Service Curve Proportional Sharing Algorithm for Service-Guaranteed Multiaccess in Integrated-Service Wireless Networks

Chia-Sheng Chang, Kwang-Cheng Chen

Institute of Communication Engineering and Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, 10617, R.O.C.
E-mail: changcs@fcom.ee.ntu.edu.tw, chenkc@cc.ee.ntu.edu.tw

Abstract— **In [1] we systematically developed Packetized Service Curve Processor Sharing (PSCPS) scheduling policy for multimedia distributed networks. In this paper, we consider PSCPS in a special kind of distributed environments, wireless channels. Due to multi-path fading and interference distortion, the error characteristics of wireless channels are even bursty and time-varying. By accounting error control mechanisms, the fluctuation of error behavior can be equivalently regarded as the channel capacity is time-varying for a fixed bit-error-rate. We apply the "change of time" technique used in [6] to transform PSCPS to a scheduler operating with a time-varying capacity while the service curve guarantees still maintained. The "virtual time implementation" of PSCPS proposed in [1] is also modified so that it can work properly with time-varying capacity.**

I. Introduction

Future high-speed networks are designed to carry multimedia traffic in addition to conventional data traffic. Over the past a few years there have been many research results on the deterministic network analysis using rate-guaranteed link-sharing algorithms for leaky bucket regulated input traffic. However, these sharing policies base on "rate" concept to share link resource. That is, link resource are shared according to a set of real numbers, each of which is used to represent the characteristic rate of the corresponding input traffic source. Rate-characterization is often satisfactory if the input traffic behaves similar to constant-bit-rate (CBR) traffic. However, it is inefficient when the input traffic is very bursty, e.g. variable-bit-rate (VBR) traffic since it is generally impossible to characterize VBR with a single parameter.

Due to the diversed characteristics of multimedia VBR traffic, Cruz [3] proposed the following deterministic traffic characterization: Let $R_i(t_1,t_2)$ (bits) be the amount of arrival generated by the traffic source m in the interval $[t_1,t_2)$. Then traffic source m is said to be constrained by constrain function $b_m(\cdot)$ if there exists a nonnegative increasing function $b_m(\cdot)$ such that $R_m(t_1,t_2) \le b(t_2-t_1)$ for $t_2-t_1 \ge 0$. Since a constrain function $b(\tau)$ represents the upper bound of arrival from a connection in an interval of length τ, it can be intuitively regarded as the integration of arrival rate with respective to time, and can convey very much rate variation information in its waveform. Suppose we share the link resource according to these constrain functions, the contained rate variation information will help us allocate resource to VBR connections more efficiently.

In [1] we introduced the service curve proportional sharing concept for multimedia distributed networks. Based on this concept, a scheduling policy Packetized Service Curve Processor Sharing (PSCPS) that can guarantee Quality-of-Service (QoS) of real-time traffic source is proposed. Because of its proportional property, PSCPS does not need any traffic pre-regulation as the optimal scheduling policy NPEDF [7]. This characteristic makes PSCPS a "truly work-conserving" scheme and more effective to provide best-effort service for available-bit-rate (ABR) traffic. Therefore, although PSCPS is only sub-optimal in term of schedulable region, in real-world multimedia networks where both real-time and best-effort services are provided, PSCPS policy is even more attractive than the optimal scheduling policy NPEDF.

In [1], PSCPS scheduler always has a constant service rate. However, in certain situations this assumption may not hold. Typical wireless links suffer from severe multi-path fading, interference distortion and therefore the error characteristics of wireless channels are bursty and time-varying. By accounting error control mechanisms, the fluctuation of error behavior can be equivalently regarded as the wireless channel capacity is varying with time while keeping the bit-error-rate at a fixed level. Therefore, to model the error effect in wireless channels, we must assume the channel capacity $r(t)$ is time-varying.

II. Review of Packetized Service Curve Proportional Sharing Scheduler

Generalized Processor Sharing (GPS) processor [4] is probably the most well-known rate-proportional sharing scheme in the literature. Let $B(t)$ denote the set of backlogged connections at time t and r the rate of the server. Then the rate to serve connection i at time t, $r_i(t)$, is

$$r_i(t) = \begin{cases} \phi_i \times \frac{1}{\left(\sum_{j\in B(t)} \phi_j\right)} \times r & \text{if } i \in B(t), \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\phi_1, \cdots, \phi_M$ are the weighting factors assigned to connection $1, \cdots, M$, respectively. However, if all *weighting factors* in (1) are replaced with nonnegative increasing *weighting functions* (called service curves), we must figure out the exact meaning of proportionality in this situation. If observed carefully, (1) is actually simplified from a more general form:

$$r_i(t) = \begin{cases} \phi_i \circ \left(\sum_{j\in B(t)} \phi_j\right)^{-1}(r) & \text{if } i \in B(t), \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

This research was supported by the National Science Council of Taiwan, R.O.C. under the contract NSC-88-2219-E-002-002. The authors are with the institute of communications engineering, college of electrical engineering, National Taiwan University Taipei, Taiwan 10617, R.O.C., Email:{changcs@fcom, chenkc@cc}.ee.ntu.edu.tw.

where ϕ_i now represents a function $\phi_i(t) \triangleq \phi_i \cdot t$ for each i, $(\sum_{j\in B(t)} \phi_j)(t)$ is defined by $\sum_{j\in B(t)} \phi_j(t)$, and the symbol "$\circ$" denotes the composition of functions.

