# FAIR ALLOCATION OF ELASTIC TRAFFIC FOR A WIRELESS BASE STATION

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

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

**Abstract**
Increasingly deployed cellular wireless systems can often become the bottleneck of the network so the fair allocations of the capacity of a wireless base station becomes important. We present a simple allocation mechanism which distributes the capacity of a base station in a fair manner, and show how it can simply be extended to compensate for the losses on the air interface. Using partial or full compensation, it is possible to be fair not in allocations, but in user goodput. Through simulations we analyse the impact of i.i.d. and bursty losses and the effects of fully and semi-reliable ARQ protocols at the link layer.

## I. Introduction

Cellular wireless systems are becoming increasingly common as access systems with the development of several new technologies (e.g. GPRS [3], UMTS [12], Wireless LAN systems [7]). With the deployment and growing use of these access technologies, the capacity of a radio cell can frequently become the bottleneck point of the network. The efficient sharing of this capacity is therefore vital.

Fair sharing of the bandwidth of a wired link has been an important area of research [2, 5, 13]. 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 can change due to mobility. 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) and it may have other MAC-specific constraints.

Recent work [9–11] on wireless fair scheduling is based on the assumption that errors are typically bursty in nature, such that the scheduler can predict the state of the channel for each user. First, the error-free service is defined assuming no channel errors, which is extended by a lead and lag model that keeps track of the differences between the real and the ideal service. A compensation model defines how to compensate lagging flows at the expense of leading flows when their channel is error-free.

In this paper we do not assume any prediction about the channel since in many cases we have knowledge about the success of a transmission only after the transmission. We study how fairness can be improved using a simple extension to the scheduler based on this information.

The focus of this paper is the fair allocation of the capacity of a wireless base station. The paper 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. Specifically, we use the most widely-deployed elastic transport protocol: TCP. After describing the architecture (section II) we present a basic allocation mechanism (section III) and address the question of how losses can be accomodated in the fairness model (section IV). The proposed mechanisms are analysed through simulations (section V).

## II. Architecture

Figure 1 illustrates the architecture considered in this paper. An example to the architecture is the HiperLAN wireless LAN system [7], but the architecture itself is general.

We call each transmission/reception entity a user. 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.

The system considered in this paper uses time division multiple access scheme. The MAC layer is frame-based using fixed frame length. The capacity of a frame can dynamically be distributed by the master-scheduler for both uplink and downlink transmissions (time division duplex system). The unit of allocation is one radio PDU which is of fixed length.

The master scheduler makes its resource allocation decisions based on the resource requests from the users. Due to the centralised 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

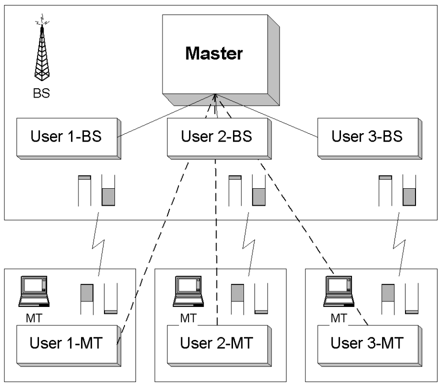



Fig. 1. System architecture

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 capacity is allocated to each user in the next frame in the unit of a radio PDU. In this paper 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.

## III. 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 paper. The basic allocation mechanism can be regarded as the adaptation of start-time fair queuing [6].

The MAC frame 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 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

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

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

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

Fig. 2. Bandwidth allocation algorithm

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 service*, $1/w_i$.

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 reclaim later the amount of allocation which they have not requested.

## IV. 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 transmission. We are interested in how the losses can be compensated in the allocation afterwards so that long-term fairness can be ensured. For this, we assume that after transmission we have knowledge whether the transmission of a PDU was successful or not. This information is provided by the ARQ (automatic repeat request) mechanism employed over the air interface.

The basic allocation mechanism described above provides fair allocation of the capacity, but due to the losses over the air interface the actual goodput of the users, defined by the carried user data payload over unit time, will not be fair according to the weights.

To take into account the losses over the air interface we modify the allocation mechanism in the following way. When a PDU is allocated to a user and is transmitted successfully, the amount of normalised service given to the user is $s_i = 1/w_i$, as previously. When the transmission fails, the normalised service is given by $s_i = (1-\beta)/w_i$ where $\beta$ is called the *compensation constant* satisfying $\beta \leq 1$. The value $\beta = 0$ 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. Using a compensation constant $0 < \beta \leq 1$ we can increase the fairness of the actual goodputs seen by the end users, as opposed to the fairness of allocations ($\beta = 0$), which is in fact not directly observed by the end users.

