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

Chia-Sheng Chang and Kwang-Cheng Chen
Institute of Communication Engineering, College of Electrical Engineering
National Taiwan University, Taipei, Taiwan, 10617, R.O.C.
Tel: +886-2-23635251 ext. 533, +886-2-23635251 ext. 246
E-mail: changcs@fcom.ee.ntu.edu.tw, chenkc@cc.ee.ntu.edu.tw

Abstract— **In this paper we introduce the service curve proportional sharing concept to guarantee Quality-of-Service (QoS) in multimedia distributed networks. Based on this concept, we systematically develop a scheduling policy Packetized Service Curve Processor Sharing (PSCPS) along with the corresponding feasible service curve allocation condition and practical implementation procedure. It is shown that PSCPS possesses the more desirable property to provide QoS-guaranteed service and best-effort service at the same time. It is further demonstrated that head packet information is sufficient for PSCPS operating in distributed environments including wireless channels. Implementation with only head packet information and an effective information exchange scheme are successfully developed. Therefore PSCPS is an attractive scheduling policy in QoS-guaranteed distributed networks.**

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 design and performance analysis using rate-guaranteed link-sharing scheduling algorithms (see [2] and references therein). However, these sharing algorithms base on "rate" concept to share link resource. Rate-characterization is often satisfactory if the input traffic behaves somewhat like 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 a VBR source with a single parameter. To guarantee packet delay in these cases, the characterization weighting factor is often set to the peak rate, and this generally results in over-allocation of link resource.

Due to the diversed characteristics of multimedia VBR traffic, Cruz [1] proposed the following traffic curve-characterization: Let $R_m(t_1, t_2)$ (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 $R_m(t_1, t_2) \leq b(t_2 - t_1)$ for $t_2 - t_1 \geq 0$. 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 rate with respect 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 should help us allocate resource to VBR connections more efficiently.

On the other hand, Packet-by-Packet Generalized Processor Sharing (PGPS) processor [3] is a very well-known rate-proportional sharing scheme in the literature, and this illustrates the importance of the proportionality concept. In this paper, by combining the concepts of service characterization and proportionality, we propose Packetized Service Curve Proportional Sharing (PSCPS) scheduling algorithm synthesized according to a set of assigned service curves under the *continuous-time* model and *variable* packet-size assumption. Briefly speaking, if PGPS is a proportional sharing scheme based on *weighting factors*, then PSCPS is a proportional sharing scheme based on *weighting functions*. The corresponding feasible service curve allocation condition that serves as an admission control criterion and practical implementation procedure that resembles the *virtual time implementation* of PGPS are also proposed. We then compare PSCPS to other scheduling policies in the literature. Because of its proportional property, PSCPS does not need any traffic pre-regulation as the optimal scheduling policy NPEDF [6]. 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 has a little-reduced schedulable region than that of the optimal case, 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.

II. Service Curve Proportional Sharing Processor

Generalized Processor Sharing (GPS) processor [3], which was first proposed by Demers et. al. under the name of weighted fair queueing, is a very well-known rate-proportional sharing scheme in the literature. Under the assumption that a packet has arrived only after its last bit has arrived, the definition of GPS can be simplified as follows: 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* 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:

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

where ϕ_i now represents a function $\phi_i(t) \triangleq \phi_i \cdot t$ for each

This research was supported by the National Science Council of Taiwan, R.O.C. under the contract NSC-88-2219-E-002-002.

0-7803-5796-5/99/$10.00 © 1999 IEEE

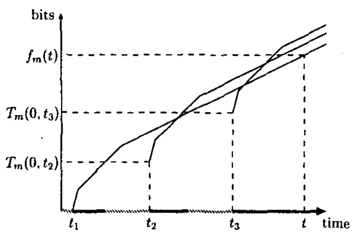

Fig. 1. The computation of $f_m(t)$.

i, $\left(\sum_{j\in B(t)} \phi_j\right)(t)$ 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 sharing instead of rate sharing. 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 the subset that 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 $\left(\sum_{j\in B(t)} S_j\right)(t) \triangleq \sum_{j\in B(t)} S_j(t)$ and the symbol "$\circ$" denotes the composition of functions. Although the above situation is just a special case for SCPS, (3) serves as the fundamental thought of the SCPS processor.