By the term "work", we refer to the total amount of "bits" served by the server from the beginning of this busy period. In a SCPS system, we concern work allocation instead of rate allocation. Suppose there are M connections with strictly increasing, continuous service curve requirements $S_1(\cdot),\cdots,S_M(\cdot)$, which are nonnegative, continuous, strictly increasing functions. Let $B(t)$ be a subset which contains the connections continuously backlogged in $[0,t)$, and other connections are continuously idle in $[0,t)$. Then we can reasonably define the service curve proportional sharing (SCPS) as follows: The work distributed to connection i in $[0,t)$, $W_i(t)$, is

$$W_i(t) = \begin{cases} S_i \circ \left(\sum_{j\in B(t)} S_j\right)^{-1}(r\cdot t) & \text{if } i \in B(t), \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $(\sum_{j\in B(t)} S_j)(t) \triangleq \sum_{j\in B(t)} S_j(t)$ and the symbol "$\circ$" denotes the composition of functions. Although the above situation is just a special case for SCPS, (3) is the key and fundamental thought of the SCPS processor.

A. SCPS Processor and PSCPS scheduler

Consider M input connections sharing a server with output rate r bps. Incoming packets of each connection are stored in its corresponding input queue waiting for service. Throughout this paper we adopt the convention that a packet has arrived only after its last bit has arrived.

Consider a busy period starting at time 0. Let $R_m(t_1,t_2)$ and $T_m(t_1,t_2)$ denote the amount of arrival and departure of connection m (bits) in time interval $[t_1,t_2)$, respectively. The amount of backlog of connection m (bits) at time t is defined by $Q_m(t) \triangleq R_m(0,t) - T_m(0,t)$, and consequently we define the set of backlogged connections $B(t) \triangleq \{m : Q_m(t) > 0\}$. The goal of SCPS processors is to simultaneously guarantee each connection m a nonnegative increasing service curve $S_m(\cdot)$ with $S_m(0) = 0$ though a finite number of discontinuous jumps are allowed. According to [2], we say the service curve $S_m(\cdot)$ of connection m is guaranteed if for each t and $m \in B(t)$, there exists $u \le t$ such that $Q_m(u) = 0$ and $T_m(u,t) \ge S_m((t-u)^+)$. For a busy period starting at time 0 and each t and connection $m \in B(t)$, we define $f_m(t)$ according to [2][5](Figure 1):

$$f_m(t) \triangleq \min\{T_m(0,s) + S_m((t-s)^+)\}, \quad (4)$$

where the minimization is taken over the ending time $s < t$ of all intervals in which $Q_m(z) = 0$. According this definition, it can be seen that $f_m(t)$ is (not necessarily strictly) increasing. With the above definition, the service index $\tau^s_m(t) \in \mathbb{R}^3$ of connection $m \in B(t)$ at t is defined by

$$\tau^s_m(t) \triangleq \Psi(f_m, T_m(0,t)) \triangleq (\Psi_1, \Psi_2, \Psi_3), \quad (5)$$

where

$$\tau^s_m(t).scalar = \Psi_1 \triangleq \sup\{y : f_m(y^-) \le T_m(0,t)\}$$

$$\tau^s_m(t).level = \Psi_2 \triangleq T_m(0,t) - f_m(\tau^s_m(t).scalar^-)$$

$$\tau^s_m(t).cavity = \Psi_3 \triangleq f_m(\tau^s_m(t).scalar) - f_m(\tau^s_m(t).scalar^-)$$

are the three components of $\tau^s_m(t)$. And, conversely, Φ is the inverse operation of Ψ such that

$$\begin{aligned} T_m(0,t) = \Phi(f_m, \tau^s_m(t)) \triangleq\ & f_m(\tau^s_m(t).scalar) \\ & - \max\{(f_m(\tau^s_m(t).scalar) \\ & - f_m(\tau^s_m(t).scalar^-) - \tau^s_m(t).level), 0\}. \end{aligned}$$

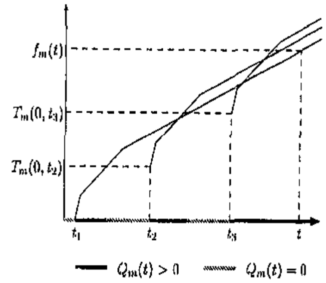

Fig. 1. The computation of $f_m(t)$.

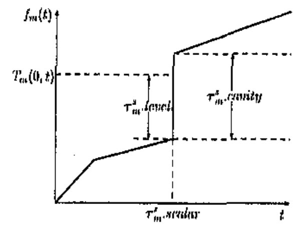

Fig. 2. Definition of $\Psi(f_m, T_m(0,t))$.

In Figure 2, we show the function Ψ graphically. Briefly speaking, $\Psi(f_m,\cdot)$ is actually a "generalized inverse function" of f_m such that the inverse mapping remains well-defined at discontinuous points of f_m.

