TCP START-UP BEHAVIOR UNDER THE PROPORTIONAL FAIR SCHEDULING POLICY*

J. H. CHOI †, J. G. CHOI ‡, AND C. YOO †

† *Department of Computer Science and Engineering*
Korea University
Seoul, Korea
E-mail: {jhchoi, hxy}@os.korea.ac.kr

‡ *KT Convergence Laboratory*
Seoul, Korea
E-mail: jinchoi@kt.co.kr

It is expected that the proportional fair (PF) scheduler will be used widely in cdma2000 1xEV-DO systems because it maximizes the sum of each user's utility, which is given by the logarithm of its average throughput. However, in terms of short-term average throughput, PF scheduler may lead to a large RTT variation. We analyze the impact of PF scheduler on TCP start-up behavior through NS-2 simulation. To show the impact of PF scheduling on TCP, we also analyze the packet transmission delay under the PF scheduling policy through mathematical model.

1. Introduction

Recent advances in communication technology make appearance of the packet-based cellular systems such as cdma2000 1xEV-DO [1] and UMTS-HSDPA [2]. Being mainly targeted on high-speed data applications that are tolerant of some packet delay, it is reasonable that their schedulers focus on maximizing the sum of each user's utility. A good way of achieving it is to serve the users with good channel condition first utilizing the time-varying feature of wireless channels. This approach increases the system throughput significantly. But, some users can be sacrificed since, in wireless environment, users have very different channel condition according to their location.

*This work is supported by grant no.r01-2004-000-10588-0 from the basic research program of the korea science and engineering foundation.

The proportional fair scheduler [3] is one of the most promising opportunistic schemes balancing system throughput and user fairness. It is very simple to implement, and also it is optimal in the sense of maximizing the sum of each user's utility that is given by the logarithm of average throughput for elastic traffic. However, it does not provide fair service in short-term scale even if it achieves the proportional fairness in long-term. In addition, owing to its reflection on channel state, the scheduler induces some variation on scheduling delay. So, PF scheduler can make a serious influence on the performance of TCP since it uses path delay or round trip time (RTT) to estimate network state.

Previous researches [6] [7] [8] have already addressed the issue of TCP performance in cellular networks. However, to our best knowledge, it is not investigated how the opportunistic scheduler like PF one affects the behavior of TCP. In this paper, we focus on the impact of PF scheduler on TCP excluding other reasons such as handoff, packet error, and so on.

This paper is organized as follows. The following section provides the description of our target system and background knowledge. In Section 3, we introduce the simulation environments and show how the TCP fairness and throughput is affected by PF scheduler. Section 4 presents our analysis for TCP start-up behavior under PF scheduler. Finally, in Section 5, we demonstrate our next plan and conclude the paper.

2. System Model and Background

2.1. *System model description*

We consider the downlink channel of cellular networks where a BS serves N mobile terminals. The downlink is a single broadband channel shared by all users in the time division multiplexing manner[a]. The BS exploits the pilot signal, which is pre-defined by the protocol, in the specified position of each time slot, and every mobile measures it to obtain the channel gain. The BS receives the fed back signal from all the users to collect the current channel status. Based on the channel information, the radio frequency scheduler selects a user among the active ones to be served in the next slot.

2.2. *PF scheduler*

Let us examine the operation of the PF scheduler. The average throughput of each user is tracked by an exponential moving average. At the beginning

[a]The downlink structure is very similar to that of the IS-856 system.

of each time slot, each user feeds back the channel state (or the feasible rate) to the BS. The BS calculates the ratio of the feasible rate to the average throughput for each user, which is defined as the preference metric and is the key selection criterion. The user with the maximum preference metric will be selected for transmission at the next coming slot.

This is described formally as follows. In time slot n, the feasible rate of user k is $R_k[n]$ and its moving average is denoted by $\tilde{R}_k[n]$. Then, user $k^* = argmax_k(\frac{R_k[n]}{\tilde{R}_k[n]})$ is served in time slot n, and the average throughput of each user is updated by

$$\tilde{R}_k[n+1] = \begin{cases} \left(1-\frac{1}{t_c}\right)\tilde{R}_k[n] + \frac{1}{t_c}R_k[n], & k = k^* \\ \left(1-\frac{1}{t_c}\right)\tilde{R}_k[n], & k \neq k^* \end{cases} \quad (1)$$

where t_c is the time constant for the moving average. It is clear that the PF scheduler affects relative preference to users with good channels as opposed to absolute preference.