Suppose 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 later. 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 simple discontinuities (i.e. a finite jump) are allowed. According to [1], we say the service curve 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 $T_m(u,t) \ge S_m((t-u)^+)$. Consider a busy period starting at time 0. For each t and connection $m \in B(t)$, we define $f_m(t)$ according to [1], [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 $Q_m(z) = 0$. Briefly speaking, $f_m(t)$ is just the "minimal" amount of service that connection m must obtain before t to achieve its service curve requirement. According to this definition, it can be seen that $f_m(t)$ is (not necessarily strictly) increasing. With the above definition, the service index $\tau_m^s(t) \in \mathbb{R}^3$ of connection $m \in B(t)$ at t is defined by (Figure 2)

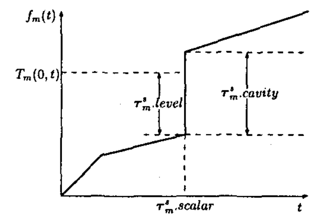

Fig. 2. Definition of $\Psi(f_m, T_m(0,t))$.

$$\tau_m^s(t) \triangleq \Psi(f_m, T_m(0,t)) = (\Psi_1, \Psi_2, \Psi_3), \quad \text{where}$$
$$\tau_m^s(t).scalar = \Psi_1 \triangleq \sup\{y : y \ge 0, f_m(y^-) \le T_m(0,t)\}$$
$$\tau_m^s(t).level = \Psi_2 \triangleq T_m(0,t) - f_m(\tau_m^s(t).scalar^-)$$
$$\tau_m^s(t).cavity = \Psi_3 \triangleq f_m(\tau_m^s(t).scalar) - f_m(\tau_m^s(t).scalar^-)$$

are the three components of $\tau_m^s(t)$. And conversely, Φ is the inverse function of Ψ such that

$$T_m(0,t) \triangleq \Phi(f_m, \tau_m^s(t)) \triangleq f_m(\tau_m^s(t).scalar) - \max\{ (f_m(\tau_m^s(t).scalar) - f_m(\tau_m^s(t).scalar^-) - \tau_m^s(t).level), 0\}. \quad (5)$$

From the about definitions and Figure 2, it can be seen that $\Psi(f_m,\cdot)$ 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 $\tau_m^s(t) \in \mathbb{R}^3$, its order relation must be defined explicitly 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:

- $\tau_h^s(t)$: $\quad \tau_h^s(t) \triangleq \min\{\tau_m^s(t) : m \in B(t)\} \quad (6)$
is the minimum of the service indices of all backlogged connections.
- $B_h(t) \triangleq \{m \in B(t) : \tau_m^s(t) = \tau_h^s(t)\}$, that is the set of backlogged connections whose service indices equal to $\tau_h^s(t)$. The subscript h implies that this set has higher priority to be served.
- $B_l(t) \triangleq B(t)\backslash B_h(t)$, that is the complement set of $B_h(t)$ with respect to $B(t)$. The subscript l implies that this set has lower priority to be served.
- $\tau_l^s(t) \triangleq \min\{\tau_m^s : m \in B_l(t)\}$, 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 $B_l(t)$.
- $E(t) \triangleq \{m : m \notin B(t)\}$

In short, SCPS distributes service among the connections with the smallest service indices as follows.

Definition 1: Given a server with service rate r, Service Curve Proportional Sharing (SCPS) processor is

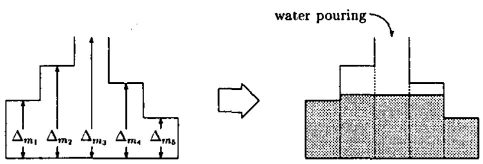

Fig. 3. Service distribution at discontinuous points: water filling model.

