GENERALIZED PROCESSOR SHARING SCHEDULING ENHANCEMENT THROUGH SESSION DECOMPOSITION

Antonis Panagakis and Ioannis Stavrakakis
Dept. of Informatics, University of Athens, Athens, Greece
{grad0260, istavrak}@di.uoa.gr*

Abstract In this paper a service discipline, which is based on Generalized Processor Sharing (GPS) and aims to improve resource utilization for deterministic guarantees at a single node is presented. According to the proposed scheme original sessions are decomposed into two components which are treated as separate sessions by a GPS scheduler; the work assigned to each session components is mapped back to the original session. A Call Admission Control (CAC) algorithm for the proposed scheme – capable of ensuring typically better and never worse resource utilization compared to that achieved by the optimal CAC applied directly to the sessions – is derived, and some numerical results demonstrating the benefits of the proposed scheme are provided.

1. Introduction

GPS scheduling discipline has been extensively studied in a deterministic ([4], [5], [8], [7]) as well as in a stochastic environment ([2], [9]). Modifications / extensions of the GPS scheduling discipline include ([1], [6]). In this paper the GPS scheduling discipline is indirectly modified in the following sense: GPS is preserved as the work assignment module but the traffic that the scheduler is serving is modified. More specifically, it is proposed that each session be "represented" by two entities – referred to as session components – in the GPS server. The work provided by the GPS server to the components of a session is mapped back to the original session. Thus, the GPS scheduling discipline guarantees a delay bound not on a *per session basis* but on a *per sessions component basis*. The GPS server is configured – through proper weight assignment – in such a way that the same delay bound is guaranteed to both components of a

*This work has been supported in part by the Special Account for Research Grants of the National and Kapodistrian University of Athens and by the General Secretariat for Research and Technology of Greece under the Joint Greek-Italian Scientific and Technological Cooperation Programme.

session. It is shown that if the original session is provided with the sum of the service provided by the scheduler to its two components its QoS requirements are met. The selection of proper components of each session is investigated for the case of (σ, ρ) constrained sessions requiring deterministic delay guarantees at a single node. Such sessions will be referred to as QoS sensitive sessions. In section 2 the proposed system, referred to as DS(Decomposition-based Service) -system, is described. In section 3 a Call Admission Control (CAC) scheme for the proposed scheme is presented. Finally in section 4 some numerical results illustrating the benefits of the proposed scheme are provided.

2. Decomposition based Service Scheme

The QoS sensitive sessions will be assumed to be leaky bucket constrained, (that is, the arrival function of session i $A_i(\tau, t)$ will be bounded as $A_i(\tau, t) \leq \sigma_i + \rho_i(t-\tau), \forall t \geq \tau \geq 0$; σ_i and ρ_i represent the burstiness and the long term maximum mean arrival rate of session i, respectively) and to have a stringent delay requirement, denoted by D_i for session i. Thus, a QoS sensitive session i will be characterized by the triplet (σ_i, ρ_i, D_i); the notation $i \sim (\sigma_i, \rho_i, D_i)$ will be used. The system depicted in Figure 1 will be used to compare the proposed scheme with. The GPS server capacity C_G is assumed to be shared by N QoS sensitive sessions with descriptors $(\sigma_i, \rho_i, D_i), i = 1, \ldots, N$ (with $\sigma_i > 0, \rho_i > 0$) and best effort traffic represented by an additional session. Each session is provided a buffer and the input links are considered to have infinite capacity. This system –referred to as the BETA (BEst Effort Traffic Aware) GPS system in [3], where the optimal CAC scheme for it has been derived – will be referred to as the reference system, and the optimal CAC scheme for this system as the R_CAC (Reference CAC) scheme. The proposed system, which will be referred to as Decomposition-based Service (DS-) system, is depicted in Figure 2. The GPS scheduler is employed by a Virtual System which is provided with an exact replica of each session's traffic. The Virtual System consists of three modules (see Figure 2): (a) the decomposition module responsible for the decomposition of the replicas of the sessions (b) the service module responsible for determining the service provided to each session's components, and (c) the service mapping module which maps the service provided to the sessions components back to the original sessions.

Each session $i \sim (\sigma_i, \rho_i, D_i)$ may be considered as the superposition (aggregation) of two component sessions $i_{(1)} \sim (\sigma_{i(1)}, \rho_i, D_i)$ and $i_{(2)} \sim (\sigma_{i(2)}, 0, D_i)$, $\sigma_{i(2)} = \sigma_i - \sigma_{i(1)}$, which will be referred to as the Long Term (LT-) and the Bursty (B-) component of session i, respectively. For the decomposition of session i a $(\sigma_{i(1)}, \rho_i)$ leaky bucket, with $\sigma_i > \sigma_{i(1)} > 0$, is employed. Session i traffic (replica) traverses the $(\sigma_{i(1)}, \rho_i)$ leaky bucket; the part of the traffic that finds tokens is considered to belong to the LT-component of session i and the rest of session's traffic is considered to belong to the B-component of the

