

TCP fairness measures for scheduling algorithms in wireless networks

Krister Norlund
Ericsson AB
Lindholmspiren 11
417 56 Göteborg, Sweden
krister.norlund@ericsson.com

Tony Ottosson
Dept. of Signals and Systems
Chalmers Univ. of Technology
412 96 Göteborg, Sweden
tony.ottosson@s2.chalmers.se

Anna Brunstrom
Dept. of Computer Science
Karlstad Univ.
651 88 Karlstad, Sweden
anna.brunstrom@kau.se

Abstract

This paper presents two new layer 4 fairness measures, the worst case TCP fairness index and the TCP fairness index. The purpose of the two indices is to measure the performance of algorithms scheduling elastic traffic in wireless networks. Numerical evaluation of round-robin, proportional-fair and max-rate scheduling shows that the new measures can accurately classify the fairness properties of the studied scheduling algorithms. This means that the new fairness measures are appropriate for measuring TCP fairness. The worst case TCP fairness index is the primary measure. The TCP fairness index can be used as a complement in certain cases.

The new fairness measures are also compared with two layer 2 fairness measures. Numerical evaluation shows that the new layer 4 measures and the layer 2 measures in some aspects show similar results. However, when evaluating transport layer fairness, it is better to use the new layer 4 measures as they actually measure fairness on layer 4 and the results are easier to interpret.

1 Introduction

Future wireless networks are expected to carry both real-time traffic such as voice and elastic traffic such as web content. Therefore, future networks need to offer two types of services: a guaranteed service for the real-time traffic sources and a best-effort service for the elastic traffic sources. For the best-effort service it is important to find packet-scheduling algorithms that both achieve high total throughput and fairness between users. From a wireless network operator point of view, maximizing throughput can increase revenues. If, for example, the operator would charge users a flat rate, an increase in network throughput can be used to admit more users into the wireless network. Individual users would get the same throughput as before and the operator would get an increase in revenues. Fairness

between users is a requirement from the users of the network. If the scheduling algorithms in a wireless network do not achieve some level of fairness, individual users will in some cases receive arbitrarily low throughput. Users typically do not accept too low throughput and this might cause the users to choose another wireless network operator. This means that the lack of fairness might decrease the revenues for the operator.

In the recent years, a number of new scheduling algorithms intended to schedule elastic traffic in wireless networks have been proposed in the literature [9, 12]. All of the new algorithms exploit multiuser diversity. Performance is typically evaluated considering layer 2 throughput and layer 2 fairness. In most cases, performance is evaluated under the assumption that the queues in the system are always backlogged. This ignores transport layer and application characteristics and their interaction with layer 2 scheduling.

As a majority of the applications that generate elastic traffic uses the transmission control protocol (TCP) as transport protocol, it is important to find and evaluate scheduling algorithms that exploit multiuser diversity on fast fading channels and are well suited for TCP. If we want to take TCP characteristics and its interaction with layer 2 scheduling into account, the performance of scheduling algorithms should be evaluated considering layer 4 throughput and layer 4 fairness. The interesting throughput for an application using TCP, is the throughput received by the application on the receiving side. This throughput is, in this paper, denoted goodput. Throughput is in this paper used to denote the throughput on layer 2. Note that it is not only the scheduling algorithm that will affect the goodput. For example, the link-layer automatic repeat request (ARQ) scheme and the queue policies in the network (especially the queue situated before the bottleneck link) will affect the goodput. This paper, however, focuses on the scheduling algorithms.

The remainder of this paper is organized as follows. Section 2 defines fairness in wireless networks. The problems with using previous defined TCP fairness measures in cel-

ular networks are described in Section 3. In Section 4, two new layer 4 fairness measures are defined. Two previously defined layer 2 fairness measures are described in Section 5. Section 6 describes three scheduling algorithms and the simulation model. Using numerical evaluation the new layer 4 measures are evaluated in Section 7. The scheduling algorithms and the layer 2 fairness measures are used in the evaluation of the new layer 4 fairness measures. Finally, discussion and conclusions end the paper in Section 8.

2 Definitions of fairness

In the context of networks, fair typically means equal sharing of the resources. A scheduling algorithm that always shares the resources equally between users, is in this paper denoted *resource-fair*. For a single link in a wire-line network, equal sharing of the resources implies that all users should receive an equal throughput. A scheduling algorithm that always gives all users an equal throughput is in this paper denoted *throughput-fair*.