However, since $\tau^s_m(t) \in \mathbb{R}^3$, its order relation must be defined carefully as follows: Suppose there are two service indices $\tau_1 = (\tau_1.scalar, \tau_1.level, \tau_1.cavity)$ and $\tau_2 = (\tau_2.scalar, \tau_2.level, \tau_2.cavity)$, then:

if $\tau_1.scalar \ne \tau_2.scalar$
  if $\tau_1.scalar < \tau_2.scalar$, then $\tau_1 < \tau_2$.
  else $\tau_2 < \tau_1$.
else
  if $\tau_1.level = \tau_1.cavity$ and $\tau_2.level = \tau_2.scalar$, then $\tau_1 = \tau_2$.
  else if $\tau_1.level < \tau_1.cavity$ and $\tau_2.level = \tau_2.scalar$, then $\tau_1 < \tau_2$.
  else if $\tau_1.level = \tau_1.cavity$ and $\tau_2.level < \tau_2.scalar$, then $\tau_2 < \tau_1$.
  else
    if $\tau_1.level < \tau_2.level$, then $\tau_1 < \tau_2$.
    else if $\tau_2.level < \tau_1.level$, then $\tau_2 < \tau_1$.
    else $\tau_1 = \tau_2$.

With the definition of service index, we define some notations as follows:

- $\tau^s_h(t)$ is defined by:

$$\tau^s_h(t) \triangleq \min\{\tau^s_m(t) : m \in B(t)\}, \quad (6)$$

which is the minimum of the service indices of all backlogged connections.

- $B_h(t) \triangleq \{m \in B(t) : \tau^s_m(t) = \tau^s_h(t)\}$, which is the set of backlogged connections whose service indices equal to $\tau^s_h(t)$. The subscript h implies that this set is of higher priority to be served.
- $B_l(t) \triangleq B(t)\backslash B_h(t)$, which is the complement set of $B_h(t)$ with respect to $B(t)$. The subscript l implies that this set is of lower priority to be served.

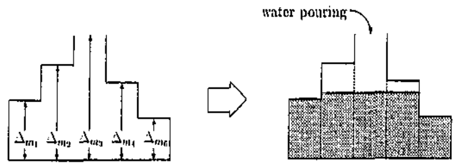

Fig. 3. Service distribution at discontinuous points: water filling model.

• $\tau_l^s(t) \triangleq \min\{\tau_m^s : m \in B_l(t)\}$, or $\tau_l^s(t) \triangleq (\infty,0,0)$ if $B_l(t)$ is empty. τ_l^s represents the minimum of the service indices of all connection in $B_l(t)$.

In short, SCPS distributes service among the connections that have the smallest service indices, as specified in the following definition.

Definition 1: Given a server with service rate r, Service Curve Proportional Sharing (SCPS) processor is work-conserving. Consider a busy period beginning at time 0. At any time instant t, the server serves each connection $m \in B_h(t)$ by updating $\tau_h^s(t)$ according to the following equation:

$$\tau_h^s(t) = \Psi((\sum_{m \in B_h(t)} f_m), rt - \sum_{m \notin B_h(t)} T_m(0,t)) \tag{7}$$

Then by the definition of service index $\tau_m^s(t)$, the amount of departure of connection $m \in B_h(t)$ in time interval $[0,t)$ equals to $\Phi(f_m, \tau_h^s(t))$.

From Definition 1, it can be seen that SCPS processor distributes service to each connection $m \in B_h(t)$ in an "inverse-function" manner. At discontinuous points of $f_{total}(t) \triangleq (\sum_{m \in B_h(t)} f_m)(t)$, our earlier definitions of service index and its order relation enable the SCPS processor to distribute the service among input connections in a special manner. Suppose $f_{total}(\cdot)$ is discontinuous at $\hat{t}$ and has a finite jump contributed from the discontinuities of $f_{m_1}, f_{m_2}, \ldots, f_{m_N}$, $m_i \in B_h(t)$ for $i = 1, 2, \ldots N$. At this discontinuous point $\hat{t}$, the service is exclusively distributed to those N connections in a "water filling" approach as in Figure 3, where $\Delta_i \triangleq f_i(\hat{t}^+) - f_i(\hat{t}^-)$ for $i \in \{m_1, m_2, \ldots, m_N\}$.

Proposition 1 identifies the sufficient condition of service curve allocation such that a SCPS processor can simultaneously guarantee the service curve of each connection.

Proposition 1: Consider M nonnegative, increasing curves $S_m(t)$ with $S_m(0) = 0$, $m = 1, \ldots, M$. If $\sum_{m=1}^{M} S_m(t^+) \leq r \cdot t$ for all $t \geq 0$, then SCPS processor guarantees a service curve $S_m(\cdot)$ for each connection m.

The complete proof of Proposition 1 is presented in [1] and is omitted here.

The operation of a SCPS processor is based on the assumption of fluid model. However, in modern packet switching networks, data is transmitted in unit of packet, which is an indivisible entity. Inspired by the Packet-by-Packet Generalized Processor Sharing (PGPS) scheduler [4], we induce a scheduling algorithm according to the packet departure order in a SCPS system. Let F_p be the time at which packet p departs SCPS, and we give the following definition.

