Optimal Call Admission Control on a Single Link with a GPS Scheduler

Antonis Panagakis†, Nandita D‡, Ioannis Stavrakakis†, Joy Kuri‡

Abstract— **In this paper the problem of Call Admission Control (CAC) is considered for leaky bucket constrained sessions with deterministic service guarantees (zero loss and finite delay bound), served by a Generalized Processor Sharing scheduler at a single node in the presence of best effort traffic. Based on an optimization process a CAC algorithm capable of determining the (unique) optimal solution is derived. The derived algorithm is also applicable, under a slight modification, in a system where the best effort traffic is absent and is capable of guaranteeing that if it does not find a solution to the CAC problem, then a solution does not exist. The numerical results indicate that the CAC algorithm can achieve a significant improvement on bandwidth utilization as compared to a (deterministic) effective bandwidth-based CAC scheme.**

I. Introduction

The Generalized Processor Sharing (GPS) scheduling discipline has been widely considered to allocate bandwidth resources to multiplexed traffic streams. Its effectiveness and capabilities in guaranteeing a certain level of Quality of Service (QoS) to the supported streams in both a stochastic ([5], [6], [4]) and deterministic ([1], [2], [3], [7]) sense have been investigated. Traffic management based on deterministic guarantees is expected to lead to lower network resource utilization compared to that under stochastic guarantees. Nevertheless, such considerations are necessary when deterministic guarantees are required by the applications. In addition they can provide valuable insight and methodology for the consideration of stochastic guarantees.

Under the GPS scheduling discipline traffic is treated as an infinitely divisible fluid. A GPS server that serves N sessions is characterized by N positive real numbers $\phi_1, ..., \phi_N$, referred to as weights. These weights affect the amount of service provided to the sessions (or, their bandwidth shares). More specifically, if $W_i(\tau, t)$ denotes the amount of session i traffic served in a time interval $(\tau, t]$ then the following relation will hold for any session i that is continuously backlogged in the interval $(\tau, t]$; session i is considered to be backlogged at time t if a positive amount of that session traffic is queued at time t.

$$\frac{W_i(\tau,t)}{W_j(\tau,t)} \geq \frac{\phi_i}{\phi_j}, j = 1, 2, ...N \qquad (1)$$

†Dept. of Informatics, University of Athens, Athens 157-84, Greece. Email: {grad0260, istavrak}@di.uoa.gr. This work has been supported in part by the Special Account for Research Projects of the National and Kapodistrian University of Athens and the IST Programme under contract IST-1999-10160 (VideoGateway).

‡Centre for Electronics Design and Technology, Indian Institute of Science, Bangalore 560012, India. E-mail: dnandita@stanford.edu, kuri@cedt.iisc.ernet.in

This paper is the result of the merger of two manuscripts submitted by the authors of [15] and [16].

In their seminal papers [1], [2] on GPS, Parekh and Gallager have analyzed the GPS scheduling discipline in a deterministic setting where the traffic of each session is regulated by a leaky bucket regulator. In the single node and the multiple node cases they obtained closed form expressions for bounds on the delay and backlog for a certain class of GPS schedulers called Rate Proportional Processor Sharing (RPPS) schedulers. In [7], Zhang et al obtained closed-form expressions for the end-to-end performance bounds for a broader class of GPS networks known as Consistent Relative Session Treatment (CRST) GPS networks. In [12], Yaron and Sidi studied GPS networks with exponentially bounded burstiness arrivals. In [5] Zhang et al investigated the behavior of GPS in a stochastic setting. The above mentioned papers derive delay and backlog bounds for a session in GPS schedulers, given a particular weight allocation for the session.

The inverse problem of mapping the QoS requirements of the sessions to weight allocations in GPS schedulers is of practical importance. Kesidis et al in [13] and Zhang et al in [6] address this problem in a stochastic setting (stochastic arrival processes and statistical guarantees) employing packet loss probability as the QoS metric. In [14] the above problem is addressed for leaky bucket regulated connections and statistical delay guarantees. The work most relevant to ours is that of [3], where tight delay bounds (also reported in [10]) have been presented in conjunction with a CAC algorithm for the single node case, which aims to calculate a weight assignment for the sessions that would not result in an over-achievement of session delays. In order to do so, the dependencies among the sessions must be considered. The CAC procedure in [3] does not address this problem directly since it employs an exhaustive search, having performance bound calculations as an intermediate step. More specifically, the maximum delay experienced by the sessions is determined for a weight assignment and the assignment is modified trying to maximize an objective function. While the search in [3] terminates after a finite number of steps, it does not guarantee that an acceptable assignment does not exist if it is not found. This could result in an over allocation of bandwidth and sometimes a call block, even if the bandwidth necessary to guarantee the call's QoS requirements is available.