For a cellular network equal sharing of the resources does not necessary imply that all users will receive an equal throughput. If the average channel conditions differ between users, giving an equal share of the resources to the users will in most cases imply that a user with a good average channel condition receives a higher throughput than a user with a poor average channel condition. So, what is a fair throughput in a wireless network? Our proposed definition of fair scheduling in a wireless network is that, a user has received a fair share if the user receives a throughput that is equal to or greater than what the user would receive if a resource-fair algorithm was used as the scheduling algorithm. This definition of fairness is in this paper denoted *relative-throughput fairness*.

The resource-fair algorithm can for example be work conserving round-robin or the ideal but unimplementable generalized processor sharing algorithm [7, chapter 9.4.1]. Work conserving round-robin works as a normal round-robin scheduling algorithm, with the addition that empty queues do not force the scheduler to wait for an incoming packet for that user, instead the queue for the next user is polled. In the remainder of this paper round-robin denotes work conserving round-robin. As round-robin is easier to implement than generalized processor sharing, we choose round-robin as the reference algorithm.

We define *relative-goodput fairness* in wireless networks in the same way as throughput fairness in wireless networks. A user has received a fair share if the user receives a goodput that is equal to or greater than what the user would receive if a resource-fair algorithm was used as the scheduling algorithm. We choose round-robin as the resource-fair reference algorithm. When evaluating the goodput achieved for the scheduling algorithm under eval-

uation and the resource-fair algorithm, the system and system parameters must be the same in both evaluations. For example, the TCP implementation must be the same.

Note that, if a scheduling algorithm exploits multiuser diversity, it is possible that all users will receive a goodput that is higher than the goodput they would receive using the resource-fair scheduling algorithm. This means that a scheduling algorithm that exploits multiuser diversity can be relative-goodput fair but at the same time not resource-fair.

3 Problem with previous TCP fairness measures

The interaction between TCP and layer 2 scheduling in wireless networks has in some aspects been investigated in previous papers. In [6], the authors propose a combined ARQ and scheduling scheme that maximizes performance from the stand point of end-to-end applications. The end-to-end applications either use TCP or UDP as transport protocol. Fairness between users (in terms of resource-fair, throughput-fair, relative-throughput fair or relative-goodput fair), is unfortunately not evaluated in the paper.

In [1], the authors evaluate the throughput and fairness of two scheduling algorithms scheduling TCP traffic. The fairness index defined in [5, page 36], here denoted Jain's fairness index, is used in the paper to measure throughput-fairness. For any given set of throughputs (y_1, y_2, \dots, y_n) , Jain's fairness index is calculated as:

$$g(y_1, y_2, \dots, y_n) = \frac{\left(\sum_{i=1}^n y_i\right)^2}{n \cdot \sum_{i=1}^n y_i^2}$$

Jain's fairness index always lies between 0 and 1. A fairness index of 1 indicates a throughput-fair algorithm. As Jain's fairness index measures throughput-fairness it is not appropriate to use it when relative-goodput fair is the definition of a fair scheduling algorithm.

Example 1: Consider a time-slotted wireless network with two users, user A and user B. Relative-goodput fair is the definition of a fair scheduling algorithm. Say that user A has a constant feasible bit rate of 2 Mbit/s and user B has a constant feasible bit rate of 4 Mbit/s. Feasible bit rate is in this paper defined as the highest possible transmission rate (bit/s), achieving the desired quality of service. If round-robin is used as scheduler in the wireless network, user A would receive a goodput of 1 Mbit/s and user B would receive a goodput of 2 Mbit/s. Jain's fairness index would in this case be $(1 + 2)^2 / 2 \cdot (1^2 + 2^2) = 9/10 \neq 1$. This means that Jain's fairness index indicates unfairness for a scheduler that is relative-goodput fair.

In [8], the authors propose a scheduling algorithm that is explicitly tuned towards goodput performance. The scheduling algorithm is an extension of the proportional-fair scheduling algorithm [13]. It is known that proportional-fair maximizes the function:

$$\sum_{i=1}^n \log(y_i) \quad (1)$$

As the algorithm in [8] is an extension of proportional-fair, the authors choose to measure layer 4 fairness as:

$$d = \frac{1}{n} \cdot \sum_{i=1}^n \log(v_i) \quad (2)$$

where v_i is the goodput for user i in kbyte/s. The measure d is here denoted the proportional-goodput fair measure. The higher value of d , the more fair a scheduling algorithm is. Using (2) as fairness measure implies that the definition of a fair scheduling algorithm is an algorithm that maximizes (2). An algorithm that maximizes (2) is in this paper denoted *proportional-goodput fair*. In [8], the authors do not give an intuitive explanation of why it is an good idea to use proportional-goodput fair as a definition of a fair algorithm. However, it has been shown in previous papers that the proportional-fair scheduling algorithm has some nice fairness properties. In [2], it was shown that given that the distance to the basestation is the same for all users and that the variability of the feasible bit rate differs between users, the users with greater variability get a higher throughput and use a lower (but not much lower) fraction of time.