Definition 2: Packetized Service Curve Proportional Sharing (PSCPS) scheduler is a work-conserving scheme which serves packets in increasing order of F_p.

Due to the proportional property of SCPS processor, we have the following lemma, and the proof is in [1]. This lemma is required to prove several theorems afterward.

Lemma 1: Consider two packets p and p' in a SCPS system at time t and assume p is the k^{th} packet of connection m, p' is the l^{th} of connection n, respectively. Suppose that packet p completes service before packet p' if there are no arrivals after time t. Then packet p will also complete service before packet p' for any pattern of arrivals after time t.

With Lemma 1, we know the following three theorems holds, which quantitatively measure the difference between the output processes of a SCPS processor and a PSCPS scheduler (developed in [4], [7]):

Theorem 1: Denote F_p and $\hat{F}_p$ the departure time at which packet p departs under SCPS and PSCPS, respectively. Suppose r is the rate of the server and L_{max} is the maximum packet length. Then for all packet p.

$$\hat{F}_p - F_p \leq L_{max}/r \tag{8}$$

Theorem 2: Let $T_m(t_1,t_2)$ and $\hat{T}_m(t_1,t_2)$ denote the amount of connection m served under SCPS and PSCPS in the interval $[t_1,t_2]$. For all time t and connection m,

$$T_m(0,t) - \hat{T}_m(0,t) \leq L_{max}. \tag{9}$$

Theorem 3: Let $Q_m(t)$ and $\hat{Q}_m(t)$ denote the backlog of connection m at time t under SCPS and PSCPS, respectively. For all time t and connection m

$$\hat{Q}_m(t) - Q_m(t) \leq L_{max}. \tag{10}$$

With the above propositions, it can be seen PSCPS scheduler is a close approximation of SCPS processor. In subsequent sections, packet delay bound and the bound on queue length will be derived in the SCPS system, which assumes a fluid model. Based on these propositions, we can easily obtain performance bounds (packet delay, queue length, etc) for PSCPS from bounds for SCPS.

B. Virtual Time Implementation of PSCPS

In [1], an efficient scheme called "virtual time implementation" is proposed as a practical implementation of PSCPS scheduler. we say an event occurs at each of the following time instants:

1. arrival event: arrival at SCPS processor.
2. departure event: departure from SCPS processor.
3. joint event: time instant t at which $\tau_h^s(t)$ equals $\tau_l^s(t)$.

Let t_j be the time at which the j^{th} event occurs (simultaneous events are ordered arbitrarily). Note that all these three events may change the set $B_h(t)$.

Let the time of the first arrival of a busy period be denoted as $t_1 = 0$. Since the $B_h(t)$ is fixed in the interval (t_{j-1}, t_j), we denote this set as B_j. The virtual time $S(t) \in \mathbb{R}^3$ is set to (0,0,0) for all times when the server is idle. Consider any busy period, and denote the virtual time that it begins as time zero. Then $S(t)$ is defined as follows:

$$S(0) = (0,0,0)$$

$$S(t_{j-1}+\tau) = \Psi((\sum_{m\in B_j} f_m), r(t_{j-1}+\tau) - \sum_{m\notin B_j} T_m(0,t_{j-1}))$$

$$\tau \le t_j - t_{j-1}, \quad j = 2,3,\ldots \tag{11}$$

Now suppose that the k^{th} packet of connection m packet arrives at time a_m^k and has length L_m^k. Then define the virtual finishing time F_m^k at which this packet completes service by $F_m^k = \Psi(f_m, (\sum_{i=1}^k L_m^i))$ where $f_m(\cdot)$ is defined in (4) according to the arrival and departure process of connection m in the SCPS system. The order relation of F_m^k is as follows: If the first components of two virtual finishing times is different, the one with smaller first component is smaller. Otherwise, if the first components are identical, the one with the smaller second component is smaller. Then the packets are served in increasing order of virtual finishing time.

Note that we still have to update virtual time $S(t)$ when there are events in the SCPS system. Define $Next(t)$ to be the real time at which the next departure or joint event in the SCPS system after t if there are no more arrivals after time t. Suppose the event just prior to t is the $(j-1)^{th}$ event and let F_{min} be the smallest virtual finishing time of a packet of connection m in the system at time t. Also recall that the service index τ_l^s represents the nearest joint event and suppose it corresponds to connection n (i.e. $\tau_l^s = \tau_n^s$). Then from (11), we have:

$$\min\{\tau_l^s, F_{min}\} = \Psi((\sum_{m\in B_j} f_m), r\cdot Next(t) - \sum_{m\notin B_j} T_m(0,t_{j-1}))$$

$$\therefore Next(t) = \frac{1}{r}\cdot(\Phi((\sum_{m\in B_j} f_m), \min\{\tau_l^s, F_{min}\}) + \sum_{m\notin B_j} T_m(0,t_{j-1}))$$

Given the mechanism for updating virtual time $S(t)$, PSCPS is defined 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 virtual finish time.