3. Observation and Problems

We are easily able to guess that PF scheduler makes an influence on RTT variation of TCP, but the key problem is to investigate the degree of such impact. If the impact does not severely harm TCP, mere influence of RTT variation could be ignored. But if TCP misbehavior results from the variation can lead to a serious performance drop, we have to find the core condition to make the wrong action.

3.1. *Simulation environments*

The simulation study is performed by NS-2 2.27 version, and topology is a typical cellular network shown in Fig.1. There is a link that has 2Mbps bandwidth and 100ms latency between MH (Mobile Host) and BS (Base Station). Also, a wired link has 10Mbps bandwidth and 50ms latency, and it is placed between BS and CH (Corresponding Host). The queue between MH and BS uses PF scheduler, and the other uses FIFO (First-In-First-Out) policy.

3.2. *Observation: TCP startup behavior*

From the analysis of the simulation results, TCP's RTO mechanism generally keeps track of RTT variation well in most cases except timeouts.

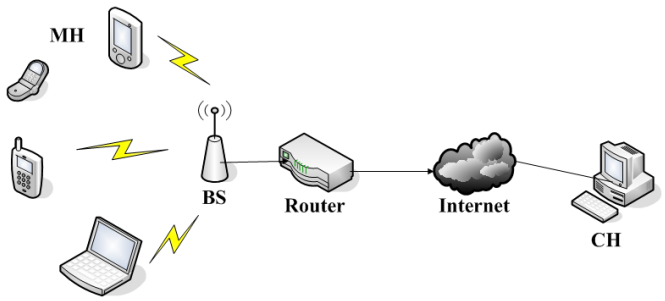


Figure 1. Simulation topology.

Specially, our observation says that timeout event often happens in slow start phase as shown in Fig.2.

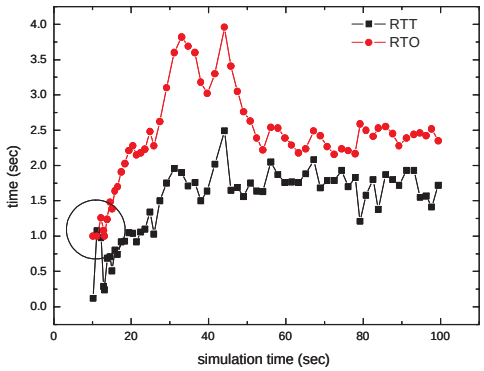
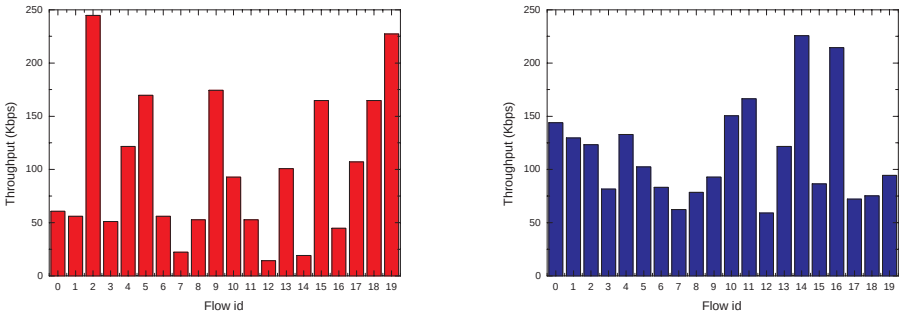


Figure 2. Initial timeout.

Due to the timeout in the slow start phase, TCP sessions cannot keep fairness in short-term average throughput, and this unfairness is in a serious degree (see Fig.3) even if we allow that PF scheduler does not consider short-term traffic. A good way to look into a problem of the initial timeout is to compare the timeout case with the non-timeout case under the same condition. Fig.3 shows well the effect of the initial timeout on TCP fairness and aggregate throughput. In Fig.3(a), we can see that the sessions with

the initial timeout are not nearly able to use the network bandwidth (in 11 of 20 sessions, the timeout occurred), and the others aggressively use the remains so the fairness comes to harm severely. In Fig.3(b), it is observed that both the aggregate throughput and the fairness are improved by just preventing the unnecessary timeout from happening. That is, according to Jain's fairness index [4], the experiment in Fig.3(a) shows the fairness of 68.5%, and the study in Fig.3(b) shows the fairness of 86.2%. Also, in point of view of the aggregate throughput, Fig.3(b) obtains the throughput of 2296 Kbps while Fig.3(a) gets the throughput of 1998.4 Kbps. Fig.5



(a) Without holding timeout (b) With holding timeout

Figure 3. Throughput of 20 flows under PF scheduler in short-term.

draws the results of the fairness and the aggregate throughput varying the number of sessions from 20 to 50. Comparing normal case with holding-timeout case, we can see that both the fairness and the throughput shows better performance in the holding-timeout case independent of the number of flows.