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

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

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 $f_{total}(t) \triangleq \left(\sum_{m \in B_h(t)} f_m\right)(t)$, 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 $f_{m_1}, f_{m_2}, \cdots, f_{m_N}$, $m_i \in B_h(t)$ 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 $i \in \{m_1, m_2, \ldots, m_N\}$.

Now Proposition 1 identifies the sufficient condition of service curve allocation under which a SCPS server can simultaneously guarantee the service curve of each connection and the complete proof is presented in [7].

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

III. Packetized Service Curve Proportional Sharing Scheduling

The operation of a SCPS processor is based on the assumption of fluid model. However, in modern packet switching networks, data is transmitted in the format of packet, which is an indivisible unit. Inspired by the Packet-by-Packet Generalized Processor Sharing (PGPS) scheduler, we induce a scheduling algorithm according to the packet departure order in a SCPS system.Let F_p be the time at that packet p departs SCPS. Then we define the Packetized Service Curve Proportional Sharing scheduler.

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

Due to the proportionality of SCPS processors, we have the following lemma, and the proof is in [7].

Lemma 1: Consider two packets p and p' in a SCPS system at time t. Suppose that packet p completes service before packet p' in case 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 hold, which quantitatively measure the difference between the output processes of a SCPS processor and a PSCPS scheduler (developed in [3], [6]).

Theorem 1: Denote F_p and $\hat{F}_p$ the departure time at that 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,

$$\hat{F}_p - F_p \leq L_{max}/r. \quad (8)$$

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,

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

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

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

With the above theorems, we observe that PSCPS scheduler is a close approximation of SCPS processor.

A. Implementation of PSCPS Scheduler

In [3], a concise and efficient scheme called "virtual time implementation" is proposed as a practical implementation of PGPS scheduler. Due to the similarity between "rate proportionality" and "service curve proportionality", PSCPS scheduler also has an efficient "virtual time implementation" as follows. We say an event occurs at each of the following time instants.

1. arrival at SCPS processor.
2. departure from SCPS processor.
3. 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). All these three events may change the set $B_h(t)$, especially when the third kind of event (called the joint event) occurs, at least one connection from $B_l(t)$ joins into $B_h(t)$.

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 $\boldsymbol{S}(t) \in \mathbb{R}^3$ is set to (0,0,0) for all times when the server is idle. Then virtual time $\boldsymbol{S}(t)$ is defined as follows:

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

$$\boldsymbol{S}(t_{j-1}+\tau) = \Psi((\sum_{m \in B_j} f_m), r(t_{j-1}+\tau) - \sum_{m \notin B_j} T_m(0, t_{j-1})) \quad (11)$$

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

Now suppose that the k^{th} packet of connection m arrives at time a_m^k and has length L_m^k. Also associated with this packet is a nonnegative increasing function f_m^k. Then, denote the virtual times at which this packet begins and completes service as $\boldsymbol{S}_m^k$ and $\boldsymbol{F}_m^k$, respectively. Defining $f_m^0(t) \triangleq 0$ for $t < 0$ and ∞ for $t \geq 0$, and $\boldsymbol{F}_m^0 = (0,0,0)$ for all m. Then we determine virtual starting time

$$\boldsymbol{S}_m^k = \max\{\boldsymbol{F}_m^{k-1}, \boldsymbol{S}(a_m^k)\} \quad (12)$$

by the following rules

if $\boldsymbol{S}(a_m^k).level = \boldsymbol{S}(a_m^k).cavity$ and $\Phi(f_m^{k-1}, \boldsymbol{F}_m^{k-1}) \leq f_m^{k-1}(\boldsymbol{S}(a_m^k).scalar)$, then $\boldsymbol{F}_m^{k-1} \leq \boldsymbol{S}(a_m^k)$.
else if $\boldsymbol{S}(a_m^k).level < \boldsymbol{S}(a_m^k).cavity$ and $\Phi(f_m^{k-1}, \boldsymbol{F}_m^{k-1}) \leq \Phi(f_m^{k-1}, \boldsymbol{S}(a_m^k))$, then $\boldsymbol{F}_m^{k-1} \leq \boldsymbol{S}(a_m^k)$.
else
$\boldsymbol{F}_m^{k-1} \geq \boldsymbol{S}(a_m^k)$.

