

# On the Vulnerability of the Proportional Fairness Scheduler to Retransmission Attacks

Udi Ben-Porat<sup>\*</sup>, Anat Bremler-Barr<sup>§</sup>, Hanoach Levy<sup>‡</sup> and Bernhard Plattner<sup>\*</sup>

<sup>\*</sup>Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland

Email: {ehudb, plattner}@tik.ee.ethz.ch

<sup>§</sup>Computer Science Dept., Interdisciplinary Center, Herzliya, Israel

Email: bremler@idc.ac.il

<sup>‡</sup>Computer Science Dept., Tel-Aviv University, Israel

Email: hanoach@cs.tau.ac.il

**Abstract**—Channel aware schedulers of modern wireless networks – such as the popular Proportional Fairness Scheduler (PFS) – improve throughput performance by exploiting channel fluctuations while maintaining fairness among the users. In order to simplify the analysis, PFS was introduced and vastly investigated in a model where frame losses do not occur, which is of course not the case in practical wireless networks. Recent studies focused on the efficiency of various implementations of PFS in a realistic model where frame losses can occur. In this work we show that the common straight forward adaptation of PFS to frame losses exposes the system to a malicious attack (which can alternatively be caused by malfunctioning user equipment) that can drastically degrade the performance of innocent users. We analyze the factors behind the vulnerability of the system and propose a modification of PFS designed for the frame loss model which is resilient to such malicious attack while maintaining the fairness properties of original PFS.

## I. INTRODUCTION

The objective of a base-station scheduler in a wireless network is to schedule the downlink transmission of data frames to the various users in a fair and efficient way. Suppose at time slot  $t$  the scheduler transmits to user  $U_i$ . What should the scheduler do at slot  $t + 1$  if  $U_i$  told him that the frame of slot  $t$  did not arrive properly (NACK)? On the one hand, if the loss is due to the network fault, it would be fair to prioritize  $U_i$  at  $t + 1$  and retransmit to it. On the other hand, if  $U_i$  is malicious and just announced a fake NACK (or simply has a very bad handset) it would be both fair and efficient *not to* retransmit to it at  $t + 1$ . But how one can distinguish between "real" and "fake" NACKs? The dilemma and problem will of course be drastically magnified if  $U_i$  repeatedly (as many times as it wishes) announces NACKs as response to retransmissions. This problem of efficiently and fairly scheduling frame transmission subject to the inherent challenge of having to distinguish between "real" and "fake" NACK's is the subject addressed in this paper.

High-speed wireless networks are becoming increasingly common. Along with that, the strategy of *scheduling* the high-speed data - which is vital to the network performance - has become the subject of active research. The modern wireless networks standards allow new generation of channel aware schedulers. One of the most popular such schedulers is the *Proportional Fairness Scheduler* (PFS) [1] [2] which

improves throughput performance by exploiting channel fluctuations while maintaining fairness among the users. PFS was introduced and vastly investigated in a simple model where frame losses do not occur - referred to as *Lossless Model* in this work - which is of course not a realistic model for wireless networks. A more practical model does allow frame losses – referred to as *Loss Model* in this work – and leaves the implementation details of retransmission as an open question. Extensive work has been focused on the *efficiency* of retransmissions in channel aware schedulers in general, and specifically in PFS [3], [4] and [5]. As far as we know, no work has been conducted on examining the *vulnerabilities* of retransmissions in PFS to malicious attacks and this is the subject of this paper.

Attacks on channel aware schedulers were investigated in recent studies in other contexts. In [6] it was shown that a coordinated malicious attack on the CDF scheduler [7] can lead to loss of throughput and time share for other users in the system and [8] and [9] revealed the vulnerability of PFS to delays/jitter and loss of throughput and time share caused to regular users by malicious users providing false channel capacity reports.

In this work we study PFS and its frame loss handling using retransmission strategies. We focus on the vulnerability of PFS and its frame-loss handling variants to attacks on the retransmission mechanism.

We start (Section II) with laying out the definition of PFS in the *Lossless Model* and the common ways to make it handle losses. We describe two such treatment approaches – *PFS with Slow Retransmissions* (PFS-SR) and *PFS with Fast Retransmissions* (PFS-FR).

We start the analysis (Section III) by showing that in the *Loss Model* the original PFS scheme can suffer significantly from a DDoS attack directed on the retransmission mechanism. The throughput allocated to innocent users drastically drops in the presence of malicious users who simply react with negative acknowledgments (NACK's) to transmissions sent to them. For example, we show loss of 40% in time share for regular users when 10% of the users are malicious. A by product of this analysis is a new result stating that under the assumption of constant rates the priority values of PFS all converge to the

same value.

In Section IV we propose the *Transmissions Average* approach to counter this attack and examine its properties. We show that while this solution cures the vulnerability to malicious attack, it distorts the fairness of the scheduler. In Section V we propose a second solution – the *Effective Average* approach – and show that it maintains *Proportional Fairness* (as defined by Kelly [10]) as well as being resistant to malicious retransmission attacks. The result is shown to hold for general rate conditions.

Finally, in Section VI we investigate PFS-FR. We show that PFS-FR is subject to the same vulnerability as PFS-SR is. Then we show that both of the solutions that we proposed for PFS-SR (*Transmissions Average* and *Effective Average*) do not work for PFS-FR. We propose a third solution (*Initial Effective Average*) which cures the vulnerability and maintains fairness as the original PFS-FR.

## II. BACKGROUND

In this section we describe the *Proportional Fairness Scheduler* (PFS) and the different environments in which it is analyzed in this work. We start with the definition of PFS in the *Lossless Model* - in which frame losses never occur - where it was originally defined and vastly investigated. Next we describe PFS in the *Loss Model* - in which frame losses can occur. Lastly, we define the *Constant Rate Assumption* (CRA) where each user experiences a constant channel condition.

### A. The Proportional Fairness Scheduler (PFS)

The wireless communication system discussed in this work consists of a base station which serves its users with down-link data. The time is divided into time slots and the base station is able to send data to exactly one user in each time slot. The role of the scheduler (of the base station) is to decide which of the users will be the next to receive data. PFS does that by assigning priority values to all users for every time slot, then the user with the highest priority is scheduled for transmission. Denote user  $i$  with  $U_i$  and his priority value for the scheduling decision of time  $t$  by  $V_i(t)$ . Then

$$V_i(t) = R_i(t)/A_i(t), \quad (1)$$

where  $R_i(t)$  is the bit rate (measured in bits per time slot *bits/slot*) in which the system sends data to  $U_i$  if he is assigned with time slot  $t$ . The value of  $R_i(t)$  is decided by the scheduler according to the channel condition of the user as expressed by the Signal to Noise Ratio (SNR) reported by the user for every time slot.  $A_i(t)$  is the throughput average of  $U_i$  until time slot  $t$  (not including) measured in *bits/slot*. The throughput average is updated after every time slot. One method of updating the average is:

$$A_i(t+1) = \frac{t-1}{t}A_i(t) + \frac{1}{t}R_i(t)1_i^{rcv}(t). \quad (2)$$

Another method of averaging is "discount averaging" which gives less weight to the data received in the past:

$$A_i(t+1) = (1-\epsilon)A_i(t) + \epsilon R_i(t)1_i^{rcv}(t), \quad (3)$$

where  $1_i^{rcv}(t) = 1$  if  $U_i$  received data on slot  $t$ ,  $1_i^{rcv}(t) = 0$  otherwise. In this work we refer to such averaging methods (Eq. 2 or 3) as *Admitted Average* since in the *Loss Model* (where frame loss are possible)  $1_i^{rcv}(t) = 1$  only if the user admitted to receive the transmission successfully by reporting ACK. We do so in order to differ it from new averaging methods we suggest later in this work. According to the above equations, the throughput average of the user assigned for transmission is not decreased like the averages of all the other users. It means that in the next time slot it is harder for him to obtain the highest priority value (Eq. 1). Therefore, throughout the work we refer to the value multiplied with  $1_i^{rcv}(t)$  (in both methods) as the "price" that the user "pays" for "winning" the time slot. The higher the price is, the higher his updated throughput average will be and this lower his potential priority value in the following time slots (Eq. 1).