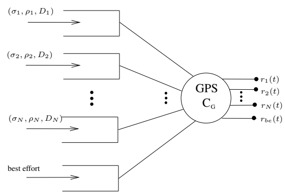

Figure 1. Reference system

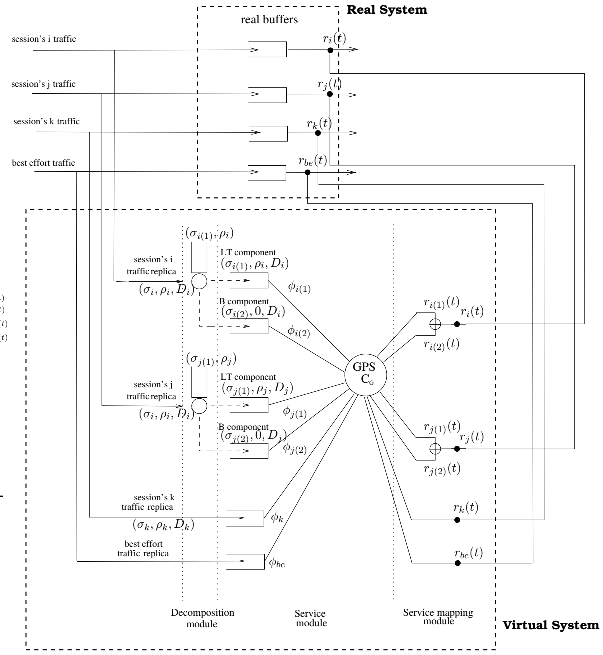

Figure 2. DS-system

session. The two components are buffered in separate virtual buffers (counters) and are assigned weights in proportion to which they are served by the GPS scheduler. Both components of session i have the same delay constraint D_i as the original session i. The real traffic of a session is served at any time instant with a rate equal to the sum of the service rates of its components in the Virtual System. It is noted that not all the sessions are necessarily decomposed into two components; only the sessions that fulfill a specific decomposition criterion, described in section 3, are decomposed.

Claim 1 *If the components of a session are served in such a way that their QoS requirements are satisfied and the real session is served at any time instant with a rate equal to the sum of the service rates of its components, then the QoS requirements of the session are also satisfied.*

Proof: The service provided to the real traffic of each session in any time interval $[t_1, t_2]$ is equal to the sum of the service provided by the Virtual System to the session's components in $[t_1, t_2]$. Let $A_{i(\cdot)}(t)$ $(W_{i(\cdot)}(t))$ denote the cumulative arrivals (service received) at the service module of the Virtual System of (by) the $\cdot$ component of session i in the interval $(0, t]$. Suppose that $i_{(1)}$ and $i_{(2)}$ are served by the scheduler in such a way that $A_{i_{(1)}}(t) \leq W_{i_{(1)}}(t + D_i)$, $\forall t$ and $A_{i_{(2)}}(t) \leq W_{i_{(2)}}(t + D_i)$, $\forall t$, that is the maximum delay experienced by the traffic of $i_{(1)}$ and $i_{(2)}$ is equal to D_i. Since $A_i(t) = A_{i_{(1)}}(t) + A_{i_{(2)}}(t) \leq W_{i_{(1)}}(t + D_i) + W_{i_{(2)}}(t + D_i) = W_i(t + D_i)$, $\forall t$, it is implied that the traffic of session i leaves the system having experienced a maximum delay less than or equal to D_i.

3. The proposed CAC scheme (D_CAC scheme)

In view of Claim 1 it is evident that it suffices to develop a CAC scheme for the Virtual System, since a schedulable traffic mix for the Virtual system is also schedulable for the DS-system whose central element is the Virtual system itself. In addition, the service module of the Virtual System does not differ in any way from the reference system (see Figures 1 and 2), and thus, the Virtual System may be considered as a reference system with a tunable input. Consequently, the CAC scheme for the Virtual System (or equivalently the DS-system) may be considered as a CAC scheme for a system similar to the reference system with an extra degree of freedom, which is the ability to determine sessions components.

The proposed CAC scheme, referred to as the D_CAC scheme (Decomposition-based Call Admission Control scheme), is a generalization of the R_CAC scheme, in the same way as the DS-system is a generalization of the reference system. It may be considered as consisting of two distinct functions: (a) one responsible for determining which sessions of the original traffic mix will be decomposed in the Virtual System and the exact form of their components, and (b) one responsible for determining the weight assignment for the sessions components.

The R_CAC scheme is employed for the determination of the weight assignment of the session components. The other function, which is responsible for determining the session components is such that a session replica is decomposed only if (and in such a way that) this decomposition: (Ca) leads to better (or at least not worse) resource utilization, and (Cb) the session is assigned a total weight (sum of the weights of the session components) no greater than if it were not decomposed.

3.1. Definitions

An all greedy GPS system is defined as a system in which all the sessions are greedy starting at the beginning of a system busy period. According to [4] (Theorem 3) a session experiences its worst case delay in such a system, and thus, **only all greedy systems are examined** in the framework of the CAC problem investigation.