If $\boldsymbol{S}_m^k = \boldsymbol{F}_m^{k-1}$, then $f_m^k(t) = f_m^{k-1}(t)$. Otherwise, if $\boldsymbol{S}_m^k =$

$S(a_m^k)$, then $f_m^k(t)=\min\{f_m^{k-1}(t), S_m(t-a_m^k)+\sum_{i=1}^{k-1}L_m^i\}$. Then define the virtual finishing time F_m^k by

$$F_m^k = \Psi(f_m^k, (\sum_{i=1}^{k} L_m^i)) \qquad (13)$$

The order relation of virtual finishing time $F_m^k \in \mathbb{R}^3$ is determined by its first components. If the first components are identical, then their order is determine by the second components. Two virtual finishing times with identical first and second components are regarded equivalent. Then the packets are served in an 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 among all virtual finishing times of the packets in the system at time t. Also recall that the service index τ_l^s represents the virtual time of nearest joint event. Then from (11), we have:

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

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

where $\min\{F_{min}, \tau_l^s\}$ is determined according to the same rules for determining S_m^k in (12). 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.

B. The Sub-Optimality of PSCPS Scheduling Algorithm

Georgiadis et. al. have shown that non-preemptive earliest deadline first (NPEDF) is the delay-optimal policies among the class of non-preemptive policies (Theorem 4 in [6]) in the sense that if M connections constrained by $b_1(\cdot), \cdots, b_M(\cdot)$ and with maximal tolerable delays $d_1, \cdots, d_M$ are schedulable (no delay violation occurs) under any non-preemptive policy, then these M connections are also schedulable under NPEDF. Suppose L_{max} is the maximal packet length. The schedulable region of NPEDF is the set of the vectors $(d_1, \cdots, d_M)$ with $d_1 \le d_2 \le \cdots \le d_M$ such that [6]

$$L_{max}/r \le d_1$$

$$\sum_{i=1}^{M} b_i(t-d_i)U(t-d_i) + L_{max} \le r\cdot t, \quad L_{max}/r \le t < d_M$$

$$\sum_{i=1}^{M} b_i(t-d_i)U(t-d_i) \le r\cdot t, \quad t \ge d_M,$$

where $U(t)$ is the unit-step function such that $U(t)=1$ for $t>0$ and $U(t)=0$ for $t\le 0$. Now we present the schedulable region of PSCPS in the following proposition.

Proposition 2: Consider a PSCPS processor that has rate r bps and serves M connections constrained by $b_1(\cdot), \cdots, b_M(\cdot)$. If L_{max} is the maximal packet size and the connections require maximum packet delays $d_1, \cdots, d_M$ $(d_1 \le d_2 \le \cdots \le d_M)$, respectively, then the delay requirements can be satisfied if

$$2L_{max}/r \le d_1$$

$$\sum_{i=1}^{M} b_i(t-d_i+\frac{L_{max}}{r})U(t-d_i+\frac{L_{max}}{r})+L_{max} \le r\cdot t, \frac{L_{max}}{r} \le t < d_M$$

$$\sum_{i=1}^{M} b_i(t-d_i+L_{max}/r)U(t-d_i+L_{max}/r) \le r\cdot t, \quad t \ge d_M,$$

Therefore PSCPS has smaller schedulable region than NPEDF ($\frac{L_{max}}{r}$ smaller in each d_i component) and is only sub-optimal in the sense of schedulable region. However, due to its proportionality property, PSCPS does not need any traffic pre-regulation as NPEDF. This characteristic makes PSCPS a "truly work-conserving" scheme and more effective to provide best-effort service for ABR traffic. Therefore, although PSCPS has a little-reduced schedulable region than that of the optimal policy, in real-world multimedia networks where both real-time and best-effort services are provided, PSCPS is even more attractive than NPEDF.