In [8], both proportional-throughput fairness and total goodput are measured for the scheduling algorithms. The proportional-throughput fair measure is a function of the goodputs for the different users and the total goodput achieved. As the proportional-throughput fair measure depend on the total goodput achieved, round-robin will be considered less fair than an algorithm that maximizes (2).

In the literature, fairness of a scheduling algorithm traditionally does not depend on the total throughput achieved. For example, the relative fairness bound [7, chapter 9.4.1], absolute fairness bound [7, chapter 9.4.1] and the fairness measures described in Section 5 do not depend on the total throughput. It can be questioned if it is a good idea that fairness of a scheduling algorithm depends on the total throughput. We prefer to have one separate measure for total throughput and another separate measure that only measures layer 4 fairness. The measure of total goodput is meant to indicate the resource utilization of the system and the fairness measure is meant to indicate how the resources have been shared among the users. As a definition of fairness for wireless networks, we prefer relative-goodput fair compared with proportional-goodput fair. The reason is that we believe it is easier to intuitively understand the meaning of relative-goodput fair.

4 Two new TCP fairness measures

We propose two new measures for measuring the level of relative-goodput fairness for a scheduling algorithm. The two measures are denoted the worst case TCP fairness index and the TCP fairness index.

Given a scheduling algorithm to evaluate, let (v_1, v_2, \dots, v_n) be the goodputs received by the users in the network. Let (u_1, u_2, \dots, u_n) be the goodputs received by the users, given that round-robin is used as scheduler. We define the worst case TCP fairness index as:

$$q(v_1, v_2, \dots, v_n, u_1, u_2, \dots, u_n) = \min_{\forall i} [M(v_i/u_i)]$$

where M is a positive real-valued function defined as:

$$M(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

The worst case TCP fairness index is the goodput for the worst case user divided by the goodput the user should have received. The worst case user is the user that suffers the worst goodput fairness calculated in percentage. Given a numerical result, it is easy to comprehend the implication of the result. If the worst case TCP fairness index is equal to 1, all users are treated perfectly fair (relative-goodput fair).

An algorithm should try to give each user a relative-goodput fair goodput. If all users receive a fair goodput, we consider it to be reasonable that any remaining resources should be used to maximize the total goodput. The purpose of the M function is to make sure that once all users receive a goodput equal to the goodput received with round-robin, an algorithm can focus on using the remaining resources to maximize the total throughput.

One disadvantage with the worst case TCP fairness index is that it only considers the worst case user. All the other users might be treated perfectly fair or as bad as the worst case user. We therefore define a new measure, called the TCP fairness index, that complements the worst case TCP fairness index. The TCP fairness index measures the aggregated unfairness of all users and it is defined as:

$$h(v_1, v_2, \dots, v_n, u_1, u_2, \dots, u_n) = \frac{\left(\sum_{i=1}^n M(v_i/u_i) \right)^2}{n \cdot \sum_{i=1}^n M(v_i/u_i)^2} \quad (3)$$

The value of the TCP fairness index always lies between 0 and 1. If the worst case TCP fairness index is equal to 1, the TCP fairness index is also equal to 1. Note that the worst case TCP fairness index should primary be used. If the worst case TCP fairness index is less than 1, the TCP fairness index can be used to get more information on how relative-goodput fair the scheduling algorithm is. The purpose of the M function in equation (3) is to make sure that

multiuser diversity can be fully exploited without necessarily decreasing fairness.

Example 2: Consider a time-slotted wireless network with two users, user A and user B. Say that user A has a constant feasible bit rate of 4 Mbit/s and user B has a fast fading channel with average feasible bit rate of 4 Mbit/s. In this example, a resource-fair scheduling algorithm that exploits multiuser diversity is used in the wireless network. With this scheduling algorithm user A and user B, for example, receive 2 Mbit/s and 3 Mbit/s. With round-robin both user A and user B receive 2 Mbit/s. If the M function is not used in equation (3), the TCP fairness index would be equal to: $(3/2 + 2/2)^2 / (2 \cdot ((3/2)^2 + (2/2)^2)) = 6.25/6.5 \neq 1$. This indicates unfairness for a scheduling algorithm that is relative-goodput fair. If the M function is used, the TCP fairness index would be equal to 1, indicating a relative-goodput fair scheduling algorithm.

