Beyond Proportional Fair: Designing Robust Wireless Schedulers

Soshant Bali[1], Sridhar Machiraju[2], and Hui Zang[2]

[1] University of Kansas sbali@ittc.uk.edu
[2] Sprint ATL {Machiraju,Hui.Zang}@sprint.com

Abstract. Proportional Fair (PF), a frequently used scheduling algorithm in 3G wireless networks, can unnecessarily starve "well-behaved" users in practice. One of the main causes behind PF-induced starvation is its inability to distinguish between users who are backlogged and users who are not. In this paper, we describe how a simple parallel PF instance can mitigate such starvation.

1 Introduction

Scheduling algorithms play a key role in deciding user performance in wireless networks. In the past, a number of *channel-aware* scheduling algorithms have been proposed [1–4] to exploit the time-varying nature of user channel conditions without sacrificing fairness. The Proportional Fair (PF) algorithm [5] is one such channel-aware algorithm that has been widely deployed in cellular data networks, especially in 3G networks such as CDMA-based EV-DO [5] and GSM-based HSDPA networks.

Our work is motivated by the observation that PF, though widely used, displays a surprising lack of robustness that causes it to unnecessarily starve well-behaved users for significant periods of time. One of the main reasons behind such starvation is the inability of PF to distinguish between users who are backlogged and users who are not. In particular, as we showed in prior work [6], a user receiving maliciously crafted "on-off" flows can starve other users.

We analyze this problem and design Parallel PF (PPF), i.e., PF with a parallel PF instance, to eliminate such starvation. Using simulations, we show that PPF is superior to PF - it is robust and achieves comparable or better throughput and fairness than PF. The PF-centric nature of our work is justified for two reasons. First, a robust alternative to PF is of practical interest given the widespread use of PF. Second, the use of a parallel mechanism is fairly general and can be used with non-PF schedulers (see [7]). Previously proposed algorithms are not suitable for one or more reasons. For instance, solutions using a strict time-based threshold [8] to limit starvation do not distinguish between the starvation of a well-behaved user experiencing "good" channel conditions - an undesirable outcome - from the starvation of a user experiencing fading - a desirable outcome. Approaches using some form of delay-throughput tradeoff [3] do not prevent vulnerability (of TCP and UDP flows) to malicious "on-off" flows and, also, compromise on channel-awareness (for instance, see [9]).

2 The PF Algorithm and Starvation

The PF algorithm typically transmits data to end-users (Access Terminals or ATs) in time slots of fixed size, for example, 1.67ms in EV-DO networks. Each AT has its own queue at the base station. Being channel-aware, PF uses the current *achievable rate* (of downlink transmission) to each AT. This rate is reported on a per-slot basis by all active ATs. In the EV-DO system, there are 10 unique achievable data rates [10, 5]. Assume that there are n ATs in the system. Denote the achievable data rate reported by user i in time slot t to be $R_i[t]$ $(i = 1 \ldots n)$. PF maintains $A_i[t]$, the exponentially-weighted *mean achieved rate*.

$$A_i[t+1] = \begin{cases} A_i[t](1-\alpha) + \alpha R_i[t] & \text{if } t \text{ allocated to } i \\ A_i[t](1-\alpha) & \text{otherwise} \end{cases}$$

PF allocates the slot of time instant t to the AT with the highest $\frac{R_i[t]}{A_i[t]}$ ratio. Typically, $\alpha = \frac{1}{T}$ is chosen to be 0.001 [5, 6]. This ensures that $A_i[t]$ remains roughly constant over time under stationary channel conditions. Under reasonably general conditions, PF maximizes the sum of the logarithms of per-AT throughput [11] when the channel conditions of ATs are independent. Moreover, if their channel conditions are identically distributed, PF ensures that all ATs are allocated equal number of slots in the long term.