IV. PSCPS in Distributed Environments

In distributed environments such as wireless channels, all mobile nodes are geographically distributed, and the arrival information of each connection is not automatically available to the scheduler. Therefore we must explicitly deal with the exchange of traffic-information between each mobile node and the PSCPS scheduler. The following theorem identifies the sufficient exchange in distributed environments and the proof is in [7].

Proposition 3: The arrival and packet size information of all head packets is sufficient to make the scheduling order decision satisfying the definition of PSCPS scheduling algorithm.

While only utilizing the head packet information, the "virtual time implementation" proposed in Section III-A must be slightly modified. We briefly state this modified implementation below:

When a busy period begins, the first scheduling decision is to serve the node that initiates this busy busy period.
loop:
1. At each decision-making instant, t_{cur}, the scheduler uses (11) to find next SCPS departure event recursively until any of the following two conditions is satisfied:
(a) the SCPS departure corresponds to one of these current head packets and the departure time is earlier than t_{cur}.
(b) the SCPS departure time is greater than t_{cur}.
2. if the above recursion is terminated because of the first condition, serve the head packet and then go to *loop*.
3. otherwise the scheduler serve the head packet with the minimal virtual finishing time F_m^k and then go to *loop*.

According to Proposition 3, in distributed environments, we only provide head packet information to PSCPS scheduler to avoid massive information exchange. Suppose PSCPS scheduler notifies node m to transmit a packet, we can piggyback the next packet's arrival and size information at the end of the transmitting packet. Under this assumption, the head packet arrival time is the only required information. When a packet has a predecessor, this information can be piggybacked. When a node with an empty queue becomes backlogged, the arrival information can be represented as a busy tone. Therefore, piggybacking and busy tone is a pretty good solution to head packet information exchange problem in distributed environments.

V. Simulation Results

In this section, we present a simulation result of PSCPS scheduler in a multiaccess environment (capacity C=150Mbps) with six distinct traffic classes (88 connections):

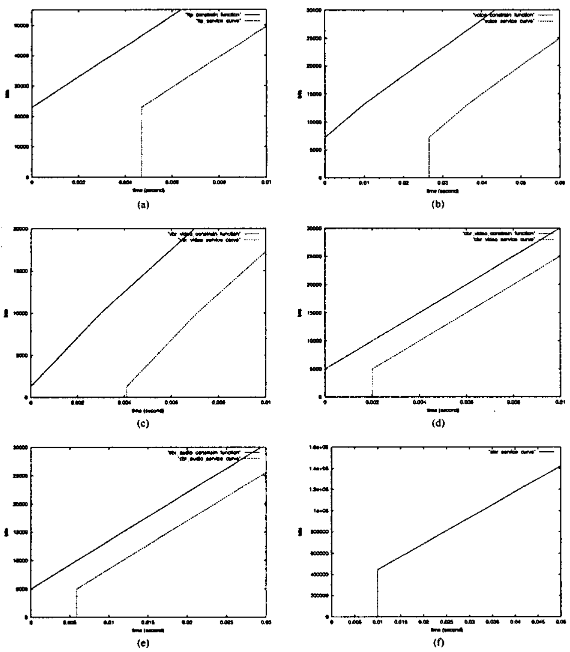

Fig. 4. Constrain function and service curve of a single connection of each class.