When measuring fairness it is important to measure fairness over the time interval relevant to the user. For an application or (protocol) downloading a file using TCP, the relevant time interval is the time it takes to download the file. If several users simultaneously download a file, download times can be different for different users. The relevant time interval for measuring fairness for a user is still the time it takes to download the file for that user. However, when comparing fairness of several simultaneous TCP flows, it is important to have all but the studied parameters constant. Measuring fairness over a fixed time interval for all TCP flows can solve this problem. The different users might receive a different amount of data during that time interval and before the measurement this amount of data is not known.

5 Layer 2 fairness measures

As we want to take TCP characteristics and its interaction with layer 2 scheduling into account when measuring the performance of scheduling algorithms, fairness should be measured on layer 4. As fairness previously in most cases has been measured on layer 2 it is interesting to find the mapping between layer 2 and layer 4 fairness. To get some idea of this mapping, the new layer 4 measures are compared with two layer 2 fairness measures using numerical evaluation.

Several different layer 2 fairness measures have been defined in the literature. There are a number of reasons why we choose the two measures described in this section. They both give relevant information on the fairness on layer 2. The measures complement each other, one measures relative-throughput fairness, while the other measures resource-fairness. The measures are statistical measures, which means that they consider the fading characteristics of the channel. Both measures are appropriate for measuring fairness of scheduling algorithms that exploit multiuser

diversity. Finally, the relevant time interval size to measure fairness over does not have to be known as the measures consider both short-term and long-term fairness.

The first layer 2 measure is called the 99-percentile wireless absolute fairness bound $WAFB_{99}$ and it is defined in [10]. $WAFB_{99}$ measures the level of relative-throughput fairness. More specifically, it measures the upper bound of unfairness (in bits) of a scheduling algorithm. The measure is based on the wireline fairness measure called the absolute fairness bound [7, chapter 9.4.1]. Let $G(i, t_1, t_2)$ be the amount of data transmitted to user i in the time interval $[t_1, t_2]$ given that generalized processor sharing is used as the scheduling algorithm. The system model in Section 6 considers a time-slotted wireless network and therefore t_1 and t_2 are multiples of the time slot interval in the simulations. $S(i, t_1, t_2)$ is the amount of data transmitted to user i in the time interval $[t_1, t_2]$ if the scheduling algorithm under evaluation is used. Let $C(i, t_1, t_2)$ be a link quality weight associated with user i in the time interval $[t_1, t_2]$. $WAFB_{99}$ is defined as:

$$\Pr \left\{ \frac{G(i, t_1, t_2) - S(i, t_1, t_2)}{C(i, t_1, t_2)} \leq WF_{99}(i, t_1, t_2) \right\} = 0.99$$

$$R(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$$WAFB_{99} = \max_{\forall i, t_1, t_2} [R(WF_{99}(i, t_1, t_2))]$$

In [9], the second layer 2 fairness measure is defined. The measure, here denoted statistical time-access fairness index, measures the level of resource-fairness. More specifically, it measures the difference in service time for different users given a scheduling algorithm. The measure is similar to the wireline fairness measure called the relative fairness bound [7, chapter 9.4.1]. Let $\alpha(i, t_1, t_2)$ be the service in time given to user i in the time interval $[t_1, t_2]$. The statistical time-access fairness index, $f(i, j, x, t_1, t_2)$, is defined as:

$$\Pr \{ |\alpha(i, t_1, t_2) - \alpha(j, t_1, t_2)| \geq x \} \leq f(i, j, x, t_1, t_2)$$

6 System model and system setup

6.1 Scheduling algorithms

For the purpose of evaluating the new fairness measures we consider three different scheduling algorithms with relatively well-known fairness properties. If the new layer 4 measures can identify the fairness properties of the scheduling algorithms, the new measures are valid fairness measures. The scheduling algorithms are described below, where μ_i is the feasible bit rate supported for user i at time t and $\bar{\mu}_i$ is the average throughput for user i measured over

a relatively long "sliding-window". In the simulations the sliding-window size is 1 second. In a time-slotted system the feasible bit rate is considered constant over one time slot.

6.1.1 Max-rate

With the max-rate scheduling algorithm, we schedule the user with the highest feasible bit rate at time t . That is, we schedule user:

$$k = \arg \max_i \mu_i(t)$$

As max-rate always schedules the user with the highest feasible bit rate, it is obvious that in most cellular network scenarios the max-rate scheduler is not resource-fair or relative-goodput fair. For example, a user located close to basestation with a very good channel condition might get all resources and a high goodput. A user located far from the basestation with a poor channel condition might not get any resources and a goodput equal to 0 bit/s. A TCP fairness measure must be able to highlight that max-rate is not relative-goodput fair.

