PGPS Servers with Time Varying Capacities

Yang-Chung Tseng and Cheng-Shang Chang, *Senior Member, IEEE*

Abstract— **In this paper, we consider a server with a time varying capacity. We show that the virtual time implementation of the PGPS scheme in Parekh and Gallager [5] does not yield the right approximation of the GPS scheme in such an environment. Via the concept of change of time, we observe that a server with a time varying capacity can be viewed as a server with a constant capacity under an appropriate change of clock. This leads to a modified version of the virtual time implementation and various bounds of the PGPS server with a time varying capacity.**

Index Terms— **Generalized processor sharing, change of time**

I. Introduction

In the queueing literature, Processor Sharing (PS) is a well known scheduling policy under which each job in a processor receives an equal amount of processor power. A generalized version of PS, called Generalized Processor Sharing (GPS), assigns each job a weight so that the amount of processor power received by a job is proportional to its weight. To apply GPS to a processor subject to different arrival streams, one allocates each arrival stream an infinite buffer and only the first job in each buffer receives the weighted processor power.

Following the terminology in telecommunication networks, in this paper we use a *session* for an arrival stream, a *packet* for a job, and a *server* for a processor. To formalize the GPS scheme, assume the server operates at a fixed rate 1 and there are N sessions. The i^{th} session is assigned a positive weight ϕ_i. A session is said to be backlogged at time t if there are packets of that session queued in the buffer at time t. Let $B(t)$ be the set of backlogged sessions at time t and $r_i(t)$ be the service rate received by the i^{th} session at time t. Then

$$r_i(t) = \begin{cases} \phi_i / \sum_{j\in B(t)} \phi_j & \text{if } i \in B(t) \\ 0 & \text{otherwise} \end{cases}.$$

GPS is work conserving, i.e., the server is busy whenever there are backlogged packets. In this paper, we denote a *busy period* as the period of time that there are backlogged packets.

Let $S_i(\tau,t) = \int_\tau^t r_i(s)ds$ be the amount of service received by the i^{th} session in the interval $[\tau,t]$. Then for any session i that is continuously backlogged in the interval $[\tau,t]$, GPS guarantees that

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} \geq \frac{\phi_i}{\phi_j}, j = 1,2,\ldots,N. \tag{1}$$

This in turns implies

$$S_i(\tau,t) \geq \frac{\phi_i}{\sum_{j=1}^N \phi_j}(t-\tau) \tag{2}$$

for any session i that is continuously backlogged in the interval $[\tau,t]$

GPS is an ideal scheduling policy that assumes packets are infinitely divisible. As packets are in practice *not* divisible, Parekh and Gallager [5] presented a packet-by-packet transmission scheme to approximate GPS, called Packetized Generalized Processor Sharing (PGPS). The basic rule of the PGPS scheme is as follows: *Whenever the server is ready to serve packets, it picks the one that would complete its service in the GPS simulation first among all packets in the queue if there were no packet arrivals after that time.*

To implement such a rule, they used the concept of virtual time to track the progress of the GPS server. Denote each arrival and departure from the GPS server as an event. Let t_j be the time at which the j^{th} event occurs. Denote the time of the first arrival in a busy period as $t_1 = 0$, and let B_j be the set of sessions that are busy in the interval (t_{j-1}, t_j). Then the virtual time $V(t)$ in a busy period evolves as follows:

$$\begin{aligned} &V(t) = 0, \ \text{ whenever } t \text{ is not in a busy period.} \\ &V(t_{j-1}+\tau) = V(t_{j-1}) + \frac{\tau}{\sum_{i\in B_j}\phi_i}, \\ &\qquad \tau \leq t_j - t_{j-1}, j = 2,3,\ldots \end{aligned} \tag{3}$$

Let a_i^k be the k^{th} arrival time from the i^{th} session and L_i^k be its packet size. Define the virtual starting times and the virtual finishing times of packets from the i^{th} session as follows:

$$\begin{aligned} S_i^k &= \max[F_i^{k-1}, V(a_i^k)], \ \text{ with } F_i^0 = 0 \text{ for all } i. \\ F_i^k &= S_i^k + \frac{L_i^k}{\phi_i}. \end{aligned}$$