Let $t = 0$ denote the beginning of a system busy period in an all greedy system. Let e_i denote the backlog clearing time of session i, that is, $e_i = max\{t > 0 : Q_i(t) > 0\}$, where $Q_i(t) = A_i(0,t) - W_i(0,t)$ denotes the amount of session i traffic queued in the server at time t. Let $N_i(0,t)$ – referred to as session i requirements – denote the minimum amount of service that must be provided to the QoS sensitive session i over the interval $(0,t]$ in order to ensure that the delay constraint for the session i is met; then:

$$N_i(0,t) = \begin{cases} \sigma_i + \rho_i(t - D_i) & t \geq D_i \\ 0 & t < D_i \end{cases} \tag{1}$$

Under the all greedy system assumption, all QoS sensitive sessions are backlogged at time $t = 0^+$. Let $\mathcal{B}(t)$ denote the set of sessions that are backlogged in the interval $(0, t]$ and let $\mathcal{E}(t)$ denote the set of sessions which have emptied their backlog before time t. Let $\hat{C}(t) \triangleq \frac{C_G - \sum_{j\in\mathcal{E}(t)} \rho_j}{1-\sum_{j\in\mathcal{E}(t)} \phi_j}$ be referred to as the Normalized Backlogged Sessions Allocated (NBSA) bandwidth. Let $\{\mathrm{b}_i\}_{i=1}^{L}$, $L \leq N$ denote the ordered set of distinct backlog clearing times and let $\mathrm{b}_0 = 0$ be the beginning of the system busy period. For two consecutive backlog clearing times b_{j-1} and b_j, $\hat{C}(\mathrm{b}_{j-1}^+) = \hat{C}(\mathrm{b}_j^-)$. Treating the NBSA bandwidth as a left continuous function implies that $\hat{C}(\mathrm{b}_j) = \hat{C}(\mathrm{b}_j^-)$ and

$$\hat{C}(t) = (C_G - \sum_{k\in\mathcal{E}(\mathrm{b}_j)} \rho_k)(1 - \sum_{k\in\mathcal{E}(\mathrm{b}_j)} \phi_k)^{-1} \quad \forall t \in (\mathrm{b}_{j-1}, \mathrm{b}_j] \tag{2}$$

$\hat{C}(t)$ is an increasing function of time since it preserves a constant value between two consecutive backlog clearing times and $\hat{C}(\mathrm{b}_j) < \hat{C}(\mathrm{b}_j^+)$ (the proof may be found in [3]). Let $\hat{W}(\mathrm{b}_{j-1}, \mathrm{b}_j) \triangleq \hat{C}(\mathrm{b}_{j-1}^+)(\mathrm{b}_j - \mathrm{b}_{j-1})$ be referred to as the Normalized Backlogged Sessions Allocated (NBSA) work. Then, session $i \in \mathcal{B}(\mathrm{b}_j)$ is allocated an amount of work equal to $\phi_i \hat{W}(\mathrm{b}_{j-1}, \mathrm{b}_j)$ over $(\mathrm{b}_{j-1}, \mathrm{b}_j]$.

Let $\tau_0 = 0$ and let $\{\tau_m\}_{m=1}^{M}, M \leq 2N$ denote the ordered set of distinct time instances which coincide with either the delay bound or the backlog clearing time of some session. The time instant τ_m will be referred to as the m^{th} ordered checkpoint. The following quantities are defined for session $i, i = 1, \ldots, N$ at all checkpoints τ_m such that $D_i \leq \tau_m < e_i$. (For sessions with $\rho_i = 0$ (that is, for the B-components) these quantities are defined (only) for $\tau_m = D_i$.)

$$\phi_i^-(\tau_m) = \frac{N_i(0, \tau_m)}{\hat{W}(0, \tau_m)}, \quad \phi_i^+(\tau_m) = \frac{\rho_i}{\hat{C}(\tau_m^+)} \tag{3}$$

where $\hat{W}(0, \tau_m) = \sum_{k=1}^{m} \hat{W}(\tau_{k-1}, \tau_k)$. The quantity $\phi_i^-(\tau_m)$ represents the fraction of the total NBSA work that must be assigned to session i in order for session i to be assigned work exactly equal to $N_i(0, \tau_m)$ up to time τ_m, given that session i has not emptied its backlog before time τ_m. The quantity $\phi_i^+(\tau_m)$ represents the percent of the NBSA bandwidth just after τ_m that must be assigned to session i in order for session i to be served with a rate equal to ρ_i. This is sufficient to ensure that its requirements are satisfied for $t > \tau_m$, if session i is assigned work at least equal to $N_i(0, \tau_m)$ up to time τ_m.

3.2. Description of the D_CAC scheme

3.2.1 Summary of the R_CAC scheme. The pseudocode of the R_CAC is exactly the same as the pseudocode of the D_CAC scheme in Figure 3 except step $B4.0$. The R_CAC scheme emulates the time evolution of the all greedy system and determines the weights of the sessions in stages. At each stage a checkpoint τ_j is considered. The first checkpoint coincides with the minimum delay bound of the QoS sensitive sessions. At each checkpoint τ_j the following main sets of sessions are used (all other sets in the description of the R_CAC scheme are temporary sets): (-) Nex_j: contains sessions i which have not been examined at a previous or at the current checkpoint (-) $Empty_j$: contains sessions which have emptied their backlog at the current or a previous checkpoint; their weights have been determined at a previous checkpoint. (-) Phi_j: contains sessions whose weights have been determined, but have not emptied their backlog yet (-) $Trans_j$: contains sessions which have been examined at a previous (and at the current) checkpoint, but their weights have not been determined yet.