However, the case of Fig.3(b) also shows considerably lower fairness than that of Fig.4, in which FIFO is used in queuing policy (FIFO case got the 99.09% fairness). The performance gap of Fig.3(b) and Fig.4 could be understood as the result from the difference of the scheduling policies since we force to hold the initial timeout in Fig.3(b).

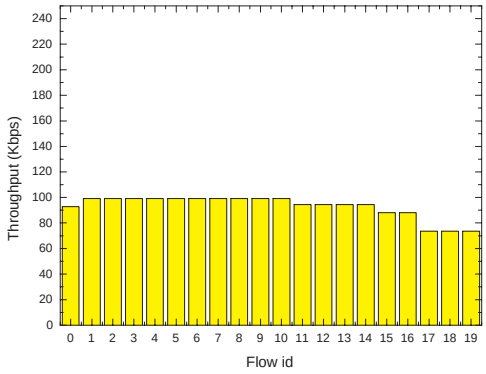
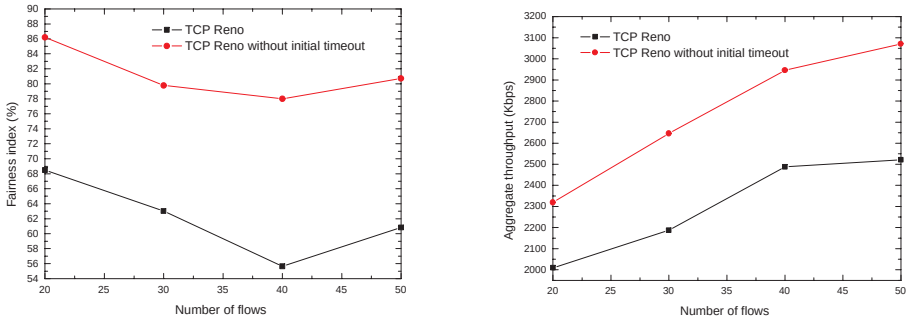


Figure 4. Throughput of 20 flows under FIFO scheduler in short-term.



(a) Fairness index (b) Aggregate throughput

Figure 5. Comparisons depending on the number of flows.

4. Analysis

4.1. *Condition of TCP timeout*

TCP retransmission timer expires when an ACK packet is not returned during RTO interval. So, we can describe the condition of the timeout occurrence as follows.

$$RTO < RTT = Q.D. + P.D., \tag{2}$$

where $Q.D.$ denotes queuing delay and $P.D.$ is propagation delay in all links. $Q.D.$ is the sum of queuing delays in wired and wireless link. We assume that the queuing delay in the wired link is steady, and PF scheduling delay is dominant factor in $Q.D.$.

TCP starts data transmission by setting the initial RTO to 6 seconds, and hereafter updates the RTO value, referencing the measured RTT. Because TCP begins the slow start phase sending a few segments (generally 2 segments), RTT is relatively small in the initial phase (when there is not much packets to be scheduled, usually the queuing and scheduling delay decrease). Based on this measurement, TCP usually sets RTO to MIN RTO that is 1 seconds according to RFC 2988 [5]4.

After this, RTT increases when the amount of data increases by TCP or other traffic. The problem is that the scheduling delay of PF scheduler leads to rapid RTT increase, which causes TCP's timer expired. Because the scheduling interval of PF scheduler is 1.667 ms, if a channel is not allocated to during n turns, $1.667 \times n$ ms is added to TCP session's RTT in the flow. Supposing that this additional time gets RTT over RTO, the sender will expire the retransmission timer. In our experiments, the timeout usually happens when RTT is over 1 second in the slow start phase. The reason of the boundary is that TCP rounds the calculation value below MIN RTO up to MIN RTO second.

4.2. *Analysis of the packet transmission delay in the cellular networks*

Above one second RTT is not rare in the wireless networks. The wireless networks sometimes have a quite long delay because a base station may have many tasks to reduce the impact of the errors such as Forward Error Correction (FEC), interleaving, retransmission, and so on. In this section, we show an analytic model for BS delay, which is simplified with only the scheduling delay and the retransmission delay.

First, we build the model with 1 user. The user has the packet size of T bytes and is able to transmit X bytes whenever the scheduling slot is allocated. For example, if T is 1500 and X is 100 in constant, the time for servicing the packet is 15 slots. But when X changes depending on the channel state, the analysis comes to be more difficult.