To update the virtual time between two successive events, define $Next(t)$ to be the time of the next departure event after time t if there were no more arrivals after time t. Thus, $Next(t)$ is the next time to update $V(t)$ after t if there were no more arrivals after time t. Let F_{min} be the smallest virtual finishing time of a packet in the server at time t. Then,

$$Next(t) = t + (F_{min} - V(t)) \sum_{t\in B_j} \phi_i. \tag{4}$$

Given the mechanism for updating virtual time, Parekh and Gallager [5] proposed the virtual time implementation

EDICS number: C.L. 1.8.3, Multiple Access Protocols

Corresponding author. Cheng-Shang Chang, Dept. of Electrical Engineering, National Tsing Hua University, Hsinchu 30043 Taiwan, R.O.C. Phone: +886-3-5742579. Fax: +886-3-5715971. Email: cschang@ee.nthu.edu.tw. This research is supported in part by the Industrial Technology Research Institute (ITRI) under the contract G4-86021-a

of the PGPS scheme as follows: "When a packet arrives, virtual time is updated and the packet is stamped with its virtual finishing time. The server is work conserving and serves packets in an increasing order of timestamp."

Let L_{max} be the maximum packet length and $\hat{F}(p)$ (resp. $F(p)$) be the departure time of packet p under PGPS (resp. GPS). It is shown in [5] that

$$\hat{F}(p) - F(p) \leq L_{max} \tag{5}$$

for all packets. Moreover, for all τ and session i,

$$S_i(0,\tau) - \hat{S}_i(0,\tau) \leq L_{max}, \tag{6}$$

where $\hat{S}_i(\tau,t))$ is the amount of service received by the i^{th} session under PGPS in the interval $[\tau,t]$. Also, the difference of the backlogged workloads is also bounded by the maximum packet length as follows:

$$\hat{Q}_i(\tau) - Q_i(\tau) \leq L_{max} \tag{7}$$

for all τ and $i = 1,\ldots,N$, where $\hat{Q}_i(\tau)$ (resp. $Q_i(\tau)$) is the backlogged workload of the i^{th} session at time τ under PGPS (resp. GPS).

II. GPS/PGPS servers with time-varying capacity

In an integrated service network, best effort services are often assigned with a low priority. The capacity for these low priority services are time varying as they can only use the remaining capacity from high priority services. Instead of having a constant capacity in [5], in this paper we consider the case that the server has a time-varying capacity $r(t)$. In this case, the GPS scheme allocates the service rate to the i^{th} session according to the following rule

$$r_i(t) = \begin{cases} r(t)\phi_i / \sum_{j\in B(t)} \phi_j & \text{if } i \in B(t) \\ 0 & \text{otherwise} \end{cases},$$

and (1) is still satisfied. Moreover, for any session i that is continuously backlogged in the interval $[\tau,t]$, we modify (2) as follows:

$$S_i(\tau,t) \geq \frac{\phi_i}{\sum_{j=1}^{N}\phi_j}\int_\tau^t r(s)ds. \tag{8}$$

However, there might be a problem for the virtual time implementation of the PGPS scheme in [5] as shown in the following example.

Example 1. Let $r(t) = 0$ for $0 \leq t < 2$ and $r(t) = 1$ for $t \geq 2$. Suppose that there are no backlogged packets at $t = 0$. Assume that there are two sessions with $\phi_1 = 1$ and $\phi_2 = 2$. A packet of session 1 arrives at $t = 0$ with length 1, while another packet of session 2 arrives at $t = 1$ with length 1. The PGPS server in [5] will stamp the first (resp. second) packet with virtual finishing time $F_1^1 = 1$ (resp. $F_2^1 = 3/2$) when it arrives. As $r(t) = 0$ for $0 \leq t < 2$, the server will not begin its service until $t = 2$. At $t = 2$ the server sees these two backlogged packets, it will pick the first packet since it has a smaller virtual finishing time. Thus, the first packet departs earlier than the second one under the virtual time implementation of the PGPS scheme in [5]. However, under GPS, the server guarantees the second packet with the rate that is two times of the rate of the first packet from $t = 2$ onward. Thus, the second packet departs earlier than the first packet under GPS. This shows that the virtual time implementation violated the basic rule of the PGPS scheme.