Note that in practice it is hard to differ between both averaging methods (Eq. 2 and Eq. 3) especially when the common recommended value  $\epsilon = 0.001$  is used [1]. In order to simplify the analysis, throughout this work we choose to use the throughput average updating method as described in Eq. 2. In addition, it was proved in [11] that regardless of the method used, PFS complies with the *Proportional Fairness* criterion given by Kelly [10], [12] which maximizes the utility function  $\theta$  given in Eq. 4.

$$\theta = \sum_{i=1}^N \log(A_i), \quad (4)$$

where  $N$  is the number of users in the system and  $A_i$  is the throughput average of  $U_i$  under the steady state.

### B. Frame Loss Handling Mechanisms

PFS as described in the previous section was introduced and defined in the *Lossless Model*. When implementing PFS for the *Loss Model*, that is, an environment where frame losses can occur, two major issues have to be addressed: 1) *Effective Rate* evaluation; 2) Frame loss handling.

**Effective Rate Evaluation:** In PFS for the *Lossless Model* the numerator of the priority value is  $R_i(t)$  - the data rate in which the SNR reported by the user was mapped to. If frame losses can occur then the rate  $R_i$  in which we send data to the user does not represent the actual bit-rate the user receives successfully. Therefore, in the *Loss Model*,  $R_i(t)$  at the numerator of the priority value (Eq. 1) is replaced with an *Effective Rate* value, denoted by  $R_i^e(t)$ , which is the rate that the user is expected to receive successfully (also measured with *bits/slot*). The common way to calculate the effective rate is by  $R_i^e(t) = G_i(t)R_i(t)$  where  $G_i(t)$  is the probability of successful transmission when sending data in rate  $R_i(t)$  given the SNR value he reported for that time slot. Eq. 5 concludes the calculation of the priority value in the *Loss Model*:

$$V_i(t) = \frac{R_i^e(t)}{A_i(t)} = \frac{G_i(t)R_i(t)}{A_i(t)}. \quad (5)$$

For example, if the actual rate that will be transmitted to the user is  $R_i(t) = 100$  *bits/slot* and the probability that the user

will receive it correctly is  $G_i(t) = 0.8$ , then the effective rate is  $R_i^e(t) = 80 \text{ bits/slot}$ . More about the calculation of error probability and the effective rate for PFS in the *Loss Model* can be found in [4].

**Frame Loss Handling:** The second issue that has to be addressed is how to handle cases of lost frames, that is, how to handle a frame waiting to be retransmitted after its previous transmission failed (the user replied with NACK). There are two approaches to address it:

- *Fast Retransmission (PFS-FR)* - The frame will be immediately retransmitted to the user in the next time slot whether he obtains the highest priority value or not.
- *Slow Retransmission (PFS-SR)* - The frame will be retransmitted only in the earliest time slot the user obtains the highest priority value.

Unless stated otherwise, throughout the work we discuss PFS-SR. PFS-FR is discussed separately in Section VI. Note that (for both PFS-SR and PFS-FR) the throughput average ( $A_i$ ) of a user is the *Admitted Average* as defined in II-A since it represents the average of the actual data he received also in the *Loss Model*. Formally, Eq. 2 (or 3) is used as is, and  $1_i^{rcv}(t) = 1$  only if the user received the frame on time  $t$ , that is, his feedback for the frame is ACK. As mentioned in Section II, we refer to this throughput average method as *Admitted Average*. Note that in order to prevent an abuse of the system, the system limits the number of possible retransmissions to a fixed value -  $L_{max}$ . If the limit is reached then the throughput average is updated as if he received the frame successfully even if his reported NACK.

### C. Constant Rate Assumption (CRA)

The *Constant Rate Assumption* (CRA) is a special case where each user experiences a constant channel condition  $\forall_{i,t}. R_i(t+1) = R_i(t)$  (simply denoted by  $R_i$  under CRA). Note that it is still possible that  $R_i \neq R_j$  for different users  $i$  and  $j$ . In our work, CRA is used to prove claims that a scheduler is vulnerable or unfair<sup>1</sup>. We do so since analyzing the behavior of PFS under CRA allows clearer understanding of PFS by preserving the influence of the changing throughput average of the users on their priority values and hence on the scheduling decisions. Note that in order to prove such negative claims, it is enough to prove them in a limited family of scenarios such as CRA. We also give an intuitive explanation (backed up by numerical results) why these claims hold also when the rates are not constant. In addition, it is important to note that CRA is *not* used in the proofs of positive claims for the immunity and fairness of a scheduler, that is, the immunity and fairness of the schedulers we recommend are proved without any limiting assumption. Note that since we focus on the behavior of PFS under the steady state, throughout the work we simplify the analysis by assuming that all users arrive at the same time ( $t=0$ ), no users are arriving or leaving the system and that each user has an infinite backlog of data.

## III. RETRANSMISSION ATTACKS AND THEIR ANALYSIS

In this section we show that PFS-SR is vulnerable to a malicious *Retransmission Attack* which causes loss of time share to the regular users in the system. In Section VI we will show that PFS-FR is vulnerable to such attacks as well. As far as we know, this work is the first to raise this vulnerability of PFS with retransmissions. First we describe the attack and give an intuitive explanation why in PFS-SR it allows a malicious user to gain more time share (on the expense of others) than he should. Then in theorems 2 and 3 we show the impact of a malicious attack under CRA. Lastly, we give numerical results of the time share created by such an attack in a system where the channel conditions of the users change over time (not under CRA) which shows that the vulnerability to retransmission attacks is not limited to CRA.

PFS-SR is vulnerable to a *Retransmission Attack* in which a malicious user reports NACK even when he receives the frame correctly. To carry the attack the malicious user does not have to persistently report NACK, just report NACKs more frequently than what the system expects. This kind of behavior results in an increased time share for the malicious user(s) on the expense of the regular users served by the scheduler. The vulnerability of the scheduler to such attacks is explained in the following way: In the *Loss Model*, if  $U_i$  reported NACK, then  $1_i^{rcv}(t) = 0$  and his average decreases exactly as for the other users  $\forall_{j \neq i}. \frac{A_i(t+1)}{A_j(t+1)} = \frac{A_i(t)}{A_j(t)}$ . That is, the fact that the scheduler has put efforts in transmitting data to  $U_i$  at time  $t$  does not count against him in future time slots (unlike the *Lossless Model* where if a frame transmitted to a user it means he received it). A malicious user can abuse this insensitivity property of the *Admitted Average* and report NACK in every opportunity he has<sup>2</sup>. As explained earlier, if NACK was reported for time  $t$  then the proportion between the *throughput averages* of all users stay the same. Moreover, the following lemma shows that under CRA the proportion between the *priority values* of the users stay the same.