6.1.2 Proportional-fair scheduling

With the proportional-fair (PF) scheduling algorithm, we schedule, at time t , the user:

$$k = \arg \max_i \frac{\mu_i(t)}{\tilde{\mu}_i}$$

The PF scheduling algorithm has been proposed for HDR [12]. As the PF scheduling algorithm exploits multiuser diversity and schedules each user approximately an equal amount of time [13], the throughput for each user will in most cases be slightly higher than the throughput using round-robin. A fairness measure should be able to distinguish this scheduling algorithm as being close to relative-goodput fair.

6.1.3 Round-robin

Over an interval where all users are backlogged, round-robin will serve all users an equal amount of time (resource-fair). Note that when TCP is used as transport protocol, all users might not be backlogged all of the time. A TCP fairness measure should indicate that round-robin is relative-goodput fair.

6.2 Simulation setup

For the purpose of evaluating the new fairness measures, we consider a simple cellular system. There are five mobile stations in a single cell. Each mobile station has a client application downloading one single file from a server using TCP Reno with a maximum segment size of 1460 bytes.

The initial TCP three-way handshake is not included in the model. This means that for short transfers the goodput is overestimated. During simulation all TCP senders have data to transfer during the measurement interval. This means that all the TCP flows are competing for the resources during the measurement. Fig. 1 depicts the network components considering one user.

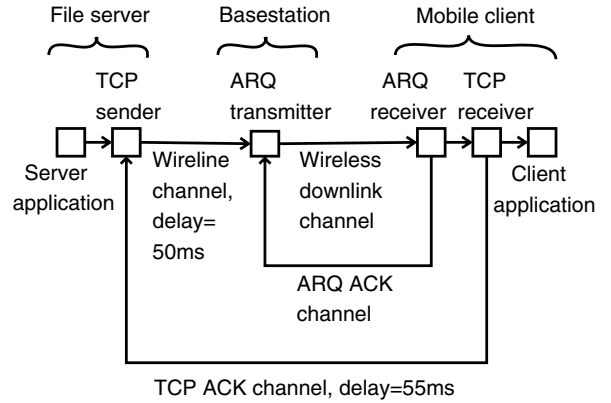


Figure 1. Network components for one user.

The wireless downlink channel is modeled as a single-carrier time-slotted channel. The carrier frequency is 1900 MHz and the bandwidth is 200 kHz. Each time slot is 0.667 ms and contains 104 payload symbols. The wireless downlink channel for a user is modeled as a correlated Rayleigh fading channel using Clarke's model [11, chapter 5.7.1]. The speed for all mobiles is 50 km/h. Fading is considered to be flat within one time slot. The fading coefficients for two different users are uncorrelated. The average signal to interference and noise ratio (SINR) per symbol for a user is constant. Starting with user 1 the average SINR per symbol is equal to 8, 12, 16, 20 and 24 dB. Only one mobile receives data in each time slot. The instantaneous signal to interference and noise ratio for the upcoming time slot is perfectly predicted. Given the instantaneous SINR, the system selects a modulation format for the upcoming time slot. Modulation formats from BPSK up to 256-QAM are used. See [14] for the exact modulation formats and switching levels. For the 99-percentile wireless absolute fairness bound measure, the link quality weights are equal to the feasible bit rate for the given time slot. One or more of the symbols transmitted in a time slot might be received erroneously. The symbol error probability is calculated given the instantaneous SINR and modulation format. As no coding is used, a symbol error causes all data in the time slot to be discarded. Four time slots after the initial transmission, the discarded data can be retransmitted by the ARQ scheme.

The ARQ scheme used is selective repeat. One simplification is made, the ARQ receiver does not send any NACKs to the ARQ transmitter. This means that only the timer in

the ARQ transmitter will trigger retransmission of data. The sequence number size for the ARQ scheme is 16384. The ARQ transmitter contains a queue with the size of 32*1500 bytes and the queue policy is drop front. When a packet needs to be dropped, the first packet in the queue (starting with the next packet to be scheduled) is discarded.

For simplicity the uplink channels for the ARQ ACKs are modeled as an error-free link with a certain delay and infinite bandwidth. To make it simple, the uplink channels for TCP ACKs are modeled in the same way.