To cope with this problem, we apply the concept of change of time. Let

$$w(t) = \int_0^t r(s)ds. \tag{9}$$

be the cumulative capacity up to time t. Note that $w(t)$ is nonnegative, increasing and continuous. Thus, it can be viewed as a *clock*. With respect to the new clock $w(t)$, the service rate is 1 since

$$\frac{\int_{t_1}^{t_2} r(s)ds}{w(t_2) - w(t_1)} = 1, \quad \text{for all } t_1 \leq t_2. \tag{10}$$

Since the server has a constant capacity for all time under the new clock $w(t)$, the virtual time implementation of PGPS can be used under the new clock $w(t)$.

Now we further assume that the time varying capacity $r(t)$ is piecewise constant, i.e., there is a sequence of *transition events* and $r(t)$ is constant between two successive transition events. To implement the PGPS under the new clock $w(t)$, we consider three types of events: arrivals, departures from the GPS server, and transition events for the capacity change. Let t_j be the time at which the j^{th} event occurs. Denote the time of the first arrival in a busy period as $t_1 = 0$, and let B_j be the set of sessions that are busy in the interval (t_{j-1}, t_j). Then the new clock $w(t)$ and the virtual time $V(w(t))$ (with respect to the new clock) can be tracked as follows:

$$\begin{aligned}
&w(t) = 0, \ \text{whenever } t \text{ is not in a busy period.} \\
&w(t_{j-1} + \tau) = w(t_{j-1}) + r(t_{j-1})\tau
\end{aligned} \tag{11}$$

$$\begin{aligned}
&V(w(t)) = 0, \ \text{whenever } t \text{ is not in a busy period.} \\
&V(w(t_{j-1}+\tau)) = V(w(t_{j-1})) + \frac{r(t_{j-1})\tau}{\sum_{i\in B_j}\phi_i}, \\
&\qquad \tau \leq t_j - t_{j-1}, j = 2,3,\ldots
\end{aligned} \tag{12}$$

Under the new clock, the time of the arrival event a_i^k is transformed to $w(a_i^k)$. Thus, when the k^{th} packet from the i^{th} session arrives, the packet is stamped with the virtual finishing time

$$F_i^k = \max[F_i^{k-1}, V(w(a_i^k))] + \frac{L_i^k}{\phi_i}, \ \text{ with } F_i^0 = 0 \text{ for all } i. \tag{13}$$

Denote by $Next(t)$ to be the next time to update $V(w(t))$ if there were no more arrival and transition events after time t. Analogously,

$$Next(t) = t + (F_{min} - V(w(t)))\frac{\sum_{i\in B_j}\phi_i}{r(t)}. \tag{14}$$

A careful examination of the above algorithm shows that the tracking of the new clock $w(t)$ in (11) is not needed. One may simply let $\tilde{V}(t) = V(w(t))$ and the virtual time $\tilde{V}(t)$ evolves in a busy period as follows:

$$\tilde{V}(t) = 0, \text{ whenever } t \text{ is not in a busy period.}$$
$$\tilde{V}(t_{j-1}+\tau) = \tilde{V}(t_{j-1}) + \frac{r(t_{j-1})\tau}{\sum_{i\in B_j}\phi_i},$$
$$\tau \leq t_j - t_{j-1}, j = 2,3,\ldots \quad (15)$$

Similarly, the virtual finishing time and the next time to update $\tilde{V}(t)$ if there were no more arrival and transition events after time t can be computed as follows:

$$F_i^k = \max[F_i^{k-1}, \tilde{V}(a_i^k)] + \frac{L_i^k}{\phi_i},$$
$$\text{with } F_i^0 = 0 \text{ for all } i. \quad (16)$$
$$Next(t) = t + (F_{min} - \tilde{V}(t))\frac{\sum_{i\in B_j}\phi_i}{r(t)}. \quad (17)$$

Given the modified mechanism for updating virtual time $\tilde{V}(t)$, the PGPS scheme with a time varying capacity can be defined as before. For Example 1, the virtual time $\tilde{V}(t)$ is stopped between $[0,2]$ as $r(t) = 0$ in this interval. Thus, we have $F_1^1 = 1$ and $F_2^1 = 1/2$, and this yields the same deparutre order as GPS.