**Lemma 1.** *For PFS-SR under CRA, if  $R_i(t) = R_i(t+1)$ ,  $R_j(t) = R_j(t+1)$  and  $1_i^{rcv}(t_0) = 1_j^{rcv}(t_0) = 0$  then  $\frac{V_i(t_0+1)}{V_j(t_0+1)} = \frac{V_i(t_0)}{V_j(t_0)}$ .*

*Proof:* Let  $T_i(t) = \sum_{s=1}^{t-1} 1_i^{rcv}(s)$  be the number of frames  $U_i$  received during time slots  $[1, t-1]$ . Since  $1_i^{rcv}(t_0) = 1_j^{rcv}(t_0) = 0$  then  $T_i(j)(t_0+1) = T_i(j)(t_0)$ . Therefore, According to Lemma 4 (at the Appendix)  $V_i(t_0+1)/V_j(t_0+1) = (G_i \frac{t_0+1}{T_i(t_0)}) / (G_j \frac{t_0+1}{T_j(t_0)}) = (G_i \frac{t_0}{T_i(t_0)}) / (G_j \frac{t_0}{T_j(t_0)}) = V_i(t_0)/V_j(t_0)$ . ■

**Corollary 1.** *If  $U_i$  obtained the highest priority value at time  $t$  and reported NACK, then he will obtain the highest priority value at time  $t+1$  as well Lemma 1. Hence, a malicious user reporting NACK over and over again will win consecutive time slots and when he finally reports ACK his throughput average is updated with the size of the frame regardless of how many times it was transmitted.*

<sup>1</sup>CRA is also used in [9] to analyze a malicious attack on PFS.

<sup>2</sup>As long as the retransmissions limit ( $L_{max}$ ) is not reached.

In the following analysis we investigate PFS-SR (under CRA) in order to understand and quantify the damage a retransmission attack causes for other users in the system. While Corollary 2 concludes that in the long run in the absence of malicious users all users should get equal time share, Theorem 2 shows that a malicious user can get a larger time share than a regular user. Theorem 3 shows how the time share of a regular user shrinks as a function of the number of malicious users and their behavior. In order to reach these results, we analyze the scheduling decisions of PFS-SR under CRA in the long run. In order to do so, we start with Theorem 1 which shows a strong property of PFS-SR under CRA. It shows the priority values of all users (malicious or not) converge to the same value under the steady state. Note that the numerical results show that this property also holds in the general model (not only under CRA). Note that for the sake of the reading flow, some of the lemmas supporting the theorems of this section are located at the appendix.

**Theorem 1.** *For PFS-SR with Admitted Average under CRA, for every two users  $U_i, U_j$  and  $\epsilon > 0$  there is a time slot  $t'$  such that  $\forall_{t > t'}. \frac{1}{1+\epsilon} \leq \frac{V_i(t)}{V_j(t)} \leq 1 + \epsilon$ . Hence, under the steady state, the priority values of all users converge to the same value.*

*Proof:* Let us choose  $t'$  such that<sup>3</sup> if  $1_i^{rcv}(t') = 1, \frac{1}{T_i(t')} \leq \epsilon, \frac{1}{T_j(t')} \leq \epsilon$  and  $V_j(t' + 1) > V_i(t' + 1)$ . We prove the claim by induction. First we prove the claim for  $t = t' + 1$ . Since  $1_i^{rcv}(t') = 1$  and  $V_j(t' + 1) \geq V_i(t' + 1)$  then according to Lemma 6,  $1 \leq \frac{V_j(t'+1)}{V_i(t'+1)} \leq 1 + \frac{1}{T_j(t')}$ . Since  $\frac{1}{T_i(t')} \leq \epsilon$  then  $1 \leq \frac{V_j(t'+1)}{V_i(t'+1)} \leq 1 + \epsilon$  and the claim holds. Now we assume the claim holds for  $t = t_0 - 1 > t$  and prove by induction the claim for  $t = t_0$ . There are five possible cases: 1.  $1_i^{rcv}(t_0 - 1) = 1_j^{rcv}(t_0 - 1) = 0$ ; 2.  $1_i^{rcv}(t_0 - 1) = 1$  and  $V_j(t_0) \geq V_i(t_0)$ ; 3.  $1_i^{rcv}(t_0 - 1) = 1$  and  $V_j(t_0) < V_i(t_0)$ ; 4.  $1_j^{rcv}(t_0 - 1) = 1$  and  $V_i(t_0) \geq V_j(t_0)$ ; 5.  $1_j^{rcv}(t_0 - 1) = 1$  and  $V_i(t_0) < V_j(t_0)$ . We exclude the proofs for cases 4 and 5 as they are symmetric to cases 3 and 4 respectively. If  $1_i^{rcv}(t_0 - 1) = 1_j^{rcv}(t_0 - 1) = 0$  then according to Lemma 1  $\frac{V_i(t_0)}{V_j(t_0)} = \frac{V_i(t_0+1)}{V_j(t_0+1)}$  and the claim holds by the induction assumption. If  $1_i^{rcv}(t_0 - 1) = 1$  and  $V_j(t_0) \geq V_i(t_0)$  then the claim is proved in the same way we proved for  $t = t' + 1$  (Lemma 6). If  $1_i^{rcv}(t_0 - 1) = 1$  but  $V_j(t_0) < V_i(t_0)$  then according to Lemma 5  $V_i(t_0) < V_i(t_0 - 1)$  and  $V_j(t_0) > V_j(t_0 - 1)$ , therefore  $\frac{V_i(t_0)}{V_j(t_0)} < \frac{V_i(t_0-1)}{V_j(t_0-1)}$ . According to the induction assumption for  $t = t_0 - 1$  that  $\frac{V_i(t_0-1)}{V_j(t_0-1)} \leq 1 + \epsilon$  we get that  $\frac{V_i(t_0)}{V_j(t_0)} \leq 1 + \epsilon$ . Lastly, since  $V_j(t_0) < V_i(t_0)$  then  $1 \leq \frac{V_i(t_0)}{V_j(t_0)}$  therefore the claim holds and we conclude that in all cases the claim holds for  $t_0$ . ■

**Corollary 2.** *Observe that for regular users  $q_i(t) = G_i p_i(t)$  where  $p_i(t) = \frac{T_i(t)}{t}$  is the time share of  $U_i$  during time slots  $[0, t - 1]$  (that is, the probability for a time slot to be assigned*

*to  $U_i$  so far), therefore, since according to Lemma 7 the proportion  $\frac{G_i}{q_i(t)}$  converges to the same value in the long run, then also therefore the time share  $p_i(t)$  of all users converges to the same value, that is, the scheduler enforces time share fairness among regular users under CRA<sup>4</sup>.*

**Theorem 2.** *Let  $U_r$  be a regular user and  $U_m$  be a malicious one with  $G_m^{rep} < G_m$  where  $G_m$  is the probability for a successful transmission (to the malicious user) as predicted by the scheduler and  $G_m^{rep}$  is the probability for the malicious user to reply on a transmission with ACK in reality. Then  $p_m(t)$  and  $\alpha_m p_r(t)$  – where  $\alpha_m = \frac{G_m}{G_m^{rep}} > 1$  – converge to the same value, that is in the long run  $U_m$  has larger time share.*

*Proof:* According to Lemma 7  $\frac{G_m}{q_m(t)}$  converges to the same value as  $\frac{G_r}{q_r(t)}$ . Since  $U_r$  is a regular user then  $q_r(t) = G_r p_r(t)$  we get that  $\frac{q_m(t)}{G_m}$  converges to the same value as  $\frac{1}{p_r(t)}$ . Since  $q_m(t) = G_m^{rep} p_m(t)$  we get  $p_m(t) = \frac{G_m}{G_m^{rep}} p_r(t)$  ■