It is also possible to have negative compensation ($\beta < 0$), in which case PDU losses are explicitly punished by the scheduler. In this case losses for one user cause the scheduler to allocate less capacity for that user in the future. As a result, we can potentially increase the system utilization by allocating less resources to users with worse channels.

It is well known [1] that TCP performance over wireless links can be very low due to the fact that packet losses over the air interface trigger the TCP congestion control mechanisms. We address this problem by assuming an ARQ protocol running at the link layer which retransmits lost PDUs. We use simulations to assess the proposed compensation mechanism working together with the link layer ARQ protocol.

## V. Simulation results

We have implemented the following scenario in our packet-based simulator PlasmaSIM[8]: the bandwidth of a single base station is shared by 8 MTs numbered 1 through 8 whose weights are 1;1;2;2;4;4;8;8; (in fact, to a single MT two users correspond — one in the MT and one in the BS — that have equal weights). Each MT is receiving data through a single greedy TCP connection from a fixed host (greedy meaning a connection in which there is always data to send). Our TCP version is New-Reno [4], which can recover from losses more efficiently than the Reno variant without reaching timeout. The wireless link is the only bottleneck in our setup. There is a 10ms delay between the fixed host and the base station in both directions. For a summary of other simulation parameters, refer to table I.

In this study we use three loss models to represent losses due to bit errors. The first model $\mathcal{M}_1$ has 5% independent (i.i.d.) loss probability for all PDUs; the second model $\mathcal{M}_2$

| | |
|---|---|
| MAC frame length | 2ms |
| # PDU per frame | 8 |
| PDU length | 48 bytes |
| System capacity | 1.5 Mbit/sec |
| # MTs | 8 |
| Weights of MTs | 1;1;2;2;4;4;8;8; |
| end-to-end RTT | 20ms |
| Length of simulations | 20sec |
| IP packet size | 552 bytes |
| TCP/IP header | 40 bytes |
| Buffer size at sender | 100 PDU |
| Buffer size at receiver | 100 PDU |
| Loss model $\mathcal{M}_1$ | 5% i.i.d. loss |
| Loss model $\mathcal{M}_2$ | 25% i.i.d. loss |
| Loss model $\mathcal{M}_3$ | 25% bursty loss, MMBP |
| Mean hold. times, $\mathcal{M}_3$ | (5,17.5) [PDU transmit time] |
| Loss rate, MMBP of $\mathcal{M}_3$ | (0.95, 0.05) |

TABLE I
SIMULATION CONSTANTS

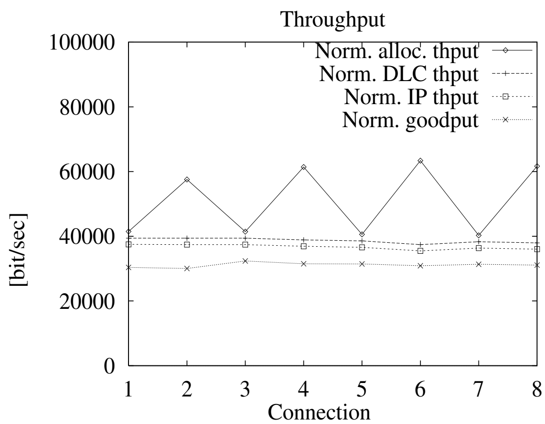
has 25% i.i.d. loss; while the third model $\mathcal{M}_3$ has 25% bursty loss as given by a Markov Modulated Bernoully Process (MMBP): the loss probability is modulated by a two-state discrete-time Markov-chain. See table I for the parameters of the model. We use the same loss model in both uplink and downlink directions in all cases.

The ARQ protocol operating at the link layer is a simple selective repeat mechanism. The protocol incorporates a discarding feature as well which allows the sender to give up further retransmission attempts after a certain number of trials. The knowledge of the number of lost PDUs in a frame are transferred implicitly by the ARQ protocol to the master scheduler in the base station so that it can perform compensation one frame later than the transmission errors actually occur.