III. PSCPS Scheduler with Time Varying Capacity

Typical wireless links suffer from severe multi-path fading, interference distortion and the error characteristics of wireless channel are even bursty and time-varying. By accounting error control mechanisms, the fluctuation of error behavior can be equivalently regarded as the wireless channel capacity is varying with time while keeping the bit-error-rate at a fixed level. To model this time-varying effect, we assume the channel capacity $r(t)$ is a function of time and piece-wise constant, i.e. there is a sequence of transition events and $r(t)$ is constant between two successive transition events [6]. Here we also assume the SCPS processor is work-conserving, which means the processor either is idle when all input connection queues are empty, or operates at the full rate $r(t)$ when at least one input connection queue is backlogged.

In the following we apply the "change of time" technique used in [6] to transform PSCPS to a scheduler operating in a time-varying capacity environment. Let

$$\nu = \omega(t) = \int_0^t r(s)ds, \tag{12}$$

where 0 denotes the beginning of a busy period and t is a time instant in this busy period. Since $\omega(t)$ is nonnegative, strictly increasing, and continuous, it can be viewed as a clock [6]. We denote the time axis with respect to this new clock $\omega(t)$ as the ν-axis, and use $\omega(\cdot)$ and its inverse, $\omega^{-1}(\cdot)$, as the forward and inverse mapping between t-axis and ν-axis and denote $\nu_i = \omega(t_i)$. It can be seen that with respect to the ν-axis, the service rate is normalized to unity since

$$\frac{\text{work in } [\nu_1,\nu_2]}{\nu_2-\nu_1} = \frac{\int_{t_1}^{t_2} r(s)ds}{\omega(t_2)-\omega(t_1)} = 1$$

for all ν_1, ν_2 in the same busy period on ν-axis with $\nu_1 < \nu_2$. Therefore the virtual time implementation of PSCPS for constant capacity can be used under this new clock with capacity set to unity.

In order to explicitly clarify the exact context (i.e. with respect to t-axis or ν-axis), a tilde ($\sim$) is added on the top of a notation to emphasize that it is expressed with respect to ν-axis. To implement PSCPS under the new clock, we consider four kinds of event on the ν-axis. We define an event occurring at each of the following time instants:

1. arrival event: arrival at SCPS processor on ν-axis.
2. departure event: departure from SCPS processor on ν-axis.
3. joint event: time instant ν at which $\widetilde{\tau}_h^s(\nu)$ equals $\widetilde{\tau}_l^s(\nu)$.
4. transition event: time instant ν corresponding to a capacity change on the t-axis.

Also, let t_j (resp. ν_j) be the time on t-axis (resp. ν-axis) at which the j^{th} event occurs and $t_1 = 0$ (resp. $\nu_1 = 0$) represents the beginning of this busy period.

Since the set $\widetilde{B}_h(\nu)$ is fixed in the interval (ν_{j-1}, ν_j), we denote this set as $\widetilde{B}_j$. Now, according to the definition of virtual time in (11), we have

$$\widetilde{S}(0) = (0,0,0)$$

$$\nu_{j-1}+\tau = \Phi((\sum_{m\in\widetilde{B}_j} \widetilde{f}_m), \widetilde{S}(\nu_{j-1}+\tau)) + \sum_{m\notin\widetilde{B}_j} \widetilde{T}_m(0,\nu_{j-1}) \tag{13}$$

$$\tau \le \nu_j - \nu_{j-1}, \quad j = 2,3,\ldots$$

Note the symbol r is vanished because the capacity is unity on ν-axis. However, the functions $\widetilde{f}_m(\cdot)$ need a more precise definition. Recall that guaranteeing service curve S_m of connection $m \in B(t)$ means that for each t, there exists $u \le t$ such that $Q_m(u) = 0$ and $T_m(u,t) \ge S_m((t-u)^+)$. However, this service curve requirement is defined with respect to t-axis, and can not be directly used on ν-axis because the uniformity of t-axis has been distorted due to the mapping $\omega(\cdot)$. Therefore, by transforming the service curve requirement to ν-axis, it follows for $m \in \widetilde{B}(\nu)$:

$$\widetilde{f}_m(\nu) \triangleq \min\{\widetilde{T}_m(0,\eta) + \widetilde{S}_m(\eta,\nu)\}$$

where $\widetilde{S}_m(\eta,\nu) \triangleq S_m((\omega^{-1}(\nu) - \omega^{-1}(\eta))^+)$, and the minimization is taken over the ending time $\eta < \nu$ of all intervals on ν-axis in which $\widetilde{Q}_m(\mu) = 0$.

From the definition of service index in (5) and (6), (13) can be written as

$$\omega(t_{j-1})+\tau = \sum_{m\in\widetilde{B}_j} \widetilde{T}_m(0,\omega(t_{j-1})+\tau) + \sum_{m\notin\widetilde{B}_j} \widetilde{T}_m(0,\omega(t_{j-1}))$$

$$\tau \le \nu_j - \nu_{j-1}, \quad j = 2,3,\ldots$$

Since the capacity $r(t)$ is constant between the $(j-1)^{th}$ and j^{th} event, the capacity is denoted as r_{j-1} when $t \in (t_{j-1}, t_j)$. Then $\omega(t_{j-1}) + \tau$ can be further expressed as

$$\omega(t_{j-1}) + \tau \cdot r_j = \sum_{m \in \widetilde{B}_j} \widetilde{T}_m(0, \omega(t_{j-1} + \tau)) + \sum_{m \notin \widetilde{B}_j} \widetilde{T}_m(0, \omega(t_{j-1}))$$
$$\tau \leq t_j - t_{j-1}, \quad j = 2, 3, \ldots \tag{14}$$

Next we prove the following theorem, which is the essential relation between the work with respect to t-axis and the work with respect to ν-axis.