**Theorem 3.** *Let  $p_r$  be the time share of a regular user in the long run in a system with  $N$  users where  $M$  of them are malicious. Then*

$$p_r = 1 / (N + \sum_{i=1}^M (\alpha_i - 1)). \quad (6)$$

*Proof:* As proved in Corollary 2 the time share of all regular users is converging to the same value<sup>5</sup>, so we simply denote it by  $p_r$ . Then, according to Lemma 2, the time share of a malicious user  $U_m \in \{U_m\}_{m=1}^M$  is  $p_m = \alpha_m p_r$ . Since every time slot is assigned to exactly one user then  $\sum_{i=1}^N p_i = 1$ , hence  $(N - M)p_r + \sum_{m=1}^M \alpha_m p_r = 1$  and we conclude the above theorem. ■

For example, in a system with  $N = 50$ , if  $M = 20$  of them are malicious users with  $\alpha_m = 3.5$ , then the time share of a regular user drops from  $1/N = 1/50$  to  $1/100$ , that is the malicious users cause a 50% loss of time share for all the regular users in the system. Observe that there are two factors that increase the damage of the attack. The first is the number of malicious users in the system ( $M$ ) which is a factor of the resources (network subscriptions, physical devices etc.) available to the attacker. The second is the  $\alpha_m$  ratio. Note that the maximal  $\alpha_m$  value derives directly from the system parameters, hence the system administrator can predict (and reduce) the vulnerability of his system. The minimal  $G_m^{rep}$  value a malicious user can obtain is when he replies with NACK as much as he can for every frame that is sent to him. Since the session length is limited to  $L_{max}$ , the minimal value of  $G_m^{rep}$  is  $1/L_{max}$ , hence according to our analysis the system is more vulnerable as either  $L_{max}$  or the share of the malicious users in the system increases.

**Corollary 3.** *In PFS-SR under CRA with  $N$  users where  $M$  of them are malicious and all malicious users report the same*

<sup>3</sup>Such  $t'$  must exist, otherwise it can be proved that  $\exists \tilde{t}. \forall t > \tilde{t}. : V_i(t) > V_j(t)$  which is impossible since no user can be starved forever.

<sup>4</sup>The result of time share fairness was observed in [9] for regular users in PFS under CRA but in a model assuming no frame loss.

<sup>5</sup>Note that Corollary 2 also holds in the presence of malicious users.

channel rate  $R_m$  with corresponding predicted transmission success  $G_m$  and always report NACK until they reach the  $L_{max}$  limit, then

$$\tilde{p}_r/p_r = 1 + (M/N)(G_m L_{max} - 1), \quad (7)$$

where  $\tilde{p}_r$  is the time share a regular gets if there were no malicious users in the system, and  $p_r$  is what he gets in the presence of the  $M$  malicious users.

*Proof:* According to the behavior of the malicious users  $G_m^{rep} = 1/L_{max}$ , hence  $\alpha_m = L_{max}G_m$  for every malicious user. Then, according to Theorem 3 we get that  $p_r = 1/(N + M(L_{max}G_m - 1))$ . In a system with  $N$  users where non of them is malicious the time share of every user is  $\tilde{p}_r = 1/N$  as (derives from Theorem 3 where  $M = 0$ ), hence we get Eq. 7. ■

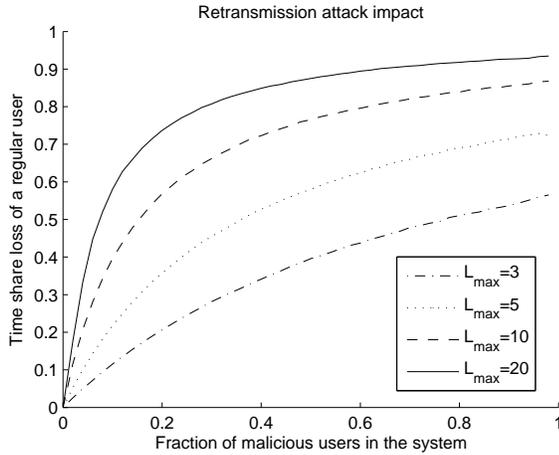


Fig. 1. Results of time share loss under the steady state for a regular user. These results were derived from simulations of PFS-SR with 50 users experiencing random channel conditions according to the Rayleigh fading model.

Figure 1 depicts simulations results in a system where all users experience variable channel condition (according to the commonly used Rayleigh fading model). The malicious users reported NACK over and over again until reached the  $L_{max}$  limit. Note that the value of  $G_m(t)$  is constant only under CRA and not in the realistic model in which these tests have been done. Therefore, in the comparison between the tests and our analytical results we concentrate on the trends and not on the exact loss values. We can see that as in Eq. 7, the time share loss of a user increases with  $L_{max}$  and the fraction of malicious users in the system increases, and see that the time share loss is already very high in a modest presence of malicious users in the system. For example, when  $L_{max} = 10$  and 10% of the users are malicious, every regular user loses 40% of his time share. The results in Figure 1 are of simulations of systems with 50 users, but additional tests showed that the time share loss was identical in systems with different number of users as long as the fraction of malicious users and the value of  $L_{max}$  are similar. This result fits Eq. 7 which shows that the damage to the system does not depend

on the number of the users in the system, but on the fraction of the malicious users in the system -  $M/N$ . These analytical and simulations results emphasize the need in a better variation of PFS for the *Loss Model* that will be immune to such attacks, which is the subject of the rest of this work.

#### IV. FIRST SOLUTION - TRANSMISSIONS AVERAGE

The problem in Section III was created because the user "pays" according to the amount of resources the system expects him to use, and not according to what he actually uses in reality. Therefore, the motivation behind this solution is to "charge" all the users with the efforts the system invests in them instead of charging them with what they admit they got. This way, the ACK/NACK feedbacks of the users cannot influence the scheduling decisions (through the average throughput). In this solution, after every time slot in which the system transmitted  $R_i(t)$  to the user,  $A_i(t)$  is updated with  $R_i(t)$  regardless of his feedback for this transmission (ACK/NACK). Formally, Eq. 2 is replaced with

$$A_i(t+1) = \frac{t-1}{t}A_i(t) + \frac{1}{t}R_i(t)1_i^{snd}(t), \quad (8)$$

where  $1_i^{snd}(t) = 1 \iff V_i(t) = \max\{V_j(t)\}_{j=1}^N$ , that is  $1_i^{snd}(t) = 1$  if the scheduler sent  $U_i$  a frame at time slot  $t$  regardless of the ACK/NACK feedback (otherwise  $1_i^{snd}(t) = 0$ ). We refer to this averaging method as *Transmissions Average* since it measures the average of the transmissions instead of what the user received. The following theorem, which holds for users with general channel conditions and not only under CRA, shows that using transmissions average solves the vulnerability to retransmission attacks:

**Theorem 4.** *When using PFS-SR with transmissions average, a malicious user cannot affect scheduling decisions by performing a retransmission attack. Therefore, this attack cannot harm other users in the system.*

*Proof:* The priority value  $V_i(t)$  depends only on  $R_i^e(t)$  and  $A_i(t)$  (5). Obviously,  $R_i^e(t)$  depends only on the SNR reported by the user (and not on his ACK/NACK feedback). In addition, since *Transmissions Average* is used – the value of  $A_i(t)$  is also independent of the NACK/ACK feedbacks of the  $U_i$ . Hence performing a retransmission attack cannot influence any of the scheduling decisions. ■

Using transmissions average makes the user pay per transmission and not per unique frame they receive but it may seem that since all regular users will now have to multiply the "price" they "pay" per bit they receive, then this modified scheduler preserve the proportional fairness among them, while malicious users have to pay more per bit than regular users under this scheduler. It can be proved that this reasoning is true as long as all regular users require the same number of transmissions per frame  $\forall_{i,j,t}. G_i(t) = G_j(t)$ . But when one user is expected to require more transmissions per frame than the others, it means that he pays for every bit he receives more than others and this results in a distorted fairness as shown in Theorem 6. In order to prove that, we first prove

that the priority values of PFS-SR with transmissions average also converge to the same value in the long run.

