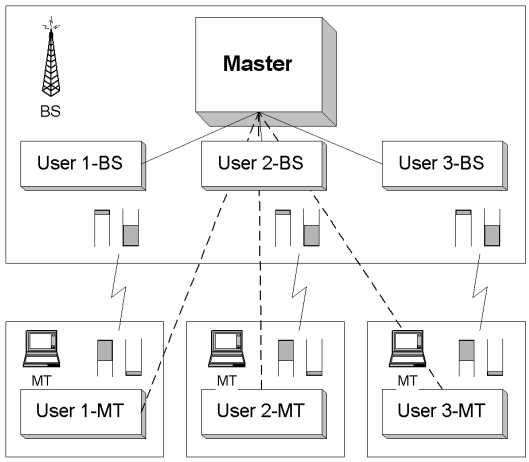

Figure 1: System architecture

The system uses time division multiple access scheme. The MAC layer is frame-based, and the capacity of a frame can dynamically be distributed by the master-scheduler.

The master scheduler makes its resource allocation decisions based on the resource requests from the users. Due to the centralized scheduling, collisions are avoided. For the users within the BS the resource requests are just internal information within the base station. For the users in the MTs the resource requests are made to the master scheduler as control information. Resource requests specify the amount of data that

the user has in its buffers to transmit.

The master scheduler runs a scheduling algorithm once for each frame. This algorithm takes the resource request from the users as its input, and determines how much each user is allocated in the next frame in the unit of a radio PDU. In this abstract, we do not deal with the order of allocations within a frame; our focus is the amount of allocations determined by the master scheduler for each frame.

3 Basic allocation mechanism

This section presents the basic allocation mechanism used for the distribution of capacity of one MAC frame. We only present the allocation mechanism; its analysis is beyond the scope of this abstract. The basic allocation mechanism can be regarded as the adaptation of start-time fair queuing [3].

The unit of allocation is one radio PDU, which is assumed to be of fixed length. The MAC frame is of fixed length, and can accommodate C PDUs (where C is integer). The scheduler takes as its input r_i for $i \in \mathcal{U}$ where $\mathcal{U}$ is the set of currently active users, r_i is the resource request giving the number of PDUs waiting at a user before the scheduling of a frame. The output of the scheduler is a_i, the number of allocated PDUs for each user.

We have several requirements for the scheduling. An obvious one is that $\sum_{i \in \mathcal{U}} a_i \leq C$. We also require that $r_i \geq a_i$ so that user is never allocated more than its request. This has the consequence that if $\sum_{i \in \mathcal{U}} r_i < C$, some of the capacity will not be allocated.

During periods of overload (overload is defined by $\sum_{i \in \mathcal{U}} r_i \geq C$) we have to define how to share the capacity. For this, we define a weight w_i for each user, so that we can share the capacity in proportion to the weights. It may not be possible however to achieve this in every frame since exactly. Therfore we introduce a state variable for each user into the scheduling called *normalized lead*, denoted by $\tilde{l}_i$. The normalized lead is initially zero for all users, and is interpreted as the amount of normalized allocation a user has received in addition to its fair share.

Figure 2 presents the basic allocation mechanism run for each frame.

The algorithm is iterative: it allocates PDUs of a frame one by one until there is available capacity or every request has been completely satisfied. In each iteration the algorithm chooses from the set of users who have unsatisfied request (denoted by $\mathcal{B}$) the one which has the minimal normalised lead (denoted by u). User u is allocated one unit (PDU), r_u is decreased by one to reflect the amount of requests pending, and the normalised lead is increased by the additional normalised allocation, $1/w_i$.

At the end of the iteration, the normalised lead of all users is decreased by the minimal normalised lead of users with unsatisfied request (but normalised lead is not decreased below zero). This ensures that users which have been satisfied cannot recleaim later the amount of allocation which they have not requested in a frame.

4 Fair allocation on a lossy channel

By assigning weights to users, we can determine the proportion of allocations assigned by the scheduler. In a wireless network however all data is subject to bit errors and therefore losses. We do not assume any knowledge or prediction of the channel ahead of

1 **for** $i \in \mathcal{U}$ /*Initialize*/
2 a_i=0;
3 $\mathcal{B} = \{i | r_i > 0\}$;

4 **while** $\sum_{i \in \mathcal{U}} a_i < C$ **and** $\mathcal{B} \neq 0$
/*While there is available capacity and there is unsatisfied demand*/
5 $u = i$ *for which* $\tilde{l}_i = \min\{\tilde{l}_j | j \in \mathcal{B}\}$;
/*User with minimal normalised lead*/
6 a_u+=1; /*Allocate one unit*/
7 r_u-=1; /*Decrease unsatisfied request*/
8 **if** r_u==0 **then** $\mathcal{B}$=$\mathcal{B} \setminus \{u\}$; /*Update $\mathcal{B}$*/
9 $\tilde{l}_u += 1/w_u$; /*Update normalised lead*/

10 $\tilde{l}^0 = \begin{cases} \min\{\tilde{l}_j | j \in \mathcal{B}\} & \text{if} \quad \mathcal{B} \neq 0; \\ \infty & \text{if} \quad \mathcal{B} = 0; \end{cases}$
/*Minimal normalised lead of users with unsatisfied request*/
11 **for** $i \in \mathcal{U}$
12 $\tilde{l}_i = \max\{\tilde{l}_i - \tilde{l}^0, 0\}$; /*Update normalised lead*/

Figure 2: Bandwidth allocation algorithm

transmission. However, we are interested in how the losses can be compensated in the allocation afterwards. For this, we assume that after transmission we have knowledge whether the transmission of a PDU was successful or not. This is typically possible by the ARQ (automatic repeat request) mechanism employed over the air interface.