Proposition 2: Suppose $\nu = \omega(t)$. The work (or, service) received by connection m in SCPS system during the interval $[0, \nu)$, i.e. $\widetilde{T}_m(0, \nu)$, is equal to $T_m(0, t)$ for all ν in a busy period on ν-axis. The same result also holds in a PSCPS system.

Proof:

(a) According to the work-conserving property of SCPS processor on ν-axis, for a fixed connection m we have

$$\widetilde{T}_m(0, \nu) + \sum_{n \neq m} \widetilde{T}_n(0, \nu) = r \cdot \nu. \tag{15}$$

However, since the server has constant capacity 1 with respect to ν-axis, (15) reduces to $\widetilde{T}_m(0, \nu) + \sum_{n \neq m} \widetilde{T}_n(0, \nu) = \nu$, which means $\widetilde{T}_m(0, \nu)$ can be regarded as the shared "time" in $[0, \nu)$ on the ν-axis. However, recall that the "time" on ν-axis is actually defined according to the accumulated work the time-varying capacity server produces in $[0, t)$ on t-axis (Equation (12)). Hence part (a) follows.

(b) Note that part (a) is based on the work-conserving property of SCPS processor. Since PSCPS scheduler is also work-conserving by definition, the same result follows. ∎

Next we prove another theorem which will be useful when the set $\widetilde{B}_j$ is transformed back to the t-axis.

Proposition 3: Suppose $\nu = \omega(t)$. Then the first components of service indices $\widetilde{\tau}^s_m(\nu)$ and $\tau^s_m(t)$ are related by $\widetilde{\tau}^s_m(\nu).scalar = \omega(\tau^s_m(t).scalar)$. The second and third components of $\widetilde{\tau}^s_m(\nu)$ are equal to their counterparts of $\tau^s_m(t)$, respectively.

Proof: By definition of service indices, we know a service index is actually a vector in $\mathbb{R}^3$. In the following, we will prove the equality of each component of a service index. By definition, the first components of $\widetilde{\tau}^s_m(\nu)$ and $\tau^s_m(t)$ are $\widetilde{\tau}^s_m(\nu).scalar = \sup\{\zeta : \widetilde{f}_m(\zeta^-) \leq \widetilde{T}_m(0, \nu)\}$ and $\tau^s_m(t).scalar = \sup\{y : f_m(y^-) \leq T_m(0, t)\}$. Since ω is a non-negative, continuous, strictly increasing mapping from t-axis to ν-axis, hence it follows

$$\begin{aligned}
&\omega(\tau^s_m(t).scalar) \\
&= \sup\{\omega(y) : f_m(y^-) \leq T_m(0, t)\} \\
&= \sup\{\zeta : f_m((\omega^{-1}(\zeta))^-) \leq \widetilde{T}_m(0, \nu)\} \\
&= \sup\{\zeta : \min\{T_m(0, s) + S_m((\omega^{-1}(\zeta) - s)^-)\} \leq \widetilde{T}_m(0, \nu)\} \\
&= \sup\{\zeta : \min_s\{\widetilde{T}_m(0, \omega(s)) + S_m((\omega^{-1}(\zeta) - s)^-)\} \leq \widetilde{T}_m(0, \nu)\} \\
&= \sup\{\zeta : \min_\eta\{\widetilde{T}_m(0, \eta) + S_m((\omega^{-1}(\zeta) - \omega^{-1}(\eta))^-)\} \leq \widetilde{T}_m(0, \nu)\} \\
&= \sup\{\zeta : \widetilde{f}_m(\zeta^-) \leq \widetilde{T}_m(0, \nu)\} \\
&= \widetilde{\tau}^s_m(\nu).scalar
\end{aligned} \tag{16}$$

As for the second components, we must first prove $\widetilde{f}_m((\widetilde{\tau}^s_m(\nu).scalar)^-) = f_m((\tau^s_m(t).scalar)^-)$. However, from the result and derivation of (16), we can see that

$$\begin{aligned}
f_m((\tau^s_m(t).scalar)^-) &= f_m((\omega^{-1}(\widetilde{\tau}^s_m(\nu).scalar))^-) \\
&= \widetilde{f}_m((\widetilde{\tau}^s_m(\nu).scalar)^-) \\
&= \widetilde{f}_m((\widetilde{\tau}^s_m(\nu).scalar)^-)
\end{aligned}$$

Consequently, it follows

$$\begin{aligned}
\tau^s_m(t).level &= T_m(0, t) - f_m((\tau^s_m(t).scalar)^-) \\
&= \widetilde{T}_m(0, \nu) - \widetilde{f}_m((\widetilde{\tau}^s_m(\nu).scalar)^-) \\
&= \widetilde{\tau}^s_m(\nu).level
\end{aligned}$$

By the very similar procedure, one can easily show $\widetilde{f}_m(\widetilde{\tau}^s_m(\nu).scalar) = f_m(\tau^s_m(t).scalar)$. Therefore, for the third components it is straightforward to see

$$\begin{aligned}
\tau^s_m(t).cavity &= f_m(\tau^s_m(t).scalar) - f_m((\tau^s_m(t).scalar)^-) \\
&= \widetilde{f}_m(\widetilde{\tau}^s_m(\nu).scalar) - \widetilde{f}_m((\widetilde{\tau}^s_m(\nu).scalar)^-) \\
&= \widetilde{\tau}^s_m(\nu).cavity
\end{aligned}$$

∎

Three immediate corollaries of Proposition 3 are presented in the following.