In practice, PF can starve an AT experiencing *good* conditions for a large period of time thereby causing serious performance degradation in the form of high delay jitter, spurious TCP timeouts, etc [6]. Specifically, PF would allocate an AT, which has a much larger $\frac{R[t]}{A[t]}$ ratio than other ATs, consecutive slots until the ratios cross-over. It turns out that an AT can have its $\frac{R[t]}{A[t]}$ ratio grow arbitrarily large simply by not receiving any traffic. To see why, note that $\frac{R[t]}{A[t]}$ is inversely proportional to $A[t]$ and recall that $A_i[\cdot]$ reduces (via multiplication with $1-\alpha$) when AT i is not allocated slot t *irrespective of whether AT i is backlogged or not.*

Hence, a malicious AT receiving traffic in an "on-off" pattern could potentially starve other ATs during the beginning of each "on" period. Recently [6], we used rigorous experiments on commercial and laboratory networks as well as analysis to show that the resulting starvation can increase jitter and reduce TCP goodput (due to spurious timeouts) of a competing AT by up to 1 second and 30% respectively. We plot a representative result showing the reduction in TCP goodput for various "off" durations in Figure 1.

3 Proposed Solution

PF is vulnerable to "on-off" traffic primarily because it reduces $A[t]$ (by multiplying it with $1-\alpha$) even when an AT is not backlogged. A naive solution is to freeze the value of $A[t]$ for such ATs. But, a frozen $A[t]$ value does not adapt to changes in the number of backlogged ATs or channel conditions. Hence, an AT

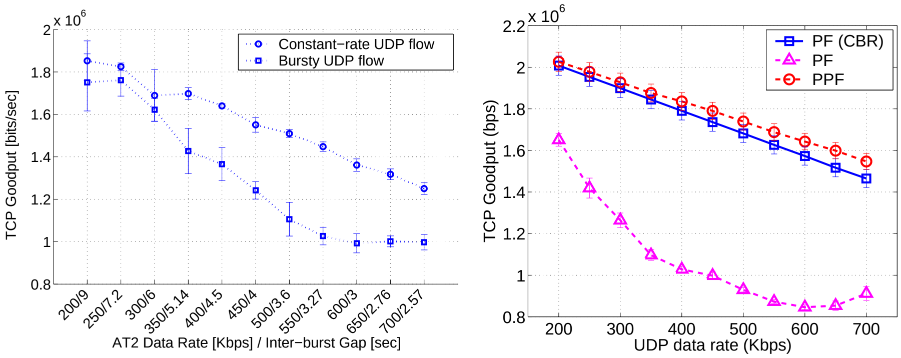

Fig. 1. (Left) Comparison of the (experimental) TCP goodput to an AT when another AT receives (1) A periodic (UDP) packet stream. (2) An "on-off" UDP flow with various inter-burst times. TCP Goodput can decrease by up to 30% due to "on-off" flows. **(Right)** Similar simulation experiments with PF and PPF. The inter-burst times decreased from 9s to 2.57s. Goodput decrease due to PF is similar to that seen experimentally but higher due to differences in TCP timeout algorithms in *ns-2* and practical implementations. Goodput reduction is eliminated with PPF.

with a recently unfrozen $A[t]$ can have a ratio that is much lower or higher than other ATs thereby causing starvation. A backlog-unaware algorithm, which always considers ATs to be backlogged, is also not desirable since it would allocate slots to ATs with no data to receive and hence, would not be work conserving.

We propose the following *Parallel PF (PPF)* algorithm that uses a backlog-unaware scheduler instance only to remove the undue advantage an "on-off" user receives at the beginning of "on" periods. A normal instance of PF drives slot allocation. The parallel instance of PF assumes all ATs are backlogged and executes simultaneously. We use $A^p[t]$ to refer to the $A[t]$ values maintained by the parallel instance. When a previously idle AT becomes backlogged, all $A[t]$ values are reset to the corresponding $A^p[t]$ values. Thus, when a previously idle AT becomes backlogged, differences in achieved throughput of backlogged and idle ATs are forgotten. Also, notice that as long as an idle AT does not become backlogged, PPF is equivalent to PF.