For convenience of the analysis, we assume that X has an exponential distribution with average m (actually this assumption is exactly correct when the transmission rate linearly increases in proportion as SNR (Signal-to-Noise Ratio) on the Rayleigh channel). When we denote the data size that is successfully transmitted in flow i as X_i, the number of slots that is required to service the packet is $N(T)$. $N(T)$ is minimum N that satisfies $\sum_{i=1}^{N} X_i \geq T$. Analyzing this problem as the Poisson counting process, we can see that $N(T) - 1$ has a Poisson distribution with both average and variance $\frac{T}{m}$.

We obtain the required number of slots to service a packet as above. However, owing to wireless channel error, the transmission does not always make a success even if BS successfully transmits the packet. In this model, we denote the error rate of each flow as p and assume that the error rate is independent of the transmission rate. At this time, to transmit the packet successfully in flow i, actually Y_i slots are taken. Because Y_i follows the discrete probability distribution with $Pr(Y_i = n) = p^{n-1}(1-p)$, we get $E(Y_i) = \frac{1}{1-p}$ and $Var(Y_i) = \frac{p}{(1-p)^2}$.

Actual number of slots to transmit a packet is given by $S = \sum_{i=1}^{N}(T)Y_i$, and we can obtain its average and variance as follows.

$$E(S) = E\{N(T)\}E(Y_i) = (\frac{T}{m} + 1)(1-p)^{-1}, \tag{3}$$

$$\begin{aligned} Var(S) &= E\{N(T)\}Var(Y_i) + E^2 Y_i Var\{N(T)\} \\ &= (\tfrac{T}{m} + 1)p(1-p)^{-2} + (1-p)^{-2}(\tfrac{T}{m} + 1) \\ &= (\tfrac{T}{m} + 1)(1+p)(1-p)^{-2} \end{aligned} \tag{4}$$

Let's consider the case of K users. We assume that each user has a packet to transmit, and the packet size, T, and channel state are same in every user. Also assuming that the scheduler chooses a user and, only after transmitting the user's one packet, selects another user, we analyze the packet transmission time of the last-selected user. When the transmission time of $k-th$ selected user is denoted as D_k, our finding time is $D = D_1 + D_2 + ... + D_k = \sum_{k=1}^{K} D_k$. By applying Central limit theorem, we approximate D to a Gaussian distribution with the average $K \cdot E(D_k)$ and the variance $K \cdot Var(D_k)$.

$$\begin{aligned} E(D_k) &= E(S) = (\tfrac{T}{m} + 1)(1-p)^{-1}, \\ Var(D_k) &= Var(S) = (\tfrac{T}{m} + 1)(1+p)(1-p)^{-2}, \end{aligned} \tag{5}$$

Finally D follows the Gaussian distribution with the average $K(\frac{T}{m} + 1)(1-p)^{-1}$ and the variance $K(\frac{T}{m} + 1)(1+p)(1-p)^{-2}$.

For example, when we consider the case of T=1500, m=100, K=50, and p=0.1, the packet transmission time of the last selected user is as follows[b]. It is necessary to keep in mind that the inter-packet interval of a user comes from the scheduling delay.

- Constant rate with no channel error: 750 slots.
- Variable rate with no channel error: 800 slots with 50% and 847 slots with 5%.
- Variable rate with channel error, p: 889 slots with 50% and 943 slots with 5%.

In this model, the scheduler services a user's packet sequentially but real PF scheduler services several users' packet little by little, depending on the channel state. Thus, every user finishes the packet transmission at similar time due to their mixed service time while the average rate of the allocated slots is so high as to have a better possibility that reduces the entire transmission time. Consequently every user has a similar finish time with the "last" user.

4.3. *Condition of the initial timeout under PF scheduler*

In previous subsection, we showed the possibility that BS delay leads to TCP timeout, but above MIN RTO RTT does not always make the timeout. Although RTT comes to over 1 second, the timeout occurs only if RTO in previous turn should be 1 second which is MIN RTO. Such type of the timeout event does not happen often because several long RTTs make also RTO large (As we already mention it, TCP's RTO mechanism keeps track of the network state in most cases). That is, the condition of the initial timeout is not just long RTT but also long relative gap of consecutive RTTs.