7 Numerical evaluation

The performance of the max-rate, proportional fair and round-robin scheduling is evaluated using the performance measures described earlier. The total goodput and the fairness is compared for the different scheduling algorithms. Goodput and fairness is measured over the time interval: 0.8, 8, 80 or 800 seconds. One effect of measuring over both large and relatively small time interval sizes, is that the simulations will catch TCP's transient and steady-state behavior. Given a measurement time interval and a scheduling algorithm, the same simulation (but with different seeds) is executed 1000 times. In the figures the mean values are displayed. As the 99% confidence intervals are small for all plotted results, they are not plotted in the figures. IT++ [4] in combination with IKR Simulation Library [3] was used as simulator. Fairness is measured using the worst case TCP fairness index and the TCP fairness index. In order to verify that the new fairness measures are valid fairness measures, it is verified that they can identify the fairness properties of max-rate, proportional fair and round-robin described in Section 6. To get some idea of the mapping between layer 4 fairness and layer 2 fairness, the results using the new measures is compared with the results using wireless absolute fairness bound and statistical time-access index as fairness measures.

7.1 Numerical goodput results

Fig. 2 and 3 depict the goodput for different users and scheduling algorithms when the measurement interval is 0.8 and 800 seconds. For the max-rate scheduling algorithm in Fig. 2, the goodput for user 1 and 2 is very small compared with user 5. For round-robin and proportional-fair there is also a significant difference in goodput for the different users but the difference between users is less as compared with max-rate.

The aggregated goodput for all users is depicted in Fig. 4. Max-rate has the highest goodput and round-robin the lowest goodput. For all three scheduling algorithms the goodput is significantly lower when the measurement interval is 0.8s compared with 8s. The main reason for this difference

is that it takes a number of round trip times before the TCP congestion window is large enough for TCP to transmit at the steady-state bit rate for the TCP flow.

7.2 Numerical layer 4 fairness results

In Fig. 5 the worst case TCP fairness index is depicted. For proportional-fair and round-robin the index is 1, indicating that proportional-fair and round-robin are relative-goodput fair. For max-rate it is easy to understand that at least one user most probably is dissatisfied with the goodput. For all measurement intervals one of the users receive less than 20% of the goodput the user would receive if round-robin was used.

Max-rate has a worst case TCP fairness index less than 1 and therefore the TCP fairness index is also evaluated. Fig. 6 depicts the TCP fairness index for the different scheduling algorithms. The TCP fairness index for proportional-fair and round-robin is equal to 1. Max-rate on the other hand, has a TCP fairness index between 0.5 and 0.9 in Fig. 6. This indicates that max-rate is not relative-goodput fair. For max-rate, the TCP fairness index decreases with higher measurement intervals. This seems reasonable as the goodput difference between max-rate and round-robin for user 1 and 2 is smaller in Fig. 2 as compared with Fig. 3.

7.3 Numerical layer 2 fairness results

Both the 99-percentile wireless absolute fairness bound and the statistical time-access index are defined to find an upper bound on the layer 2 unfairness regardless of time interval size. Given that the interesting measurement interval is fixed in our simulation, the 99-percentile wireless absolute fairness bound and the statistical time-access index only need to measure over that interval. In order to make a just comparison between the new layer 4 fairness measures and the previous defined layer 2 fairness measures, normalized versions of the wireless absolute fairness bound and the statistical time-access index are used.

For the wireless absolute fairness bound we choose to find the 50% percentile (the expected value), instead of the 99% percentile. Given a certain confidence interval, the 50% percentile requires a lower number of simulations than the 99% percentile. Instead of finding an upper bound on unfairness in bits given any time interval size, we choose to find the expected unfair bit rate given a specific time interval size. This bit rate is here denoted the wireless absolute fairness rate $WAFR(t_1, t_2)$. We define it as:

$$\Pr \left\{ \frac{G(i, t_1, t_2) - S(i, t_1, t_2)}{C(i, t_1, t_2) \cdot (t_2 - t_1)} \leq WR_{50}(i, t_1, t_2) \right\} = 0.50$$

$$WAFR(t_1, t_2) = \max_{\forall i} [R(WR_{50}(i, t_1, t_2))]$$

For the wireless absolute fairness rate a value of 0 indicates a relative-throughput fair scheduling algorithm. For increasing values of WAFR the level of unfairness increases. We therefore use the notation unfair bit rate for the wireless absolute fairness rate.

The wireless absolute fairness rate is depicted in Fig. 7. For proportional-fair and round-robin the unfair bit rate is close to 0. This indicates that proportional-fair and round-robin are close to perfectly relative-throughput fair. For max-rate the unfair bit rate increases with the measurement interval size.