Now we compare how the PGPS scheme approximates the GPS scheme with a time varying capacity. Following the notations used in (5)-(7), we let $\tilde{F}(p)$ be the departure time of packet p under the PGPS scheme with a time varying capacity. Also, let $\tilde{S}_i(\tau,t)$ be the amount of service received by the i^{th} session under PGPS in the interval $[\tau,t]$ and $\tilde{Q}_i(t)$ be the backlogged workload of the i^{th} session at time t under PGPS.

Theorem 1: *(i)* $S_i(0,t) - \tilde{S}_i(0,t) \leq L_{max}$ for all t, and *(ii)* $\tilde{Q}_i(t) - Q_i(t) \leq L_{max}$ for all t.

Proof: (i) As we have argued in (10) that the PGPS scheme with a time varying capacity is the same as the one with a constant capacity when viewed under the new clock $w(t)$, we have from (6) that

$$S_i(0,w(t)) - \tilde{S}_i(0,w(t)) \leq L_{max}$$

for all $w(t)$. This in turns implies that

$$S_i(0,t) - \tilde{S}_i(0,t) \leq L_{max}$$

for all t.

(ii) That

$$\tilde{Q}_i(t) - Q_i(t) \leq L_{max}$$

holds for all t follows from the same argument and (7). ■

To derive the bound for the packet departure times, we introduce the concept of the traffic envelopes in [3], [4], [1].

Definition 1: A process $A(t)$ is g-upper (resp. lower) constrained if $g : \mathcal{R} \mapsto \mathcal{R}$ is increasing with $f(0) = 0$ and

$$A(t_2) - A(t_1) \leq \text{ (resp. } \geq) \ g(t_2 - t_1)$$

for all $t_1 \leq t_2$.

Theorem 2: Suppose that $w(t)$ is g-lower constrained. Then

$$\tilde{F}(p) - F(p) \leq g^{-1}(L_{max}),$$

where $g^{-1}(t) = \inf\{s : g(s) > t\}$ is the inverse function of g.

Proof: As the PGPS scheme with a time varying capacity is the same as the one with a constant capacity when viewed under the new clock $w(t)$, we have from (5) that

$$w(\tilde{F}(p)) - w(F(p)) \leq L_{max}.$$

From the assumption that $w(t)$ is g-lower constrained, it follows that

$$g(\tilde{F}(p) - F(p)) \leq w(\tilde{F}(p)) - w(F(p)) \leq L_{max}.$$

Taking the inverse function on both sides of the inequality yields

$$\tilde{F}(p) - F(p) \leq g^{-1}(L_{max}).$$

■

III. Conclusions

In this paper, we proposed the virtual time implementation of the PGPS scheme with a time varying capacity. The key observation is that under an appropriate change of time, a server with a time varying capacity can be viewed as a server with a constant capacity. Based on such an observation, we also derive various bounds for the difference between the PGPS scheme and the GPS scheme, including packet departure times, the backlogged workload at time t, and the service received by time t. Other performance bounds in [5], [6] may be obtained similarly, with the help of the recently developed filtering theory for deterministic traffic regulation and service guarantees (see e.g., [2] and references therein). Moreover, the results could be extended to PGPS servers with priority as the service rates for low priority sessions are time varying. Some results along this line were derived in [7] and they will be reported separately.

References

[1] C.S. Chang, "Stability, queue length and delay of deterministic and stochastic queueing networks," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 913-931, 1994.
[2] C.S. Chang, "On deterministic traffic regulation and service guarantees: a systematic approach by filtering," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1097-1110, 1998.
[3] R.L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114-131, 1991.
[4] R.L. Cruz, "A calculus for network delay, Part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132-141, 1991.
[5] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: the single-node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344-357, 1993.
[6] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: the multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137-150, 1994.
[7] Y.C. Tseng, "Service guarantees for networks of GPS queues with time varying capacities," M.S. thesis, Dept. of Electrical Engineering, National Tsing Hua University, Hsinchu, June, 1997.