Corollary 1: Consider the k^{th} connection m packet with length L^k_m, and denote the virtual finishing time on ν-axis and t-axis by $\widetilde{F}^k_m \triangleq \Psi(\widetilde{f}_m, \sum_{l=1}^k L^l_m)$ and $F^k_m \triangleq \Psi(f_m, \sum_{l=1}^k L^l_m)$, respectively. Then the first components of virtual finishing times $\widetilde{F}^k_m$ and F^k_m are related by $\widetilde{F}^k_m.scalar = \omega(F^k_m.scalar)$. The second and third components of $\widetilde{F}^k_m$ are equal to their counterparts of F^k_m, respectively.

Corollary 2: The order relation of service indices with respect to ν-axis is preserved through the mapping $\omega^{-1}(\cdot)$. That is, for two connection m, $n \in \widetilde{B}(\nu)$ and $\nu = \omega(t)$ in a busy period, if $\widetilde{\tau}^s_m(\nu) > \widetilde{\tau}^s_n(\nu)$ (or, $=$, $<$), then we also have $\tau^s_m(t) > \tau^s_n(t)$ (or, $=$, $<$).

Corollary 3: The order relation of virtual finishing time with respect to ν-axis is preserved through the mapping $\omega^{-1}(\cdot)$. That is, for the k^{th} connection m packet and the l^{th} connection n packet, if $\widetilde{F}^k_m > \widetilde{F}^l_n$ (or, $=$, $<$), then we also have $F^k_m > F^l_n$ (or, $=$, $<$).

According to our earlier terminologies and Corollary 2, we know that B_j (the set of connection with the smallest service indices in (t_{j-1}, t_j) on t-axis) and $\widetilde{B}_j$ (the set of connection with the smallest service indices in (ν_{j-1}, ν_j) on ν-axis) are actually identical. Consequently, (14) can be written as

$$\omega(t_{j-1}) + \tau \cdot r_j = \sum_{m \in B_j} T_m(0, t_{j-1} + \tau) + \sum_{m \notin B_j} T_m(0, t_{j-1})$$
$$\tau \leq t_j - t_{j-1}, \quad j = 2, 3, \ldots$$

which implies the virtual time $S(t)$ for a time-varying capacity SCPS is defined by

$$S(0) = (0, 0, 0)$$
$$\omega(t_{j-1}) + \tau \cdot r_j = \Phi((\sum_{m \in B_j} f_m), S(0, t_{j-1} + \tau)) + \sum_{m \notin B_j} T_m(0, t_{j-1})$$
$$\tau \leq t_j - t_{j-1}, \quad j = 2, 3, \ldots \tag{17}$$

Since the order of virtual finishing times computed on ν-axis and t-axis are the same (Corollary 3), the packets can just be served in increasing order of virtual finishing time F^k_m computed on t-axis according to the order relation prior to Lemma 1.

Note that we still have to update virtual time $S(t)$ when there are events in the SCPS system. Define $Next(t)$ to be the real time at which the next departure or joint event in the SCPS system after t if there are no more arrivals after time t. Suppose the event just prior to t is the $(j-1)^{th}$ event. Let F_{min} be the smallest virtual finishing time of a packet in the system at time t. Then from (17), we have:

$$\omega(Next(t)) = \Phi((\sum_{m\in B_j} f_m), \min\{\tau_l^s, F_{min}\}) + \sum_{m\notin B_j} T_m(0, t_{j-1})$$

$$\therefore Next(t) = \omega^{-1}(\Phi((\sum_{m\in B_j} f_m), \min\{\tau_l^s, F_{min}\}) + \sum_{m\notin B_j} T_m(0, t_{j-1}))$$

Given the mechanism for updating virtual time, PSCPS is defined as follows: When a packet arrives, virtual time is updated and the packet is stamped with its service finishing index. The server is work conserving and serves packets in an increasing order of virtual finish time.

Now we want to find out how close the PSCPS scheduler approximates the SCPS processor under the assumption of time-varying capacity. We follow the same notations used in (8)-(10). To derive the difference between the packet departure times in SCPS and PSCPS under time-varying capacity assumption, we adopt a definition in [6].

Definition 3: An nonnegative increasing function $f(t)$ is g-lower constrained if $g : \mathbb{R} \to \mathbb{R}$ is nondecreasing with $g(0) = 0$ and $f(t_2) - f(t_1) \geq g(t_2 - t_1)$ for all $t_1 \leq t_2$.