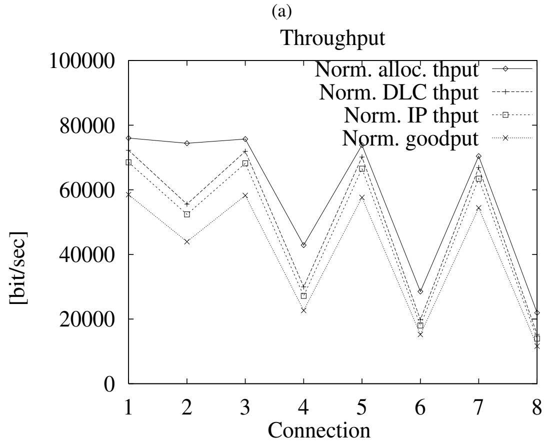
Our performance metrics are based on the throughput values of the connections (where a connection corresponds to a MT). We define four levels of throughput.

The *allocated throughput* is the total transmission capacity allocated to a connection, i.e. the number of allocated PDUs per second times the bit length of a PDU. The *DLC-throughput* is the throughput above the ARQ protocol, i.e. the number of successfully transmitted PDUs per second times the bit length of a PDU. The DLC-throughput is less than the allocation throughput due to losses over the air interface. The *IP-throughput* is the throughput above the DLC layer, i.e. the number of successfully transmitted IP packets per second times the bit length of an IP packet. The IP-throughput is less than the DLC-throughput due to padding at the last PDU of a packet in segmentation and some imperfections of the discarding mechanism. The *application-level throughput*, or *goodput*, is the throughput seen by applications, i.e. the number of bits transferred from the end-sender to the end-receiver per second. The goodput is smaller than IP-throughput because of the IP and TCP headers in each packet, the presence of acknowledgement packets in the reverse direction, and due to the imperfections of TCP which allow unnecessary retransmissions.

We first illustrate the results of full compensation. In our first scenario, we use model $\mathcal{M}_1$ (5% i.i.d. loss) for connections 1,3,5,7; and use model $\mathcal{M}_2$ (25% i.i.d. loss) for connections 2,4,6,8; and fully-reliable ARQ at the link layer. Figure 3 shows the throughput values for each connection normalised by the weights of connections (1,1,2,2,4,4,8,8 for connections 1 through 8 respectively). Figure 3(a) illustrates full compensation ($\beta = 1$): by increasing the allocation for MTs with worse channels, the normalised DLC and IP-level throughputs are nearly identical, resulting in fair goodput allocation. Figure 3(b) will be considered later.




(a)




(b)

Fig. 3. Throughput measures in case of (a) $\mathcal{M}_1/\mathcal{M}_2$ models, fully reliable ARQ, full compensation (b) $\mathcal{M}_1/\mathcal{M}_3$ models, semi-reliable ARQ, no compensation. (The values for each connection are connected only for easier presentation.)

In order to achieve fairness in the goodput, as Figure 3(a) illustrates, we have to allocate more capacity to users with worse channel conditions. This implies a tradeoff between the fairness of goodput seen by the users and the utilization of the system.

To analyse this tradeoff, we define the following metrics. *utilization* is defined as the sum of the throughput of the connections divided by the system capacity, and is interpreted for each of the four levels of throughput. *Fairness* is characterised by the variability of the normalised throughputs of connections. Here we measure fairness by the coefficient of variation (standard deviation divided by the mean) of the normalised throughput values, and it is also interpreted for each of the 4 levels of throughput. (The smaller the fairness measure is, the more fair the allocation is.)

We first consider the same scenario with loss model $\mathcal{M}_1$ (5% i.i.d. loss) for odd-numbered and loss model $\mathcal{M}_2$ (25% i.i.d. loss) for even-numbered connections. Figure 4 shows the utilization and fairness measures as a function of the compensation constant.

The results show that it is possible to improve fairness of IP-level throughput and goodput by more then an order of magnitude. The goodput fairness improves monotonically with increasing compensation constant as a result of the compensation mechanism. This comes at the expense of a slight decrease in the utilization: goodput utilization drops by 1.8% as $\beta$ increases from 0 to 1. By applying a negative compensation of $\beta = -2$, we can increase utilization slightly by 1.2%.

The second scenario we consider is the same loss model $\mathcal{M}_1$ (5% i.i.d. loss) for odd-numbered connections but loss model $\mathcal{M}_3$ (25% bursty loss) for even-numbered connections. Figure 5 shows the results. We can observe a similar improvement of the fairness of goodput values, but this comes at the expense of significantly deteriorated utilization. (10.3% decrease in the goodput utilization as $\beta$ goes from 0 to 1.) By applying negative compensation, we can improve goodput utilization by 4.8% at the expense of reduced fairness.