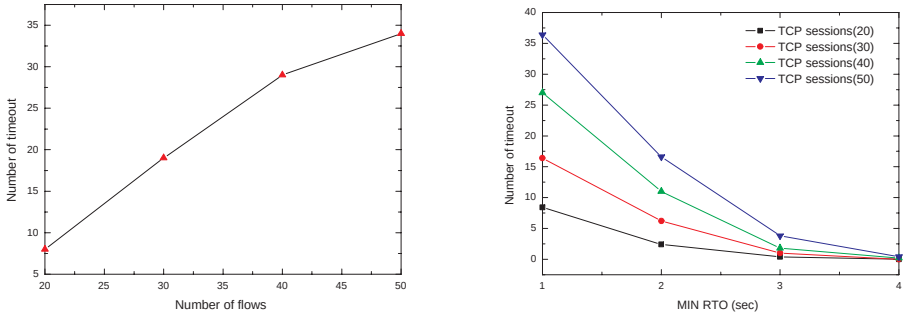
From the result, the condition of the initial timeout can be summarized as follows.

- RTT in previous turn has to make RTO value small enough.
- RTT in next turn increases sharply, and consequently it should be over the RTO value.

[b]Note that one slot takes 1.667 ms.

4.4. *Tendency analysis*

Such initial timeout happened often as the number of session increases. Fig.6(a) draws the number of the initial timeouts varying the number of sessions. As shown in Fig.6(a), we can observe that the sacrificed sessions also increase as the number of sessions increase, and as a result RTT comes to over RTO in several sessions. Also, it is observed that increasing the number of flows leads to consecutive timeout in the slow start phase even if the event does not happen frequently.



(a) Number of timeout depending on number of flows (b) MIN RTO value and the number of flows

Figure 6. Tendency of the timeout number and MIN RTO value.

The timeout does not occur at all TCP sessions, and it is because the slot count for each TCP session can be extremely different in PF scheduler though other delaying factors are similar in each flow. Since PF scheduler preferentially allocates the slot to good channel depending on the channel state, the allocation may be converged on the specific session from the point of view on the short-term. Also, the scheduler may almost not assign the slot to some flows during considerable period.

TCP's initial timeout occurs more frequently, for minimum RTO is briefly set to 1 second. We could see that as MIN RTO increases, the number of the initial timeout in the slow start phase remarkably decreases (refer to Fig.6(b)). Both the throughput and the fairness achieve conspicuous improvement in the performance as the number of the initial timeout decreases, and we can see that in Fig.5. But we cannot say that increasing

MIN RTO is not best choice since TCP reacts too slowly in handling the segment loss if the value is large. In addition, it is a difficult problem to propose any specific value as a recommended MIN RTO, as the suitable value for the various network environments is different according to circumstances. However, our a clear opinion is that current TCP's MIN RTO is set too short under the PF scheduling policy, and the value had better be larger.

5. Conclusion and Future work

In this paper, we analyzed the impact of PF scheduler on TCP start-up behavior. We introduced the initial timeout in the slow start phase from PF scheduling and demonstrated the cause and effect of the timeout through the simulation experiments. From the simulation results and their analysis, a conclusion is that TCP's MIN RTO is so short as to lead to non-necessary timeout under the PF scheduling policy. We do not recommend any specific MIN RTO value in this paper, and the reason is that a fitful MIN RTO is so different depending on the network environments that determining the value is a difficult problem. However, we plan to investigate the method to determine a proper MIN RTO for a variety of network environments.

References

1. Qi Bi, S. Vitebsky, *Performance analysis of 3G-1X EVDO high data rate system*, **Proc. of IEEE WCNC**, March(2002).
2. M. Assaad, B. Jouaber, and D. Zeohlache, *Effect of TCP on UMTS-HSDPA system performance and capacity*, bf Proc. of IEEE GLOBECOM, November(2004).
3. F. Kelly, *Charging and Rate Control for Elastic Traffic*, **European Transactions on Telecommunications**, volume 8 (1997) pages 33-37.
4. R. Jain, W. Hawe, D. Chiu, *A Quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems*, **DEC-TR-301**, September(1984).
5. V. Paxson, M. Allman, *Computing TCP's Retransmission Timer*, **RFC 2988**, November(2000).
6. Khafizov, F. and M. Yavuz, *Running TCP over IS-2000*, **Proc. of IEEE ICC**, April(2002).
7. Yavuz, M. and F. Khafizov, *TCP over Wireless Links with Variable Bandwidth*, **Proc. of IEEE VTC**, September(2002).
8. H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, F. Khafizov, *TCP over Second (2.5G) and Third (3G) Generation Wireless Networks*, **RFC 3481**, February(2003).