According to the R_CAC scheme at each τ_j: (-) The sessions that empty their backlog at τ_j are determined and are included in the set $Empty_j$ (-) Sessions i with $D_i = \tau_j$ are examined and if $\phi_i^-(\tau_j) \geq \phi_i^+(\tau_j)$ session i is assigned a weight $\phi_i = \phi_i^-(\tau_j)$ and is included in the set Phi_j, else session i is included in the set $Trans_j$ (-) Sessions i in $Trans_{j-1}$ are examined and if $\phi_i^-(\tau_j) \geq \phi_i^+(\tau_j)$ session i is assigned a weight $\phi_i = \phi_i^-(\tau_j)$ and is included in the set Phi_j, else session i is included in the set $Trans_j$ (-) The next checkpoint is determined as:

$$min\left\{\left(\begin{array}{c}\text{minimum delay bound of sessions}\\\text{that have not been examined yet}\end{array}\right),\left(\begin{array}{c}\text{minimum virtual clearing time}\\\text{of sessions belonging to } Phi_j\end{array}\right)\right\}$$

where the virtual clearing time of session i at τ_j is the backlog clearing time of session i assuming that no other session is going to empty its backlog before session i does. (-) If $\tau_j = \infty$, then all sessions i in $Trans_{j-1}$ are assigned weight equal to $\phi_i = \phi_i^+(\tau_{j-1})$. (-) The algorithm stops if the assignment is not feasible, or when the weights for all QoS sensitive sessions have been determined. The best effort session is assigned the remaining weight. The following statement is provided for future reference.

Statement 1 *Under the R_CAC scheme, each session i is assigned a weight ϕ_i, where (a) $\phi_i = \phi_i^-(\tau_{d_i})$ if $\phi_i^-(\tau_{d_i}) \geq \phi_i^+(\tau_{d_i})$, and generally (b) $\phi_i \geq \phi_i^-(\tau_{d_i})$*

3.2.2 Overview of the D_CAC scheme. The D_CAC scheme pseudocode is provided in Figure 3. It is exactly the same as the R_CAC scheme description in [3] except step $B4.0$ of the description where the Decomposition Function is included.

The D_CAC scheme allocates weights to the sessions exactly as the R_CAC scheme would; that is, the time evolution of the all greedy system is emulated and the weights of the sessions are determined in stages based on the same principles as the R_CAC scheme. The difference is that at each checkpoint and after the set of sessions (or sessions components) that empty their backlog at the specific checkpoint has been determined (step B.3 in Figure 3), instead of investigating whether the weights of some sessions can be determined at the specific checkpoint, as would happen according to the R_CAC scheme, the Decomposition Function is called (step B.4.0 in Figure 3).

The Decomposition Function, whose description is provided in Figure 4, examines the set of sessions that would be examined for the first time at the current checkpoint τ_j according to the R_CAC scheme, (step A.1 in Figure 4). Based on the decomposition criterion, explained in detail in the next section, it determines a subset of sessions that should be decomposed (set B_j, which is determined at steps A.2-B.3 in Figure 4). The exact form of the components is determined (step C in Figure 4) and two sets containing the LT-and the B-components of the sessions to be decomposed are created (sets $B_{j(1)}$ and $B_{j(2)}$ at step D in Figure 4). The B-components of the sessions to be decomposed are assigned such weights (step E in Figure 4) that they empty their backlog at τ_j and are included in the set of the sessions that empty their backlog at τ_j (step F in Figure 4). Finally the set of sessions to be examined at the current checkpoint by the weight assignment function is properly modified, that is, the sessions to be decomposed are replaced by their LT-components (step G in Figure 4).

3.2.3 Description of the decomposition criterion.

Intuitive consideration. In this section, and for the sake of simplicity of the presentation, it is assumed that only one session of the (original) traffic mix is decomposed in the Virtual System of the DS-system.

Initially it is assumed that the DS-system employees the R_CAC scheme, or equivalently a degenerated version of the D_CAC scheme where the Decomposition Function is never called. The execution of the R_CAC scheme for a specific checkpoint τ_j is considered. Assume that among the sessions that are examined for the first time at τ_j, and for at least one of them, say session k with $D_k = \tau_j$, the condition

$$\phi_k^-(\tau_j) > \phi_k^+(\tau_j) \tag{4}$$

is fulfilled. Due to (4), and according to Statement 1(a), session k would be assigned a weight equal to $\phi_k = \phi_k^-(\tau_j)$ (assuming that such an assignment is feasible). However, the fact that $\phi_k > \phi_k^+(\tau_j)$ implies that session k would be served with a rate greater than ρ_k after τ_j, that is, session k would receive more work (consume more bandwidth) than that required for its QoS requirements to be met. *The rationale behind the decomposition concept is to reduce this effect by decomposing session k into two components and assigning proper weights to them, such that session's k requirements are just satisfied at and after τ_j.*

In order to illustrate this, assume that a different CAC scheme is employed by the DS-system, whose only difference compared to the R_CAC scheme is that before examining the checkpoint coinciding with the delay bound of session k, session k is replaced by its two components; that is, session k is decomposed into two components of the form $(\sigma_{k(1)}, \rho_k, D_k)$ and $(\sigma_{k(2)}, 0, D_k)$, $\sigma_{k(1)} + \sigma_{k(2)} =$

σ_k, the LT-component and the B-component of session k, respectively. For the two components of session k the following hold.