**Theorem 5.** *Theorem 1 holds also for PFS-SR with transmissions average under CRA.*

*Proof:* Observe that PFS-SR with transmissions average is identical to PFS-SR with *Admitted Average* when  $L_{max} = 1$ . Since the proof of Theorem 1 holds for any  $L_{max}$  value, it holds also when  $L_{max} = 1$ , hence also for PFS-SR with transmissions average. ■

**Theorem 6.** *Even in the presence of only regular users, PFS-SR with transmissions average can distort the fairness severely in a way that its scheduling decisions do not comply with Proportional Fairness or any other notion of fairness.*

*Proof:* We prove the claim by showing an example for such fairness distortion. We show that PFS-SR with transmissions average can lead to a situation where in a system with two regular users, the one with the better channel condition will get less time share and less throughput than the other, and this is of course does not comply with any notion of fairness. Assume a system with two regular users  $U_1$  and  $U_2$  where  $U_1$  has a better channel condition than  $U_2$ . For the sake of notational simplicity we refer to  $A_i$  and  $p_i$  as the *Transmissions Average* and the time share (respectively) of  $U_i$  in the long run. That is,  $R_1^e = R_1 G_1 > R_2^e = R_2 G_2$ . In addition we assume  $G_2 > G_1$ . It is possible as long as  $R_1/R_2 > G_2/G_1$ . Since  $G_2 > G_1$ , then obviously  $R_1/R_2 > 1$  and therefore  $R_1/R_2 > \sqrt{R_1/R_2}$ . We will now prove the claim for  $G_1$  and  $G_2$  such that  $R_1/R_2 > G_2/G_1 > \sqrt{R_1/R_2}$ . According to Theorem 5 the priority values of the users converge to the same values in the long run, that is  $R_1^e/A_1 = R_2^e/A_2$ , then  $\frac{R_1 G_1}{A_1} = \frac{R_2 G_2}{A_2}$ . Let  $A_i^{rcv}$  be the average of throughput  $U_i$  actually received in the long run. Then  $A_i^{rcv} = G_i A_i$ , and therefore  $\frac{R_1 (G_1)^2}{A_1^{rcv}} = \frac{R_2 (G_2)^2}{A_2^{rcv}}$ , and we get  $A_2^{rcv} = \frac{R_2 (G_2)^2}{R_1 (G_1)^2} A_1^{rcv}$ . When  $\frac{G_2}{G_1} > \sqrt{\frac{R_1}{R_2}}$  we get that  $A_2^{rcv} > A_1^{rcv}$ , hence  $U_2$  receives more throughput. Lastly, since also  $A_i^{rcv} = p_i R_i^e$  then  $p_1 R_1^e < p_2 R_2^e$  and since  $R_1^e > R_2^e$  we get that  $p_2 > p_1$ , that is, the time share of user  $U_2$  is larger than that of  $U_1$ . ■

The reason that the transmissions average distorts the fairness is that it makes the proportion  $\frac{G_i}{p_i(t)}$  converge to the same value for all the users (derives from Theorem 5 and the definition of the effective rate  $R_i^e = G_i R_i$ ). Then, a user with relatively (to others) high frame error rate (that is, his  $G_i$  is low) requires more retransmissions per frame than the others, that is, he will have to "pay" more for every bit he actually receives unlike in the *Admitted Average* (where the "payment" per bit is equal among all users). In order to solve this flaw, in the following section we suggest the *Effective Average* where the proportion that converges is independent of  $G_i$ .

## V. SECOND SOLUTION - EFFECTIVE AVERAGE

In this section we propose a different averaging method – *Effective Average* – which is free of the flaw of the

*Transmissions Average* and hence can be proved to maintain *Proportional Fairness* in addition to being invulnerable to retransmission attacks. In this solution the throughput average  $A_i(t)$  is the expected throughput  $U_i$  should have gotten so far, given the efforts the system has put into him. Formally, Eq. 2 is replaced with

$$A_i(t+1) = \frac{t-1}{t} A_i(t) + \frac{1}{t} R_i^e(t) 1_i^{snd}(t), \quad (9)$$

where  $1_i^{snd}(t)$  is as defined in Section IV and  $R_i^e(t)$  in Section II-B.

**Remark 1.** *PFS-SR with Effective Average is invulnerable to retransmission attacks. The proof of Theorem 4 holds for PFS-SR with Effective Average.*

While maintaining the resilience to retransmission attacks as PFS-SR with transmissions average, the *Effective Average* also truly relates to the throughput the user actually received as proved in the following theorem:

**Theorem 7.** *When Effective Average is used, the value of  $A_i(t)$  is the expected throughput of  $U_i$  until time  $t$ , given the values of  $1_i(s)$  for  $s \in [1, t-1]$ .*

*Proof:* In *Effective Average*,  $A_i(t) = \sum_{s=1}^{t-1} 1_i^{snd}(s) R_i^e(s)$  (derives from Eq. 9). If given that  $1_i^{snd}(s) = 1$  then, by definition,  $R_i^e(t) = G_i(t) R_i(t)$  is the expected throughput  $U_i$  receives in that time slot. Otherwise, if given that  $1_i^{snd}(s) = 0$  then the throughput at time  $s$  is 0. Hence the claim is proved. ■

In the following Lemma we show a reduction of PFS-SR to the original PFS that was originally defined in the *Lossless Model*, then in Theorem 8 we use it together with Theorem 7 to prove that PFS-SR with *Effective Average* maintains the *Proportional Fairness* criterion as appears at the end of Section II-A, that is, maximizing the utility function in Eq. 4.

**Lemma 2.** *Let system A be a system serving  $N$  users  $\{U_i\}_{i=1}^N$  with PFS-SR using Effective Average in the Loss Model, with the set of all possible effective rates  $\{r_k^e\}_{k=1}^K$ .*

*We will construct a system B such that it serves  $N$  users  $\{\tilde{U}_i\}_{i=1}^N$  with the original PFS in the Lossless Model (As described in Section II) with the set of all possible rates  $\{\tilde{r}_k\}_{k=1}^K$  such that  $\tilde{r}_k = r_k^e$ . The initial throughput averages is  $\forall_i. A_i(1) = \tilde{A}_i(1)$ . In addition,  $\forall_i. R_i^e(t) = r_k^e \iff \tilde{R}_i(t) = \tilde{r}_k$  at time  $t$ . Then the following claims holds: 1.  $\forall_{i,t}. A_i(t) = \tilde{A}_i(t)$ ; 2.  $V_i(t) = \max\{V_j(t)\}_{j=1}^N \iff \tilde{V}_i(t) = \max\{\tilde{V}_j(t)\}_{j=1}^N$  where  $V_i$  and  $\tilde{V}_i$  are the priority values of users  $U_i$  and  $\tilde{U}_i$  respectively.*