To test if PPF is vulnerable to "on-off" traffic patterns, we recreated our laboratory-based setup (see [6] and Figure 1(Left)) using *ns-2* simulations with two ATs - AT1 and AT2. AT1 received a long-lived TCP flow and AT2 received a (malicious) "on-off" UDP flow consisting of 225KB bursts sent at various inter-burst time periods. The simulations used a wireless link, governed by PF or PPF, that connected to a wired 100Mbps link with mean round trip times of 250ms. We assigned achievable data rates based on measurements in a commercial EV-DO network. To collect these 30-minute long traces, we used the Qualcomm CDMA Air Interface Tester [12] software on a stationary AT and a mobile AT moving at an average speed of 40mph. We plot the TCP goodput obtained with PF and PPF in Figure 1 (Right). The results clearly show that, with PPF, the

TCP goodput is not affected by the "on-off" flow. In fact, by not causing the UDP flow to have low $A[t]$ values, PPF better implements channel-awareness. This is the reason why the TCP goodput with PPF is slightly higher than the goodput with PF and a CBR flow consisting of periodically-sent UDP packets.

4 Conclusions and Future Work

In this paper, we focused on the starvation induced by PF due to "on-off" traffic and showed that a parallel PF instance can eliminate such starvation. The parallel instance is a fairly general mechanism that can be used with non-PF schedulers. Indeed, in the longer version of this paper [7], we designed a quantile-based scheduling algorithm that is more robust than PF under other starvation-inducing scenarios and showed that a parallel instance endows it with robustness to "on-off" traffic too.

References

1. Bonald, T.: A Score-based Opportunistic Scheduler for Fading Radio Channels. In: Proc. of European Wireless. (2004)
2. Viswanath, P., Tse, D., , Laroia, R.: Opportunistic Beamforming using Dumb Antennas. IEEE Transactions on Information Theory **48** (June 2002) 1277–1294
3. Shakkottai, S., Stolyar, A.: Scheduling Algorithms for a Mixture of Real-time and Non-real-time Data in HDR. In: Proc. of ITC-17. (September 2001)
4. Borst, S., Whiting, P.: Dynamic Rate Control Algorithms for HDR Throughput Optimization. In: Proc. of INFOCOM. (2001)
5. Jalali, A., Padovani, R., Pankaj, R.: Data Throughput of CDMA-HDR: A High Efficiency-high Data Rate Personal Communication Wireless System. Proc. of IEEE Vehicular Technology Conference **3** (May 2000) 1854–1858
6. Bali, S., Machiraju, S., Zang, H., Frost, V.: A Measurement Study of Scheduler-based Attacks in 3G Wireless Networks. In: Proc. of Passive and Active Measurement (PAM) Conference. (2007)
7. Bali, S., Machiraju, S., Zang, H., Frost, V.: On the Performance Implications of Proportional Fairness (PF) in 3G Wireless Networks. Technical Report RR06-ATL-040624, Sprint ATL (2006)
8. Park, D., Seo, H., Kwon, H., Lee, B.G.: A New Wireless Packet Scheduling Algorithm based on the CDF of User Transmission Rates. In: Proc. of GLOBECOM. (2003)
9. Patil, S., de Veciana, G.: Measurement-based Opportunistic Feedback and Scheduling for Wireless Systems. In: Proc. of Annual Allerton Conference on Communication, Control and Computing. (2005)
10. Bender, P., Black, P., Grob, M., Padovani, R., Sindhushayana, N., Viterbi, A.: CDMA/HDR: A Bandwidth-efficient High-speed Wireless Data Service for Nomadic Users. IEEE Communications Magazine**38** (July 2000) 70–77
11. Kushner, H., Whiting, P.: Convergence of Proportional-Fair Sharing Algorithms Under General Conditions. IEEE Trans. on Wireless Communications (july 2004)
12. CDMA Air Interface Tester: `http://www.cdmatech.com/download\_library/pdf/CAIT.pdf` (2006)