If the B-component of session k is assigned a weight equal to $\phi_{k(2)} = \phi^-_{k(2)}(\tau_j)$ (which is exactly the weight that would by assigned under the R_CAC scheme (see Statement 1(a))) its QoS requirements are met, and it empties its backlog at τ_j (and thus, should be included into the set $Empty_j$).

The LT-component of session k is of the form $(\sigma_{k(1)}, \rho_k, D_k)$. Given (4) and the fact that the B-component of session k empties its backlog at τ_j it is feasible to select such $\sigma_{k(1)}$ that (the feasibility of this condition follows directly from equation (4) and is discussed in the sequel):

$$\phi^-_{k(1)}(\tau_j) = \phi^+_{k(1)}(\tau_j) \tag{5}$$

Assigning weight equal to $\phi_{k(1)} = \phi^-_{k(1)}(\tau_j) = \phi^+_{k(1)}(\tau_j)$ to the LT-component (which is the weight that would be assigned under the R_CAC scheme (see Statement 1(a))) is sufficient to guarantee its QoS requirements.

The total weight assigned to the two components of session k is:

$$\phi_{k(1)} + \phi_{k(2)} = \frac{\sigma_{k(1)}}{\hat{W}(0,\tau_j)} + \frac{\sigma_{k(2)}}{\hat{W}(0,\tau_j)} = \frac{\sigma_k}{\hat{W}(0,\tau_j)} \tag{6}$$

which is equal to the weight that session k would be assigned if it were not decomposed, implying that (Cb) holds. Next it is shown that (Ca) also holds.

Let $\hat{\mathrm{C}}_k(\tau_j^+)$ ($\hat{C}(\tau_j^+)$) denote the NBSA bandwidth right of τ_j if session k is (not) decomposed. By definition:

$$\hat{C}(\tau_j^+) = \frac{C_G - \sum_{i\in Empty_j}\rho_i}{1-\sum_{i\in Empty_j}\phi_i},\ \hat{\mathrm{C}}_k(\tau_j^+) = \frac{C_G - \sum_{i\in Empty'_j}\rho_i}{1-\sum_{i\in Empty'_j}\phi_i} \tag{7}$$

where $Empty_j$ is the set of sessions that empty their backlog at or before τ_j and $Empty'_j = Empty_j \cup k_{(2)}$. For a non degenerated decomposition (i.e. $\sigma_{k(2)} > 0 \Rightarrow \phi_{k(2)} > 0$) $\hat{\mathrm{C}}_k(\tau_j^+) > \hat{C}(\tau_j^+)$ holds. An increased value of the NBSA bandwidth implies an increased value of the NBSA work at future checkpoints. In addition, at τ_j there may be some sessions for which $\phi^-_{\cdot}(\tau_j) \geq \phi^+_{\cdot}(\tau_j)$ holds if session k is decomposed while $\phi^-_{\cdot}(\tau_j) < \phi^+_{\cdot}(\tau_j)$ holds if session k is not decomposed ($\phi^+_{\cdot}(\tau_j)$ depends on the value of the NBSA bandwidth right of τ_j, which depends on whether session k is decomposed). Thus all sessions that will be assigned weight at or after τ_j will be assigned a weight less than or equal to that if session k were not decomposed, allowing the best effort traffic to be assigned a greater or equal weight, and implying that (Ca) holds. As far as condition (5) is concerned, it can be easily verified that:

$$\phi^-_{k(1)}(\tau_j) = \phi^+_{k(1)}(\tau_j) \Leftrightarrow \sigma_{k(2)} = \left(\phi^-_k(\tau_j) - \phi^+_k(\tau_j)\right)\hat{W}(0,\tau_j)F^{-1} \tag{8}$$

where $F = 1 - \frac{\phi^+_k(\tau_j)}{1-\sum_{i\in Empty_j}\phi_i}$ is a positive quantity, since it has been assumed that the assignment is feasible (see [10]). Due to (4) $\sigma_{k(2)} > 0$,and since it has been assumed that the assignment is feasible, $\sigma_{k(2)} < \sigma_k$ holds (see [10]). Thus, a proper value of $\sigma_{k(1)} = \sigma_k - \sigma_{k(2)}$ (satisfying equation (5) with $\sigma_k > \sigma_{k(1)} > 0$) exists.