*Proof:* We prove by induction the two claims. For  $t = 1$  it is given that  $A_i(1) = \tilde{A}_i(1)$  hence claim 1 holds. Since that  $\forall_i. R_i^e(t) = r_k^e \iff \tilde{R}_i(t) = \tilde{r}_k$  and  $A_i(1) = \tilde{A}_i(1)$  then  $\forall_i. V_i(1) = \tilde{V}_i(1)$  hence claim 2 also holds. Now assume that claims 1 and 2 hold for  $t = t_0$  and we prove them for  $t_0 + 1$ . Let  $i$  be some user index, then  $A_i(t_0 + 1) = \frac{t_0-1}{t_0} A_i(t_0) + \frac{1}{t_0} 1_i^{snd}(t_0) R_i^e(t_0)$  and  $\tilde{A}_i(t_0 + 1) = \frac{t_0-1}{t_0} \tilde{A}_i(t_0) + \frac{1}{t_0} 1_i^{rcv}(t_0) \tilde{R}_i(t_0)$ . It is given that  $\forall_i. R_i^e(t) =$

$r_k^e \iff \tilde{R}_i(t) = \tilde{r}_k$ . In addition, according to claim 2 of the induction assumption  $1_i^{snd}(t_0) = 1_i^{cv}(t_0)$ . Lastly according to claim 1  $A_i(t_0) = \tilde{A}_i(t_0)$  hence  $A_i(t_0 + 1) = \tilde{A}_i(t_0 + 1)$  and claim 1 holds for  $t_0 + 1$  as well. Lastly, since claim 1 holds for every  $i$  in  $t_0 + 1$  and  $\forall_{i,t}. R_i^e(t) = r_k^e \iff \tilde{R}_i(t) = \tilde{r}_k$ , then  $\forall_i. \tilde{V}_i(t_0 + 1) = V_i(t_0 + 1)$ , therefore claim 2 holds as well. ■

**Theorem 8.** *PFS-SR with Effective Average maintains Proportional Fairness. That is, maximizing the utility function (Eq. 4).*

*Proof:* According to Lemma 2, for every system A using PFS-SR with effective throughput, there is an original PFS scheduler in the *Lossless Model* in a parallel system B that takes parallel scheduling decisions and maintains the same  $A_i$  values for the users. The original PFS algorithm in the *Lossless Model* was proved to maximize the utility function in 4 (see [11]), hence PFS-SR with *Effective Average* maximizes these utility function as well, the only difference is the meaning of  $A_i$  in the different schedulers. While in scheduler B  $A_i(t)$  represents the bit rate the user admitted to receive until time slot  $t$ , in scheduler A,  $A_i(t)$  represents the average throughput the scheduler expects the user to receive considering the efforts invested in him, that is, the expected bit rate of the user. Therefore, if the effective rate predicted by the scheduler is correct, then the value of  $A_i$  in the long run represents the real throughput average of a user in the long run, and the sum of effective rates maximizes the utility function in Eq. 4. ■

Note that this proof holds for a general channel rate conditions of the users. Hence PFS-SR with *Effective Average* is proved to be resilient to retransmission attacks while maintaining the original proportional fairness. Hence this is a safe and fair way to implement PFS in a frame loss environment.

## VI. PFS WITH FAST RETRANSMISSIONS SCHEDULER (PFS-FR)

In *PFS with Fast Retransmissions* (PFS-FR), if a user replies with a NACK for a frame, then it will be immediately retransmitted to him in the next time slot whether he has the highest priority value or not. PFS-FR persists on a retransmission even if it predicts it will fail in high probability (since the receiver experiences a sudden drop in his channel condition). Therefore, PFS-SR offers better efficiency and fairness since it utilizes the channel fluctuations better than PFS-FR. However, PFS-FR offers better latency<sup>6</sup> and simpler implementation than PFS-SR and might be preferred over PFS-SR for some systems. Note that the efficiency differences between them diminish as the retransmissions limit  $L_{max}$  is smaller.

The rest of this section is arranged as follows: In the first subsection we show that PFS-FR with *Admitted Average* (the original averaging method) is vulnerable to retransmission attacks as PFS-SR with *Admitted Average* is. Then we disqualify the use of PFS-FR with *Transmissions Average* for

the same reason we did with PFS-SR - it distorts the fairness severely. Then we show that in contrary to PFS-SR, PFS-FR with *Effective Average* is vulnerable. We show it creates a new vulnerability that can be abused by both selfish users who seek only their own benefit (on the expense of others) and malicious users who try to maximize the time share loss of other users. Finally, we propose a third averaging method called *Initial Effective Average* which makes PFS-FR immune to such an abuse. In addition, when the channel condition of the users can be assumed to be constant during retransmissions, PFS-FR with *Initial Effective Average* take similar scheduling decisions as PFS-FR with *Admitted Average*. This assumption can be made in a system where the transmission limit  $L_{max}$  is small and/or the channel rate of the users does not change significantly in the short term. Note that this is usually the case in systems where PFS-FR is preferred (over PFS-SR).

### A. Vulnerability of PFS-FR to Retransmission Attacks

In Corollary 4 we use CRA to show that PFS-FR with *Admitted Average* is vulnerable to retransmission attacks. We use the following theorem to infer the vulnerability of PFS-FR directly from the vulnerability of PFS-SR (that was already proved in Section III):

**Theorem 9.** *Under CRA, both PFS-SR and PFS-FR take the same scheduling decisions when Admitted Average is used.*

*Proof:* By definition, the only difference between PFS-SR and PFS-FR (when both use the same average method) is the handling of NACK reports. In Corollary 1 it is shown that for PFS-SR under CRA, when a user reports NACK he is immediately assigned for transmission in the next time slot. Hence both PFS-SR and PFS-FR perform the same scheduling decisions under CRA. ■

**Corollary 4.** *Under CRA, PFS-FR with Admitted Average is vulnerable to retransmission attacks.*

*Proof:* Immediate from Theorem 2 and Theorem 9. ■

### B. PFS-FR with Transmissions Average

As for PFS-SR, using PFS-FR with *Transmissions Average* can prevent malicious users from damaging the system by performing a retransmission attack. However, for PFS-SR it was proved in Theorem 6 that using *Transmissions Average* can lead to a severely distorted fairness, and this is also true for PFS-FR as stated in the following corollary:

**Corollary 5.** *Theorem 6 also holds for PFS-FR. As for PFS-SR with Transmissions Average - it can be proved that also for PFS-FR with Transmissions Average the priority values of the users converge to the same value. Therefore, the analysis of the example given in the proof for Theorem 6 also holds for PFS-FR with Transmissions Average.*

### C. PFS-FR with Effective Average

Corollary 5 concludes that using *Transmissions Average* distorts the fairness of PFS-FR as it does for PFS-SR. Now we investigate whether using *Effective Average* (as defined in

<sup>6</sup>The time passed since the first transmission of a frame to when it is received successfully is minimal.

Eq. 9) will be good solution for PFS-FR as it is for PFS-SR (Section V) and conclude that while using PFS-FR with *Effective Average* makes it immune to retransmission attacks, it exposes the scheduler to a different more dangerous kind of attacks.