- FTP (File Transport Protocol) traffic: There are 7 FTP connections in this class, with packet size 11500 bits.
- voice traffic: There are 30 voice connections in this class. The voice traffic is generated from the three-state Markov model presented in [4], with voice packet size 720 bits.
- VBR video traffic: There are 10 VBR video connections in this class. The arrival processes of all VBR video sources are taken from packetized MPEG-1 outputs of movie "star war", with VBR video packet size 384 bits.
- CBR video traffic: There are 10 CBR video connections in this class. The constant bit rate and packet size are set to 2.5Mbps and 5000 bits, respectively.
- CBR digital audio traffic: There are 30 CBR audio connection in this class. The constant bit rate of these high-quality digital audio sources is assumed to be 850Kbps. The CBR audio packet size is set to 5000 bits.
- ABR data traffic: We use a Poisson process to model the packet generation behavior of the aggregate ABR traffic sources. The Poisson arrival rate and ABR packet size are set to 5000 and 10000 bits, respectively. Hence the average bit rate is 50Mbps. We assume ABR traffic *does not* have packet delay requirements.

The service curves allocated to a single connection of each class are shown in Figure 4. Since the summation of all service curves is still less than $C \cdot t$, we can see that the allocation criterion in Proposition 1 is satisfied. Therefore Proposition 1, Proposition 2 in [7], and Theorem 1 imply that the packet delay in this PSCPS system is bounded by: The maximum horizontal distance between $b(\cdot)$ and $S(\cdot) + L_{max}/C$. The simulation time of this experiment is set to 400 seconds and the simulation program tracks the maximal packet delay of each class of traffic. The resultant maximal delays v.s. theoretical delay bounds are shown in Table I. From these numerical

TABLE I
Simulation results v.s. theoretical bounds.

class	PSCPS max. delay (sec.)	PSCPS bound (sec.)	NPEDF max. delay (sec.)	NPEDF bound (sec.)
FTP	0.001255	0.004776	0.0005067	0.0047
voice	0.01632	0.02657	0.001448	0.0265
VBR video	0.0007261	0.004176	0.0004299	0.0041
CBR video	0.0004564	0.002076	0.0004099	0.0020
CBR audio	0.003128	0.005956	0.000832	0.00588

results, our theoretical delay bounds are justified.

For the purpose of comparison, we also evaluate the performance of the the optimal NPEDF scheduling policy using the same traffic data. The resultant maximal delays v.s. theoretical delay is shown in Table I. We can immediately see that NPEDF indeed results in smaller packet delays for a given set of traffic constrain functions. However, in 400 seconds simulation time, only 978027 ABR packets were served by NPEDF scheduler while 1998416 ABR packets were served by PSCPS scheduler. This important observation shows that PSCPS is more effective to serve ABR traffic and more desirable than NPEDF in real-world multimedia networks providing both real-time and best-effort service.

VI. Conclusions

In this paper, we presented the service-curve proportional sharing concept and proposed an efficient scheme called "virtual time implementation" as a practical implementation of PSCPS scheduler.

It is shown that PSCPS is a suboptimal policy in the class of non-preemptive scheduling policies in terms of schedulable region. Although PSCPS has a little-reduced schedulable region than that of the optimal NPEDF, it has an advantage over the optimal policy: Because of its proportional property, PSCPS policy integrates the regulation and service curve sharing together and therefore can work without extra traffic pre-regulators. This characteristic makes PSCPS a "truly work-conserving" sharing scheme. In real-world multimedia networks where both real-time and best-effort services are provided, PSCPS is even more attractive than NPEDF policy. Consequently, PSCPS is a potential and effective approach for distributed multiple access in multimedia distributed networks.

References

[1] 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.

[2] R. L. Cruz, "SCED+: Efficient management of quality of service of service guarantees," *Proc. IEEE INFOCOM'98.*

[3] 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.

[4] D. J. Goodman and S. X. Wei, "Efficiency of Packet Reservation Multiple Access," *IEEE Trans. Veh. Technol.,* vol. 40, pp. 170-176, Feb. 1991.

[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] 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.

[7] Chia-Sheng Chang, Kwang-Cheng Chen, "Service Curve Proportional Sharing Algorithm for Service-Guaranteed Multiaccess in Multimedia Distributed Networks," submitted for publication.