General case. Let $\{D^k\}_{k=1}^{M}$ denote the ordered set of distinct delay bounds of the sessions consisting the traffic mix and let L^k denote the set of sessions with delay bound equal to D^k. If at most one session from each delay class were to be decomposed, the procedure described in the previous section could be applied at each checkpoint coinciding with the delay bound of some delay class. More specifically, at each checkpoint τ_j such that $\tau_j = D^k$, for some k, a session in L^k, say session i, satisfying the condition of equation (5) ($\phi_i^-(\tau_j) > \phi_i^+(\tau_j)$) would be a candidate for decomposition; for the selected session the relations presented in the previous section could be applied in order to determine the exact form and the weight assignment of its components; conditions (Ca) and (Cb) would hold for each decision made at each checkpoint. The execution of the algorithm could continue since the produced (until τ_j) results are independent of whether a session with delay bound equal to τ_j is decomposed or not. The only additional required step is that the B-component of each (at τ_j) decomposed session should be included into the set $Empty_j$. The decomposition procedure described in steps A.2–B.3 in Figure 4 is an extension of the criterion of equation (5) to the case of a set of sessions. More specifically, assume that the D_CAC scheme is applied and some checkpoint τ_j, which coincides with the delay bound of sessions in L^k (where k is such that $\tau_j = D^k$), is examined. In addition, it is assumed that sessions with delay bound less than τ_j have already been examined and that generally all needed parameters have been computed. It is investigated which sessions in L^k should be decomposed.

It is assumed that the weight assignment for the sessions in L^k is *"best case feasible"*, that is, the sum of the weights allocated by the algorithm would be less than one if all the sessions i in L^k were assigned weight equal to $\phi_i^-(\tau_j)$ (minimum weight that could be assigned (see Statement 1(b))), else there is no feasible weight assignment independently of whether some sessions i, i in L^k, are decomposed or not. The following claims, which are a straightforward extension of the discussion in the previous section, hold (the proofs may be found in [10]).

Claim 2 *Consider a set of sessions* $\mathbf{B} \subseteq L^k$ *with* $D^k = \tau_j$ *(that is,* $D_i = \tau_j, \forall i \in \mathbf{B}$*) and suppose that the system of equations*

$$\overline{\phi_{i(1)}^-(\tau_j)} = \overline{\phi_{i(1)}^+(\tau_j)}, \quad i \in \mathbf{B} \tag{9}$$

$$\overline{\phi_{i(1)}^-(\tau_j)} = \sigma_{i(1)}/\hat{W}(0,\tau_j), \quad \overline{\phi_{i(1)}^+(\tau_j)} = \rho_i/\overline{\hat{\mathrm{C}}_{\mathbf{B}}(\tau_j^+)}, \ \overline{\phi_{i(2)}^-(\tau_j)} = \sigma_{i(2)}/\hat{W}(0,\tau_j)$$

$$\sigma_{i(1)} + \sigma_{i(2)} = \sigma_i \quad \forall i \in \mathbf{B}, \ \overline{\hat{\mathrm{C}}_{\mathbf{B}}(\tau_j^+)} = \frac{C_G - \sum_{i \in Empty_j} \rho_i}{1 - \sum_{i \in Empty_j} \phi_i - \sum_{i \in \mathbf{B}} \overline{\phi_{i(2)}^-(\tau_j)}} \tag{10}$$

has a solution such that $\sigma_{i(1)} > 0$, $\sigma_{i(2)} > 0, \quad \forall i \in \mathbf{B}$.

If each session i in $\mathbf{B}$ is decomposed into two components of the form $(\sigma_{i(1)}, \rho_i, D_i)$ and $(\sigma_{i(2)}, 0, D_i)$, $\forall i \in \mathbf{B}$, and these components are assigned weights $\phi_{i(1)} = \overline{\phi^-_{i(1)}(\tau_j)} = \overline{\phi^+_{i(1)}(\tau_j)}$ and $\phi_{i(2)} = \overline{\phi^-_{i(2)}(\tau_j)}$ (as they would by the R_CAC scheme), then this decomposition (a) leads to better (or at least not worse) resource utilization, and (b) no session is assigned a total weight greater than if the sessions belonging to $\mathbf{B}$ were not decomposed. It is assumed that no other session in L^k is decomposed.

Claim 3 *Consider a set of sessions $\mathbf{B}$ for which the assumption of claim 2 hold. If for a session $p \in L^k$*

$$\phi^-_p(\tau_j)(1 - \sum_{i \in Empty_j} \phi_i - \sum_{i \in \mathbf{B}} \phi^+_i(\tau_j)) > \phi^+_p(\tau_j)(1 - \sum_{i \in Empty_j} \phi_i - \sum_{i \in \mathbf{B}} \phi^-_i(\tau_j)) \quad (11)$$

then the assumption of Claim 2 hold for the set $\mathbf{B} \cup \{p\}$.

In addition, assuming that $\mathbf{B}$ is the subset of L^k containing all the sessions in L^k to be decomposed (as described in Claim 2), the inclusion of session p (and only session p) in the set of the sessions to be decomposed fulfils the following conditions (a) leads to better (or at least not worse) resource utilization, and (b) no session in $\mathbf{B} \cup p$ is assigned a total weight greater than if only the sessions in $\mathbf{B}$ were decomposed.