Generally, the goal of a malicious user is to maximize the loss to other users in the system, regardless of the impact of his behavior on his own throughput. By performing a retransmissions attack, the malicious user delays the transmission of new frames to all users, including himself. Therefore, it can be proved that while gaining more time share, his throughput decreases. Therefore, while vulnerable to malicious users, PFS-FR (and PFS-SR) with *Admitted Average* is invulnerable to selfish users aiming at maximizing their throughput. Now we show that while PFS-FR with *Effective Average* is immune to malicious retransmission attack (for the same reasons as PFS-SR), it creates a new vulnerability in the scheduler. This vulnerability can be used by both malicious and selfish users to increase their time share and throughput on the expense of others. In PFS-SR with *Effective Average*, low effective rate can cause a user not to win a time slot<sup>7</sup>. In PFS-FR, if a user has a pending retransmission then he is assigned for transmission regardless of his effective rate. That is, a selfish user in PFS-FR with *Effective Average*, can pay less for the retransmissions he receives by reporting a fake<sup>8</sup> low SNR that is mapped to a low effective rate, pay a low price for his retransmissions and hence get larger time share and throughput.

This vulnerability is demonstrated in the following toy example: Assume PFS-FR with *Effective Average* in a system with only two users - a regular user  $U_r$  and a selfish user  $U_s$  such that in reality both users experience a constant SNR value which is mapped to a rate of  $3X \text{ bits/slot}$  and a corresponding effective rate of  $X \text{ bits/slot}$ . That is, the expected number of transmission per frame is 3 for both users but in order to keep the example simple also assume it always takes exactly 3 times. Assume they both joined the system at the same time, their throughput average value was initialized to the same value and that whenever they have an identical priority values the scheduler chooses to transmit to  $U_r$ .  $U_r$  is a regular user who always reports his real SNR, therefore  $\forall t, R_r^e(t) = X$ . The selfish user reports his real SNR value (which is mapped to  $R_r^e(t) = X$ ) only if at time slot  $t$  when he does not have a pending retransmission. At time slot  $t'$  where he has a pending retransmission, he reports a fake low SNR which is mapped to  $R_r^e(t') = \frac{X}{4}$ .

**Lemma 3.** *If at  $t = t_0$  there is no pending retransmission in the system and  $A_r(t_0) = A_s(t_0)$  then: 1. In  $[t_0, t_0 + 8]$   $U_r$  receives one frame successfully while  $U_s$  receives two; 2. In  $t = t_0 + 9$  there is no pending retransmission in the system and  $A_r(t_0 + 9) = A_s(t_0 + 9)$ .*

<sup>7</sup>Reporting fake *high* SNR might allow a user to win a few slots, but since it increases his throughput average it will not benefit him in the long run. Such an attack was already proved to be ineffective in [9].

<sup>8</sup>The idea of users reporting fake channel rate to exploit the properties of channel aware scheduler was already introduced in [6] and [9].

*Proof:* Since  $A_r(t_0) = A_s(t_0) = a$  and  $R_r(t_0) = R_s(t_0) = X$  then  $V_r(t_0) = V_s(t_0)$  hence as assumed earlier,  $U_r$  is assigned with time slot  $t_0$ . Since it takes 3 transmissions until  $U_r$  receives his frame successfully, PFS-FR retransmits his frame also in  $t_0 + 1$  and  $t_0 + 2$ . At time  $t_0 + 3$ , according to the definition of *Effective Average* (Eq. 9) their throughput is as follows:  $A_r(t_0 + 3) = \frac{t_0-1}{t_0+2}a + \frac{3}{t_0+2}X$ ,  $A_s(t_0 + 3) = \frac{t_0-1}{t_0+2}a$  - that is,  $A_r(t_0 + 3) > A_s(t_0 + 3)$  and since  $R_r(t_0 + 3) = R_s(t_0 + 3) = X$  ( $U_s$  has no pending retransmission), then  $V_s(t_0 + 3) > V_r(t_0 + 3)$  and  $U_s$  is assigned with  $t_0 + 3$ . Since it takes 3 retransmissions for  $U_s$  to receive the frame he is assigned with  $t_0 + 4$  and  $t_0 + 5$  as well. As described, the selfish user has different effective rate values when he has pending retransmission. That is,  $R_s^e(t_0 + 3) = X$ ,  $R_s^e(t_0 + 4) = \frac{X}{4}$ ,  $R_s^e(t_0 + 5) = \frac{X}{4}$ . Therefore  $A_s(t_0 + 6) = \frac{t_0-1}{t_0+5}a + \frac{1}{t_0+5}X + \frac{2}{t_0+5}(\frac{X}{4})$  which is smaller than  $A_r(t_0 + 5) = \frac{t_0-1}{t_0+5}a + \frac{3}{t_0+5}X$ . Therefore, as for time slot  $t_0 + 3$ , since  $U_s$  still has smaller average in  $t = t_0 + 6$ , he is assigned for transmission in  $[t_0 + 6, t_0 + 8]$  where he receives another frame successfully. Lastly,  $A_s(t_0 + 9) = \frac{t_0-1}{t_0+8}a + \frac{1}{t_0+8}X + \frac{2}{t_0+8}(\frac{X}{4}) + \frac{1}{t_0+8}X + \frac{2}{t_0+8}(\frac{X}{4}) = \frac{t_0-1}{t_0+8}a + \frac{3}{t_0+8}X = A_r(t_0 + 9)$ . ■

Lemma 3 shows the selfish user receives a double time share and throughput than what the regular user gets. That is, the time share of the regular user is  $\frac{1}{3}$  instead of  $\frac{1}{2}$  in the case where the selfish user was reporting his real channel condition (can be proved in a similar way). A malicious user can create even larger loss by faking low SNR value during retransmissions. Unlike the selfish user, he keeps reporting NACK even when the frame sent to him was received successfully. Hence while *Effective Average* is good for PFS-SR, it leaves PFS-FR vulnerable by allowing also selfish users to harm the system.

#### D. PFS-FR with Initial Effective Average

The problem in combining PFS-FR with effective rate is that the payment for a frame is based also on the effective rates the user reported during his retransmissions. That is, time slots that he wins regardless of what he report. Therefore, there is no incentive for the user to report his real rate. In the following equation we suggest an averaging method called *Initial Effective Average* which avoids this problem:

$$A_i(t + 1) = \frac{t-1}{t}A_i(t) + \frac{1}{t}R_i^e(F(t))1_i^{snd}(t), \quad (10)$$

where  $F(t)$  is the time slot in which the frame (sent at time  $t$ ) was sent for the first time. That is, in this average method, when the system transmits a frame to a user for the first time at time  $t$  - it updates his average with  $R_i^e(t)$ . If the system is required to retransmit the frame at time  $t' > t$  then the average of the user will be updated again with  $R_i^e(t)$  (and not with  $R_i^e(t')$  as in the *Effective Average*). This method prevents malicious and selfish users from benefiting a "reduce payment" by making fake reports during the retransmissions of their frames. The problem with this solution is that it can cause the system to "overcharge" regular users who always experience bad channel rate during retransmissions, and hence their  $A_i(t)$

value might not represent the expected throughput average they actually receive. As explained at the beginning of this section, PFS-FR is typically used in a system where the transmission limit  $L_{max}$  is small and/or the channel rate of the users does not change significantly in the short term. Then the  $A_i(t)$  value calculated by PFS-FR with *Average by Initial Transmission* – equals the one calculated by PFS-FR with *Effective Average*. According to Theorem 7, this average represents the expected throughput of the user, just like the average in PFS-FR with the original averaging method (*Admitted Average*). Hence, it can be proved<sup>9</sup> that users under PFS-FR with *Initial Effective Average* are expected to get the same time share and the same throughput as he would have gotten under PFS-FR with *Admitted Average*. Therefore, we conclude that PFS-FR with *Initial Effective Average* is a fair, immune variation of the PFS-FR in systems where fast retransmissions are preferred.