We modify the amount of normalised service given to a user based on the losses on the channel. When a PDU is allocated to a user and is transmitted successfully, the amount of normalised service given to the user is $s_i = 1/w_i$. When the transmission fails, the normalised service is given by $s_i = (1 - \beta)/w_i$ where β is called the compensation constant. The value of β must satisfy $\beta \leq 1$.

A value of $\beta = 0$ means no compensation for losses; values $0 < \beta < 1$ correspond to partial compensation whereas $\beta = 1$ corresponds to full compensation, which follows that the scheduler provides fairness in the number of successful transmissions. The consequence of compensation is that a user with bad channel is allocated more capacity, thereby increasing the fairness of goodput values. It is also possible to apply negative compensation, in which case users with a bad channel are allocated less capacity.

We are currently doing simulation studies to evaluate the proposed bandwidth allocation and loss compensation mechanism and to find out the relationship between the system utilisation, application-level and DLC-level fairness of bandwidth allocation depending on the type of scheduler, the applied ARQ mechanism and the impact of TCP transport. Here we summarise our results so far.

We have made simulations using one base station of 1.5Mbps capacity. We set up 8 connections corresponding to 8 MTs each downloading data through a greedy New-Reno TCP connection. We have set the weights of the connections to 1;1;2;2;4;4;8;8; respectively. We have used different loss models for the odd-numbered and even-numbered connections. The loss rate was set to 5% or 25%.

Figure 3 illustrates the compensation mechanism. Both show the throughput values for the eight connections normalised by the weight. The throughput values are shown

for the allocated capacity, and at the DLC and IP level and at application level (i.e. goodput), respectively. For the figure on the left, we used 5% i.i.d. loss for odd-numbered and 25% i.i.d. loss for even-numbered connections. The figure illustrates that compensating for the lost allocations, the allocation can be made fair in terms of goodput allocations. For the figure on the right, we also used 5% i.i.d. loss for odd-numbered but 25% bursty loss for even-numbered connections. For this figure, the ARQ employed at the DLC layer was semi-reliable: it gave up after 8 transmission attempts. This figure illustrates that for bursty losses and semi-reliable ARQ, the fairness properties are determined by the TCP congestion control mechanisms and are not improved by the compensation mechanism.

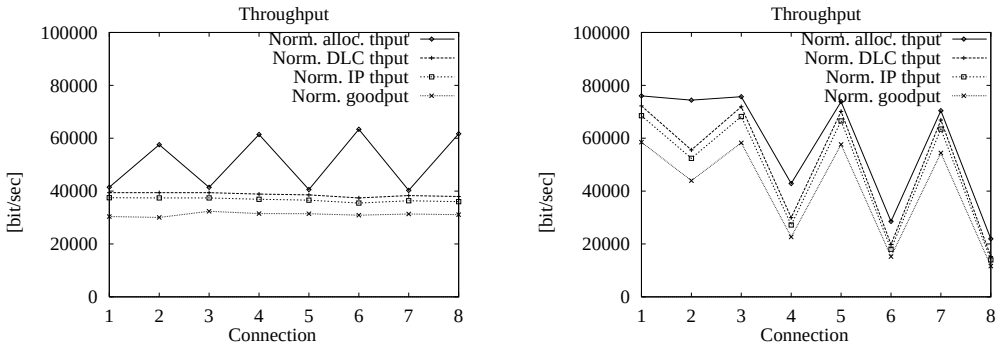

Figure 3: Illustration of compensation mechanism

Further simulation results, not detailed in this abstract, show that fairness of goodput values can be considerably improved by positive compensation at the expense of reduced system utilisation. The reduction in system utilisation is negligible in case of i.i.d. loss models, but can grow to considerable reduction (e.g. 10%) in the case of bursty loss models. Higher utilisation can be achieved by applying negative compensation. Semi-reliable ARQ can also result in slightly improved utilisation, however the fairness properties in this case can be much worse and can not be significantly improved by the compensation mechanism.

5 Conclusion

We have presented a simple fair allocation mechanism and a simple extension for compensation of the losses on the channel. We have made initial simulations to evaluate the proposed mechanisms. The results show that in the case of independent losses on the channel the compensation can significantly improve the fairness of goodput allocations at a minimal reduction of the utilisation. However at bursty losses the reduction of utilisation can be significant. We can apply a semi-reliable ARQ at the link layer to increase the utilisation, but in this case the compensation mechanism is not effective: the fairness of goodput values is determined by the TCP congestion control mechanism.

We plan to extend our simulation studies in the future with other types of scheduling and compensation schemes and present the results in a subsequent paper.

References

[1] Jon C. R. Bennett and Hui Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689.

[2] ETSI Broadband Radio Access Networks project. http://www.etsi.fr/bran/.

[3] Pawan Goyal, Harrick M. Vin, and Haichen Cheng. Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Transactions on Networking*, 5(5):690–704, October 1997.

[4] Songwu Lu, Vaduvur Bharghavan, and Rayadurgam Srikant. Fair shceduling in wireless packet networks. In *SIGCOMM'97*.

[5] T. S. Eugene Ng, Ion Stoica, and Hui Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. In *INFOCOM'98*.

[6] Hui Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10), October 1995.