In view of the above claims the set of sessions to be decomposed at τ_j is determined as follows. Given a set of sessions to be decomposed, Claim 3 determines if a specific session should be added to the set of sessions to be decomposed. From Claim 3 (for $\mathbf{B} = \emptyset$) it follows directly that the set of sessions for which $\phi^-_i(\tau_j) > \phi^+_i(\tau_j)$ may be used as an initial set of sessions to be decomposed. Claim 2 may be used to derive the relations needed for the determination of the sessions components.

4. Numerical results

In this section a numerical example is provided. The traffic mix consists of five sessions (see Table 1a). The capacity of the link is equal to 1 and all weights are normalized to sum to 1. The weights assigned to the sessions in the reference system (under the R_CAC scheme) are denoted as ϕ_i in Table 1b. For the DS-system, where some sessions (replicas) are decomposed and the sessions components are of the form $(\sigma_{i(1)}, \rho_i, D_i)$ and $(\sigma_{i(2)}, 0, D_i)$, the (computed by the D_CAC scheme) weights of the components are given in Table 1b as $\phi_{i(1)}$ and $\phi_{i(2)}$ for the LT- and the B-component of each session, respectively. The parameters $\sigma_{i(1)}$ and $\sigma_{i(2)}$ of sessions components are also given in Table 1b. It is noted that session 2 is not decomposed in the Virtual System, and thus, session 2 is represented by only one component. From Table 1b it is easily concluded that: (a) in the DS-system no session is assigned a (total) weight greater than that in the reference system, and (b) the sum of the weights assigned to the QoS sensitive sessions in the DS-system ($\sum_{i,j} \phi_{i(j)} = 0.9423$) is less than that in the reference system ($\sum_i \phi_i = 0.9957$), or equivalently, the weight assigned to the best effort session is greater in the DS-system. It is noted that the specific traffic mix is schedulable in the DS-system even for a link capacity equal to

Table 1a. Sessions parameters

i	σ_i	ρ_i	D_i
1	2	.25	6.8
2	1	.2	8
3	1	.05	8
4	3	.2	11.4
5	2	.15	14.2

Table 1b. Sessions weights and components parameters

session id i	1	2	3	4	5
ϕ_i	0.2942	0.1732	0.1250	0.2632	0.1401
$\sigma_{i(1)}$	1.6	1	0.3474	1.8368	1.7619
$\sigma_{i(2)}$	0.4	-	0.6526	1.1632	0.2381
$\phi_{i(1)}$	0.2353	0.1473	0.0430	0.1527	0.1123
$\phi_{i(2)}$	0.0589	-	0.0809	0.0967	0.0152
$\phi_{i(1)} + \phi_{i(2)}$	0.2942	0.1473	0.1239	0.2494	0.1275

0.9355 (with $\sum_{i,j} \phi_{i(j)} = 0.9999$) , while it is not schedulable in the reference system for a link capacity equal to or less than 0.9955.

References

[1] C.-S. Chang and K.-C. Chen. Service curve proportional sharing algorithm for service-guaranteed multiaccess in integrated-service distributed networks. *In Proceedings of GLOBECOM '99*, pages 1340 –1344, 1999.

[2] A. Elwalid and D. Mitra. Design of generalized processor sharing schedulers which statistically multiplex heterogeneous qos classes. *In Proceedings of IEEE INFOCOM '99*, pages 1220–1230, March 1999.

[3] A. Panagakis and I. Stavrakakis. Optimal call admission control under generalized processor sharing scheduling. *In Proceedings of IWQoS '01*, 2001.

[4] A.K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[5] A.K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[6] F. Toutain. Decoupled generalized processor sharing: A fair queueing principle for adaptive multimedia applications. *In Proceedings of INFOCOM '98*, pages 291–298, 1998.

[7] Z.L. Zhang, Z. Liu, J. Kurose, and D. Towsley. Call admission control schemes under generalized processor sharing scheduling. *The Journal of Telecommunication Systems, Modeling, Analysis, Design, and Management*, 7(1), July 1997.

[8] Z.L. Zhang, Z. Liu, and D. Towsley. Closed-form deterministic end-to-end performance bounds for the generalized processor sharing scheduling discipline. *Journal of Combinatorial Optimization*, 1(4), 1998.

[9] Z.L. Zhang, D. Towsley, and J. Kurose. Statistical analysis of the generalized processor sharing scheduling discipline. *In Proceedings of ACM SIGCOMM*, pages 68–77, 1994.