A fully reliable ARQ protocol in the link layer continues retransmissions indefinitely which may result in frequent losses in the case of bursty losses. We investigated the effect of limiting the number of retransmission attempts. The results for the limit of maximum eight transmission attempts of a given PDU are shown in Figure 6.

The results show that using a semi-reliable ARQ the utilization can be slightly improved (for the $\beta = 0$ case it improved by 3.8%.) However, the allocation in this case becomes unfair and the fairness cannot be improved by increasing the compensation constant. Fairness in the case of semi-reliable ARQ protocol is determined by TCP's response to the losses not recovered by the ARQ protocol. It is interesting to note that fairness actually worsens in this case by increasing the compensation constant, which can be explained by the fact that additional capacity is allocated to a user following a loss, which in the case of bursty errors ac-

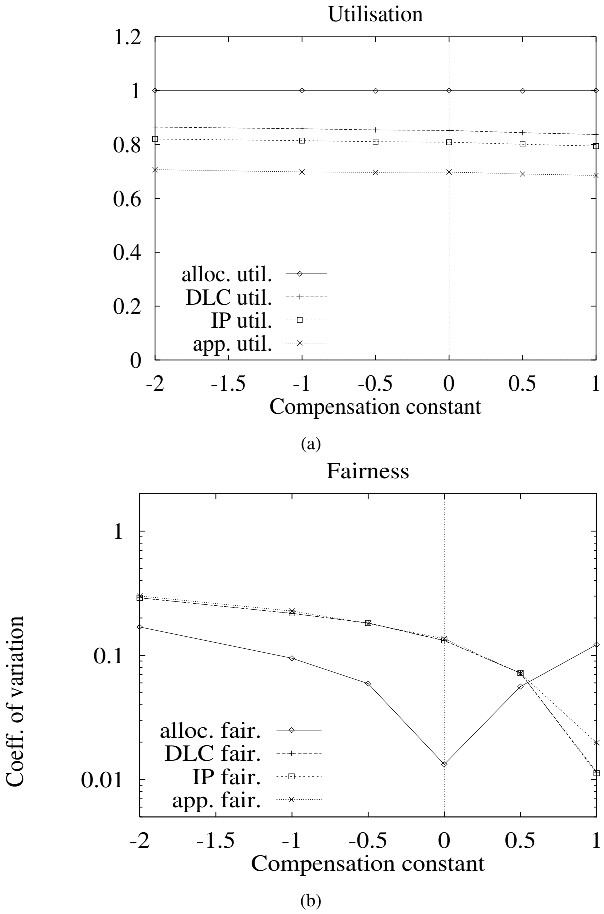



Fig. 4. utilization and fairness in the case of 25% i.i.d. loss for even-numbered connections.

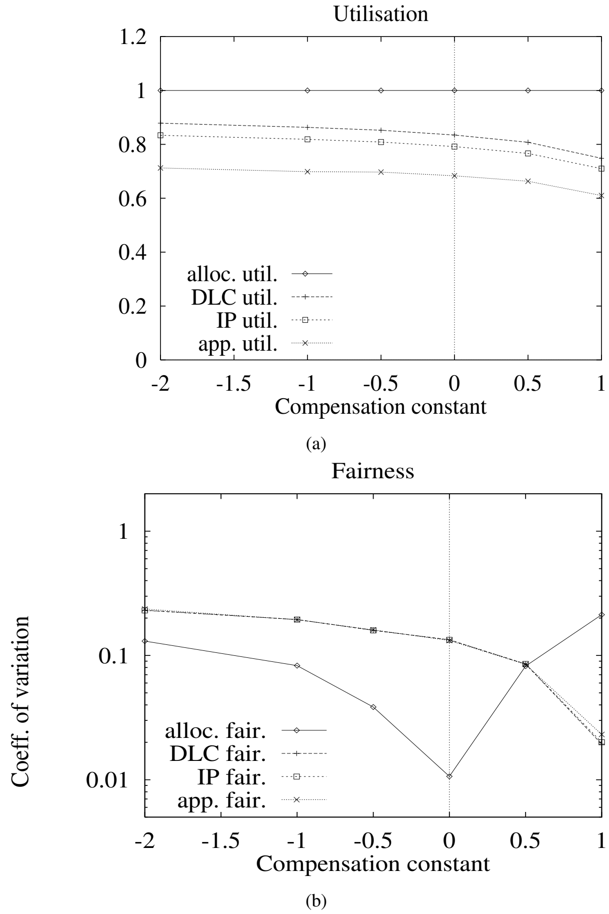



Fig. 5. Utilization and fairness in the case of 25% bursty loss for even-numbered connections, fully reliable ARQ.

tually increases the probability of repeated losses and therefore IP packet drops.