Theorem 4 is first proved in [6], and the proof follows along the lines of the proof in [6, Theorem 2]. We present it here for completeness and the convenience of the readers (note that we use the same notations in Theorem 1):

Theorem 4: Suppose that $\omega(t)$ is g-lower constrained. Then $\hat{F}_p - F_p < g^{-1}(L_{max})$, where $g^{-1}(t) \triangleq \inf\{s : g(s) > t\}$ is the inverse function of g.

Proof: Since the PSCPS scheme with a time-varying capacity is the same as the one with a constant unity capacity when viewed under the new clock $\omega(t)$, we have from (8) that $\omega(\hat{F}_p) - \omega(F_p) \leq L_{max}$. From the assumption that $\omega(t)$ is g-lower constrained, it follows that $g(\hat{F}_p - F_p) \leq \omega(\hat{F}_p) - \omega(F_p) \leq L_{max}$. Taking the inverse function on both sides of the inequality yields the result. ■

The next proposition concern the differences between the received service and queue length in PSCPS and SCPS system, respectively.

Proposition 4: (a) $T_m(0,t) - \hat{T}_m(0,t) \leq L_{max}$ for all t, and (b) $\hat{Q}_m(t) - Q_m(t) \leq L_{max}$ for all t.

Proof:

(a) From Theorem 1, we know that with respect to the new clock $\nu \triangleq \omega(t)$

$$\tilde{T}_m(0,\nu) - \tilde{\hat{T}}_m(0,\nu) \leq L_{max}$$

By applying Proposition 2 to the left hand side of the above inequality, the result follows.

(b) From the result of (a), we have:

$$(R_m(0,t) - Q_m(t)) - (R_m(0,t) - \hat{Q}_m(t)) \leq L_{max},$$

which implies our result. ■

In [1] we define a server guarantees the service curve $S_m(\cdot)$ of traffic source m if for each t and $m \in B(t)$, there exists $u \leq t$ such that $Q_m(u) = 0$ and $T_m(u,t) \geq S_m((t-u)^+)$. Our last proposition is a sufficient condition for guaranteeing service curve for each connections in wireless channels where the capacity is time-varying. The proof can be easily obtained from the proof of constant capacity case in [1].

Proposition 5: Consider M nonnegative, increasing curves $S_m(t)$ with $S_m(0) = 0$, $m = 1, \ldots, M$. If $\omega(t)$ is g-lower constrained and $\sum_{m=1}^{M} S_m(t^+) \leq g(t)$ for all $t \geq 0$, then SCPS processor guarantees a service curve $S_m(\cdot)$ for each connection m.

IV. Conclusions

In this paper, we consider PSCPS scheduling policy in QoS-guaranteed wireless multiaccess networks where the capacity is varying with time for fixed bit-error-rate. By applying the "change of time" technique in [6], the PSCPS scheduler with time-varying capacity can be viewed as one with unity capacity with respect to the new clock. Via appropriate transform of this constant capacity scheduler, we systematically derived the PSCPS scheduling operation with time-varying capacity along with the practical "virtual time implementation" implementation in this time-varying case. In addition to various difference bounds (amount of received service, queue length) between PSCPS and SCPS schemes, we also provided the sufficient conditions for guaranteeing service curve in time-varying capacity wireless channels, which is critical for guaranteeing QoS for real-time traffic in wireless multiaccess networks. Therefore, we believe PSCPS policy is a potential approach for multimedia wireless networks.

References

[1] Chia-Sheng Chang, Kwang-Cheng Chen, "Service Curve Proportional Sharing Algorithm for Service-Guaranteed Multiaccess in Integrated-Service Distributed Networks," Submitted to *IEEE Globecom'99*.

[2] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal of Selected Area in Commu.*, Vol. 13, pp. 1048-1056, 1995.

[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] A. K. Parekh, R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 344-357, June 1993.

[5] Cheng-Shang Chang, "On deterministic traffic regulation and service guarantees: a systematic approach by filtering," *IEEE Trans. on Information Theory*, vol. 44, no. 3, May, 1998.

[6] Y. C. Tseng, C. S. Chang, "PGPS Servers with Time-Varying Capacities," *IEEE Communications Letters*, Vol. 2, No. 9, Sep. 1998.

[7] L. Georgiadis, R. Guérin, A. Parekh, "Optimal Multiplexing on a Single Link: Delay and Buffer Requirements," *IEEE Transactions on Information Theory*, Vol. 43, No. 5, Sep, 1997.

[8] H. Sariowan, "A Service Curve Approach to Performance Guarantees in Integrated-Service Networks," *Ph.D. dissertation in Electrical and Computer Engineering*, University of California, San Diego, 1996.

[9] Kwang-Cheng Chen, Cheng-Hua Lee, "RAP—A Novel Medium Access Control Protocol for Wireless Data Networks," *Proc. IEEE Globecom'93*, Houston, 1993.

[10] K. C. Chen, "Multiple Access for Wireless Packet Networks," Multiple Access, Mobility, and Traffic for Personal Comm. Workshop, Melboune, Dec. 1997.

[11] Chia-Sheng Chang, Kwang-Cheng Chen " Guaranteed Quality-of-Service Wireless Medium Access by Packet-by-Packet Generalized Processor Sharing Algorithm," *Proc. IEEE ICC'98*, Atlanta, 1998.