[10] A. Panagakis and I. Stavrakakis. Generalized Processor Sharing Scheduling Enhancement through Session Decomposition. *Technical report*, http://www.di.uoa.gr/ $\sim$ apan/gpsesd.ps.ps

```
Determine_π(C_G, Nex_0)
A. j=0, Trans_0=Phi_0=Empty_0=∅, Nex_0 = QoS, τ_0 = 0
B. repeat (B1.-B6.) until (φ_i = φ_i^π  ∀i ∈ Nex_0)
B1. j = j + 1, τ_j = min{ min_{i∈Nex_{j-1}} D_i , min_{i∈Phi_{j-1}} vct_i(τ_{j-1})}
B2. If (τ_j = ∞) then ∀i ∈ Trans_{j-1} {φ_i = φ_i^π = φ_i^+(τ_{j-1}), if Σ_{i∈QoS} φ_i ≥ 1 then error},
goto C.
B3. PE_j = {i ∈ Phi_{j-1} : vct_i(τ_{j-1}) = τ_j}, Empty_j = Empty_{j-1} ∪ PE_j
B4.0 if ∃ i ∈ Nex_{j-1} : τ_{d_i} = τ_j then Decompose_at_τ_j
/*The procedure responsible for determining if the sessions whose delay bound coincides with
τ_j should be decomposed is applied */
B4. TP_j = {i ∈ Trans_{j-1} : φ_i^-(τ_j) ≥ φ_i^+(τ_j)}, NP_j = {i ∈ Nex_{j-1} : D_i = τ_j, φ_i^-(τ_j) ≥
φ_i^+(τ_j)}, NT_j = {i ∈ Nex_{j-1} : D_i = τ_j, φ_i^-(τ_j) < φ_i^+(τ_j)}
B5. Nex_j = Nex_{j-1} \ (NP_j ∪ NT_j), Trans_j = (Trans_{j-1} \ TP_j) ∪ NT_j
B6. ∀i ∈ NP_j ∪ TP_j {φ_i = φ_i^π = φ_i^-(τ_j), if Σ_{i∈QoS} φ_i ≥ 1 then error}, Phi_j =
(Phi_{j-1} \ PE_j) ∪ NP_j ∪ TP_j
C. φ_be = 1 − Σ_{i∈QoS} φ_i
```

Figure 3. Description of the D_CAC scheme.

```
Decompose_at_τ_j  /*Decomposition Function*/
A.1 New_j = {i ∈ Nex_{j-1} : τ_{d_i} = τ_j}, if Σ_{i∈New_j} φ_i^-(τ_j) ≥ 1 − Σ_{i∈{Empty_j ∪ Phi_{j-1}}} φ_i then error
/*New_j is the set of sessions whose delay bound coincides with τ_j. The algorithm stops, if
the assignment for the sessions in New_j is not ‘‘best case feasible".*/
A.2 k = 0, B_j^0 = {i ∈ New_j : φ_i^-(τ_j) > φ_i^+(τ_j)}, A_j^0 = New_j \ B_j^0
/*B_j^0 is the initial set of sessions to be decomposed at τ_j.*/
B. do    k = k + 1,  B_j^k = B_j^{k-1},   A_j^k = A_j^{k-1}
/*loop is executed until all the sessions to be decomposed that can be determined at τ_j are
determined. k is the iteration variable. B_j^k is the set of sessions to be decomposed at
the k-th iteration. A_j^k is the set of sessions to be examined. At the start of the k-th
iteration these sets coincide with the respective sets as formed at the previous iteration.*/
B1. ∀i ∈ A_j^k {If φ_i^-(τ_j)(1 − Σ_{i∈Empty_j} φ_i − Σ_{i∈B_j^k} φ_i^+(τ_j)) > φ_i^+(τ_j)(1 − Σ_{i∈Empty_j} φ_i − Σ_{i∈B_j^k} φ_i^-(τ_j))
then B_j^k = B_j^k ∪ {i}, A_j^k = A_j^k \ {i} } /*each time that a session that fulfills the
decomposition criterion is determined appropriate modification of the sets takes place*/
B2. B_j = B_j^k /*B_j is the set of sessions to be decomposed that is determined at τ_j*/
B3. until (B_j^k = B_j^{k-1}) /*if no additional session is determined the search stops*/
C. ∀i ∈ B_j   σ_{i(2)} = Ŵ(0, τ_j) (φ_i^-(τ_j) − φ_i^+(τ_j)(1 − Q)),   ∀i ∈ B_j,   σ_{i(1)} = σ_i − σ_{i(2)}
where Q = ( Σ_{i∈B_j} φ_i^-(τ_j) − Σ_{i∈B_j} φ_i^+(τ_j))(1 − Σ_{i∈Empty_j} φ_i − Σ_{i∈B_j} φ_i^+(τ_j))^{-1}
/*determination of the σs of the B- and the LT-components*/
D. B_{j(1)} = {i_1 = (σ_{i(1)}, ρ_i, D_i) : i ∈ B_j},   B_{j(2)} = {i_2 = (σ_{i(2)}, 0, D_i) : i ∈ B_j} ,
QoS = {QoS \ B_j} ∪ B_{j(1)} ∪ B_{j(2)}
/*B_{j(1)} (B_{j(2)}) contains all the LT-(B) components of the sessions for which the decision to
be decomposed is made at τ_j*/
E. ∀i ∈ B_{j(2)}  {φ_{i(2)} = φ_{i(2)}^-(τ_j),  if  Σ_{i∈QoS} φ_i ≥ 1 then error }
/*weight assignment of the B-components.*/
F. Empty_j = Empty_j ∪ B_{j(2)}
/*since these B-components empty their backlog at τ_j, set Empty_j is updated*/
G. Nex_{j-1} = {Nex_{j-1} \ B_j} ∪ B_{j(1)}
/*Proper modification of Nex_{j-1} takes place so that the execution of the algorithm can
continue without modification. The LT-components of the sessions to be decomposed will be
examined by the algorithm as if they were original sessions.*/
```

Figure 4. Description of the Decomposition Function