The normalized version of the statistical time-access index, $f(i, j, x, t_1, t_2)$, is defined as:

$$\Pr \left\{ \left| \frac{\alpha(i, t_1, t_2)}{t_2 - t_1} - \frac{\alpha(j, t_1, t_2)}{t_2 - t_1} \right| \geq x \right\} \leq f(i, j, x, t_1, t_2)$$

Fig. 8 and 9 depict the statistical time-access index for the measurement interval size 0.8 and 8 seconds. The dotted line in Fig. 9 shows that for max-rate, the probability that the difference in the normalized time served between two users is equal to 0.2, is approximately 0.6. For round-robin and proportional-fair this probability is close to 0 in Fig. 9. For a difference equal to 0, the probability is 1 for all scheduling algorithms in Fig. 8 and 9. The area marked off by the x-axis, y-axis and $f(i, j, x, t_1, t_2)$ gives information on how fair a scheduling algorithm is. A small area indicates a scheduling algorithm that is relatively more resource-fair (time-access fair) than a scheduling algorithm with a large area. The curves for round-robin and proportional-fair is close to the y-axis in the figures, indicating that they are resource-fair. Comparing Fig. 8 and 9 it can be noted that the area for max-rate is smaller in Fig. 8 than in Fig. 9. This means that the max-rate scheduler is more resource-fair in Fig. 8 than in Fig. 9.

8 Discussion and conclusions

If we want to take TCP characteristics and its interaction with layer 2 scheduling into account, the performance of scheduling algorithms should be evaluated considering layer 4 throughput and layer 4 fairness. As fair throughput in cellular networks in most cases is not the same as equal throughput, Jain's fairness index is not an appropriate layer 4 fairness measure for cellular networks.

Round-robin is in this paper used as a reference algorithm for measuring fairness. The motivation behind choosing round-robin as reference algorithm is that round-robin is resource-fair and easy to implement.

Numerical evaluation of round-robin, proportional-fair and max-rate scheduling shows that the new layer 4 measures presented in this paper can identify if the algorithms are fair or unfair. This means that the worst case TCP

fairness index and the TCP fairness index are appropriate layer 4 fairness measures for measuring fairness of scheduling algorithms scheduling TCP traffic in cellular networks. The worst case TCP fairness index should primary be used. If the index is less than one, the TCP fairness index can be used to get more information on the level of relative-goodput fairness.

In some aspects, the results using the two new layer 4 measures are similar to the results using the statistical time-access index or the wireless absolute fairness bound. Considering relative-goodput fairness, relative-throughput fairness or resource-fairness, the measures indicate that max-rate is unfair and, for our simulation setup, fairness decreases with increasing measurement interval. Furthermore, proportional-fair is considered close to perfectly fair (relative-goodput fair, relative-throughput fair or resource-fair).

Even if the layer 4 and layer 2 results in some aspects are similar, it is still better to measure fairness using the new layer 4 measures. For other scheduling algorithms and simulation setups the layer 4 and layer 2 results might differ. Also, the worst case TCP fairness index has the advantage that the results are easy to interpret. The index gives a direct indication if users are likely to be dissatisfied with a scheduling algorithm.

References

- [1] U. Bodin and A. Simonsson. Effects on TCP from radio-block scheduling in wcdma high speed downlink shared channels. In *Proc. of Fourth COST 263 International Workshop on Quality of Future Internet Services*, 2003.
- [2] J. Holtzman. Asymptotic analysis of proportional fair algorithm. In *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages F33–F37, 2001.
- [3] IKR Simulation Library. <http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/>.
- [4] IT++. <http://itpp.sourceforge.net/>.
- [5] R. Jain. *The art of computer systems performance analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York, 1991.
- [6] W. S. Jeon, D. G. Jeong, and B. Kim. Packet scheduler for mobile internet services using high speed downlink packet access. *IEEE Transactions on Wireless Communications*, 3(5):1789–1801, Sept. 2004.
- [7] S. Keshav. *An Engineering Approach to Computer Networking*. Addison Wesley, 1997.
- [8] T. Klein, K. Leung, and H. Zheng. Improved TCP performance in wireless IP networks through enhanced opportunistic scheduling algorithms. In *Proc. IEEE Global Telecommunications Conference*, pages 2744–2748, Dallas, TX USA, 2004.
- [9] Y. Liu, S. Gruhl, and E. W. Knightly. WCFQ: an opportunistic wireless scheduler with statistical fairness

bounds. *IEEE Transactions on Wireless Communications*, 2(5):1017–1028, Sept. 2003.

- [10] K. Norlund, T. Ottosson, and A. Brunstrom. Fairness measures for best effort traffic in wireless networks. In *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2953–2957, Barcelona, Spain, 2004.
- [11] T. S. Rappaport. *Wireless Communications: Principles and practice*. Prentice Hall, 2nd edition, 2002.
- [12] S. Shakkottai and A. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in HDR. In *Proc. of the 17th International Teletraffic Congress*, pages 793 – 804, Salvador da Bahia, Brazil, 2001.
- [13] P. Viswanath, D. N. C. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Wireless Communications*, 48(6):1277–1294, June 2002.
- [14] W. Wang, T. Ottosson, M. Sternad, A. Ahlén, and A. Svensson. Impact of multiuser diversity and channel variability on adaptive OFDM. In *Proc. Vehicular Technology Conference*, pages 547 – 551, Fall 2003.