## VII. SUMMARY

We showed that in practical networks where frame loss are considered standard Proportional Fairness is vulnerable to malicious attacks. We aimed at devising a policy that is immune to such attacks while maintaining fairness. We proposed and analyzed a number of approaches whose derived properties are summarized in the following table:

Average Method	PFS-FR		PFS-SR	
	Fair	Immune	Fair	Immune
Standard PFS (Admitted Avg.)	✓	✗	✓	✗
Transmission Avg.	✗	✓	✗	✓
Effective Avg.	✓	✗	✓	✓
Initial Effective Avg.	✓	✓		

## ACKNOWLEDGMENT

We would like to thank Erez Biton for helpful discussions.

## APPENDIX

**Lemma 4.** For PFS-SR with Admitted Average under CRA,  $V_i(t)$  is independent of the value of  $R_i$  and is given by  $V_i(t) = G_i \frac{t}{T_i(t)}$  where  $T_i(t) = \sum_{s=1}^{t-1} 1_i^{rcv}(s)$  is the number of frames  $U_i$  received during time slots  $[1, t-1]$ .

*Proof:* The rate of  $U_i$  is  $R_i$  hence his throughput average is  $A_i(t) = \frac{T_i(t)}{t} R_i$ . Since  $V_i(t) = \frac{G_i R_i}{A_i(t)}$  (Eq. 5) we get the claimed result. ■

**Lemma 5.** For PFS-SR with Admitted Average under CRA, If  $U_i$  successfully received a frame on time slot  $t$  ( $1_i^{rcv}(t) = 1$ ) then his priority value cannot increase in the next time slot ( $V_i(t) \geq V_i(t+1)$ ). Otherwise ( $1_i^{rcv}(t) = 0$ )  $V_i(t) < V_i(t+1)$ .

*Proof:* From Lemma 4 we get that  $V_i(t) = G_i \frac{t}{T_i(t)}$ . If  $1_i^{rcv}(t) = 1$  then  $T_i(t+1) = T_i(t) + 1$ , therefore  $V_i(t+1) = G_i \frac{t+1}{T_i(t)+1}$ . Since  $\forall_i. t \geq T_i(t)$  then  $V_i(t) \geq V_i(t+1)$ . If  $1_i^{rcv}(t) = 0$  then  $T_i(t+1) = T_i(t)$ , therefore  $V_i(t+1) = G_i \frac{t+1}{T_i(t)}$ , hence  $V_i(t) < V_i(t+1)$  ■

<sup>9</sup>For lack of space the formal proof is not provided here. It can be found in [13].

**Lemma 6.** Let  $U_i$  and  $U_j$  be two users in PFS-SR with Admitted Average under CRA. If  $1_i^{rcv}(t_0) = 1$  and then in the next time slot  $V_j(t_0 + 1) \geq V_i(t_0 + 1)$  – then the proportion between their priority values is limited as follows:  $1 \leq \frac{V_j(t_0+1)}{V_i(t_0+1)} \leq 1 + \frac{1}{T_i(t_0)}$ .

*Proof:* Since  $U_i$  received a frame successfully at time slot  $t_0$ , then  $T_i(t_0 + 1) = T_i(t_0) + 1$  and hence  $V_i(t_0 + 1) = G_i \frac{t_0+1}{T_i(t_0)+1}$  (Lemma 4). In addition, since  $V_i(t_0) \geq V_j(t_0)$  then according to Lemma 4  $G_i \frac{1}{T_i(t_0)} \geq G_j \frac{1}{T_j(t_0)}$ . Therefore  $G_i \frac{t_0+1}{T_i(t_0)} \geq G_j \frac{t_0+1}{T_j(t_0)}$ . And since  $T_j(t_0) = T_j(t_0 + 1)$  we get that  $G_i \frac{t_0+1}{T_i(t_0)} \geq G_j \frac{t_0+1}{T_j(t_0+1)} = V_j(t_0+1)$ . Using the expression for  $V_i(t_0+1)$  together with the inequality for  $V_j(t_0+1)$  we get that  $\frac{V_j(t_0+1)}{V_i(t_0+1)} \leq (G_j \frac{t_0+1}{T_j(t_0+1)}) / (G_i \frac{t_0+1}{T_i(t_0)+1}) = 1 + \frac{1}{T_i(t_0)}$ . Lastly, since  $V_j(t_0 + 1) \geq V_i(t_0 + 1)$  then obviously  $1 \leq \frac{V_j(t_0+1)}{V_i(t_0+1)}$ . ■

**Lemma 7.** Under the steady state of PFS-SR under CRA, the proportion  $G_i/q_i(t)$  of all users converges to the same value where  $q_i(t) = (\sum_{s=1}^{t-1} 1_i^{rcv}(s)) / (t-1)$  is the fraction of time slots  $[0, t-1]$  which were assigned to  $U_i$  and in which he reported to receive them successfully.

*Proof:* The value of  $A_i(t)$  is updated with  $R_i$  after every time slot in which  $U_i$  receives a frame successfully. Therefore,  $A_i(t) = q_i(t) R_i$ . Since  $R_i^e = G_i R_i$ , we get that  $V_i(t) = G_i / q_i(t)$ . Then, under the steady state the proportion  $G_i / q_i(t)$  of every user converges to the same value (Theorem 1). ■

## REFERENCES

- [1] R. P. A. Jalali and R. Pankaj, "Data throughput of CDMA-HDR: A high efficiency high data rate personal communication wireless system," in *Proceedings of the IEEE Vehicular Technology Conference*, 2000.
- [2] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 636–647, 2005.
- [3] Y. H. Wang Yafeng, "Retransmission priority scheduling algorithm for forward link packet data service," in *International Conference on Communication Technology (ICCT)*, 2003.
- [4] P. D. Systems, H. Zheng, H. Viswanathan, and S. Member, "IEEE transactions on wireless communications, vol. 4, no. 2, march 2005 495 optimizing the ARQ performance in downlink," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 495–506, 2005.
- [5] W. W. Mugen Peng, "Advanced HARQ and scheduler schemes in TDD-CDMA HSDPA systems," in *The Joint Conference of the 10th Asia-Pacific Conference on Communications*, 2004.
- [6] U. Ben-Porat, A. Bremler-Barr, and H. Levy, "On the exploitation of CDF based wireless scheduling," in *INFOCOM*, pp. 2821–2825, 2009.
- [7] D. Park, H. Seo, H. Kwon, and B. G. Lee, "Wireless packet scheduling based on the cumulative distribution function of user transmission rates," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1919–1929, 2005.
- [8] S. Bali, S. Machiraju, H. Zang, and V. Frost, "A measurement study of scheduler-based attacks in 3G wireless networks," in *PAM*, pp. 105–114, 2007.
- [9] H. C. Radmilo Racic, Denys Ma and X. Liu, "Exploiting opportunistic scheduling in cellular data network," in *NDSS*, 2008.
- [10] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.
- [11] P. A. W. Harold J. Kushner, "Asymptotic properties of proportional-fair sharing algorithms," *Proc. 40th Annual Allerton Conf.*, pp. 1051–1059.
- [12] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1250–1259, 2003.
- [13] U. Ben-Porat, A. Bremler-Barr, H. Levy, and B. Plattner, "On the vulnerability of the proportional fairness scheduler to retransmission attacks," 2010. [http://people.ee.ethz.ch/~ehudb/PFS\\_Ret\\_Technical\\_Report.pdf](http://people.ee.ethz.ch/~ehudb/PFS_Ret_Technical_Report.pdf).