Figure 3(b) shows the per connection throughput values ($\beta = 0$), implying that the relative decrease in the throughput of even-numbered connections ($\mathcal{M}_3$ loss model) is much higher for connections with higher weight, due to the fact that the throughput in this case is limited by the TCP congestion control mechanism.

## VI. Conclusion

We have presented a novel scheme for fair allocation and loss compensation for the master scheduler of a wireless base station. The performance evaluation of the proposed scheme has been carried out based on simulations that show the impact of TCP traffic and link-layer ARQ.

The results show that fairness of goodput values can be considerably improved by positive compensation at the expense of reduced system utilization. The reduction of utilization is slight in the case of independent channel losses, and increases with the burstiness of the loss model. Higher utilization is achieved by applying negative compensation or by using a semi-reliable ARQ at the link layer. The fairness of allocation can become considerably worse and can not be improved by the compensation mechanism in case of using semi-reliable ARQ.

## References

[1] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5:756–769, Dec 1997.

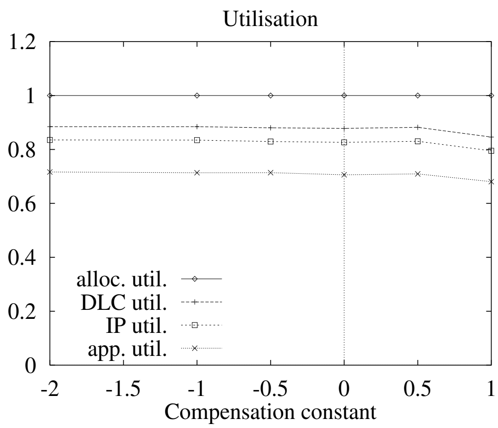
[2] Jon C. R. Bennett and Hui Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689.

[3] Jian Cai and David J. Goodman. General packet radio service in GSM. *IEEE Communications Magazine*, October 1997.

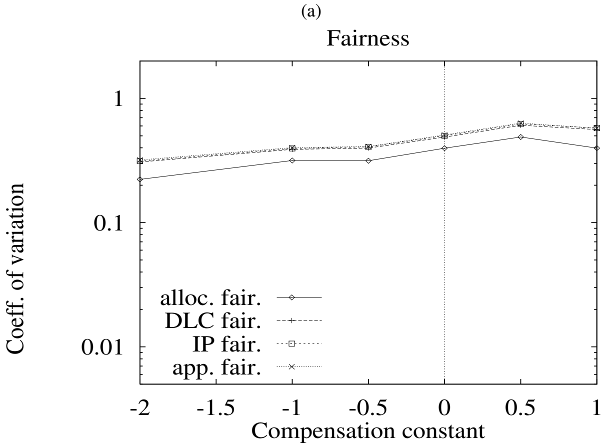
[4] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. *Computer Communications Review*, 26(3):5–21, July 1996.

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

[6] Pawan Goyal, Harrick M. Vin, and Haichen Cheng. Start-time fair queuing: A scheduling algorithm for integrated services packet




(a)




(b)

Fig. 6. Utilization and fairness in the case of 25% bursty loss for even-numbered connections, semi-reliable ARQ.

switching networks. *IEEE/ACM Transactions on Networking*, 5(5):690–704, October 1997.

[7] HiperLAN/2 Global Forum. http://www.hiperlan2.com/.

[8] Zs. Haraszti, I. Dahlquist, A. Faragó, and T. Henk. PLASMA - an integrated tool for ATM network operation. In *International Switching Symposium ISS '95*, Berlin, Germany, April 1995.

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

[10] Songwu Lu, Thyagarajan Nanadagopal, and Vaduvur Bharghavan. A wireless fair service algorithm for packet cellular networks. In *ACM MOBICOM'99*, August 1999.

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

[12] Tero Ojanpera and Ramjee Prasad. *Wideband CDMA for Third Generation Mobile Communications*. Artech House Publisher, 1998.

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