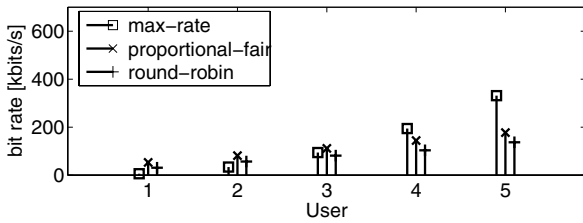


Figure 2. Goodput, 0.8 s measurement interval.

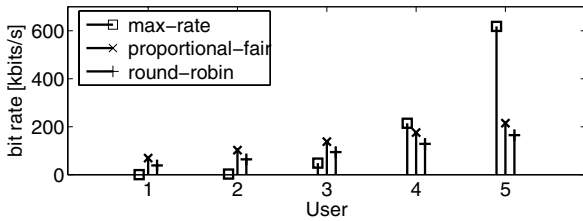


Figure 3. Goodput, 800 s measurement interval.

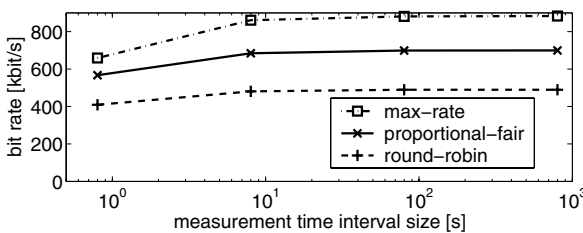


Figure 4. Total goodput.

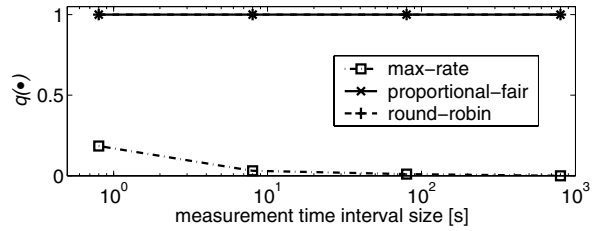


Figure 5. Worst case TCP fairness index.

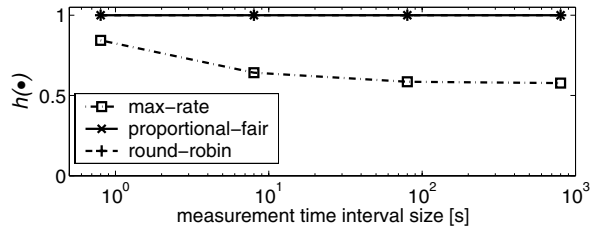


Figure 6. TCP fairness index.

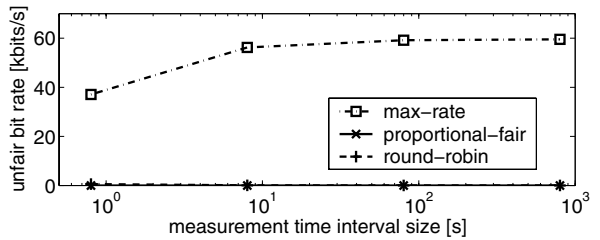


Figure 7. Wireless absolute fairness rate.

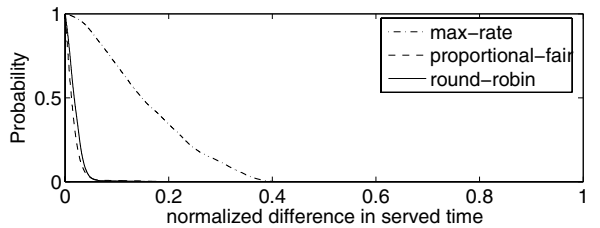


Figure 8. Statistical time-access index, measurement interval = 0.8 s.

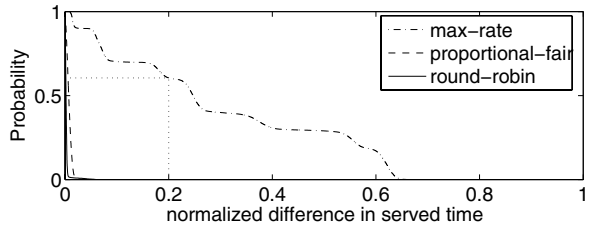


Figure 9. Statistical time-access index, measurement interval = 8 s.