A major contribution of this paper is a CAC algorithm for the single node case which fully exploits the bandwidth sharing mechanism of GPS and determines the *optimal weights ϕ directly from the QoS requirements of the sessions*, rather than through a recursive computation of the induced delay bounds and weight re-assignment. In addition, it turns out that the optimal scheme is less complex than that of [3].

The major results are derived by considering a mixed traffic environment in which the bandwidth resource controlled by the GPS server is assumed to be shared by a number of QoS sensitive streams and best effort traffic. This system will be referred to as a Best Effort Traffic Aware Generalized Processor Sharing (BETA-GPS) system. The developed algorithm determines the minimum ϕ assignments for the QoS sensitive streams which are just sufficient to meet their QoS and, consequently, maximizes the (remaining) ϕ assignment to the best effort traffic. Based on the main results an optimal CAC scheme is proposed in this paper for a decoupled system of GPS-controlled QoS sensitive traffic and best effort traffic (referred to as pure QoS system, see section IV). The formulation of the pure QoS system facilitates the derivation of the minimum required GPS scheduler capacity to support N QoS sensitive streams, which is, in itself, an interesting problem.

In section II some basic definitions are presented and the BETA-GPS environment is described. In section III an optimization process for the assignment is developed, the properties of the optimal assignment are studied and the proposed optimal CAC algorithm is derived. In section IV it is described how the results of section III can be utilized in the pure QoS GPS system. In section V some numerical results are presented. Section VI concludes the paper.

II. Definitions and description of the BETA-GPS system

A. GPS-related definitions

QoS sensitive sessions will be assumed to be leaky bucket constrained. That is, the amount of session i traffic arriving at the GPS server over any interval $(\tau, t]$ -referred to as the (assumed to be left continuous as in [1]) session i arrival function $A_i(\tau, t)$- will be bounded as follows: $A_i(\tau, t) \leq \sigma_i + \rho_i(t - \tau), \forall t \geq \tau \geq 0$; σ_i and ρ_i represent the burstiness and long term maximum mean arrival rate of session i. A session i is characterized as greedy starting at time τ, if the aforementioned bound is achieved, that is if $A_i(\tau, t) = \sigma_i + \rho_i(t - \tau), \forall t \geq \tau$.

A GPS system busy period is defined to be a maximal time interval during which at least one session is backlogged at any time instant in the interval.

An all-greedy GPS system is defined as a system in which all the sessions are greedy starting at time 0, the beginning of a system busy period. The significance of the all-greedy system follows from [1] (Theorem 3): If the input link speed of any session i exceeds the GPS service rate, then for every session i, the maximum delay D_i^* and the maximum backlog Q_i^* are achieved (not necessarily at the same time) when every session is greedy starting at time zero, the beginning of a system busy period.

This implies that if the server can guarantee an upper bound on a session' s delay under the all greedy system assumption this bound would be valid under any (leaky bucket constrained) arrival pattern. In view of the previous observation and by examining only all greedy systems, the CAC problem for a GPS system is simplified.

Let $t = 0$ denote the beginning of a system busy period in an all greedy system. For each session i the arrival function takes the form $A_i(0, t) = \sigma_i + \rho_i \cdot t, \forall t \geq 0$. If $Q_i(t)$ denotes the amount of session i traffic queued in the server at time t, then $Q_i(t) = A_i(0, t) - W_i(0, t)$ and $Q_i(t) = 0$ for all $t \leq 0$ by assumption. Let e_i denote the backlog clearing time of session i, then:

$$e_i = \sup\{t > 0 : Q_i(t) > 0\} \tag{2}$$

and $B_i = (0, e_i)$, corresponds to the session i busy period. The delay experienced by a session i "bit" arriving at time t is given by: $D_i(t) = \inf\{\tau \geq 0 : A_i(0, t) = W_i(0, t + \tau)\}$.

As indicated in the next section, the BETA-GPS system will "jointly" support QoS sensitive and best effort traffic. In addition to beeing leaky bucket constrained and greedy, the QoS sensitive sessions will be assumed 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).

To ensure that the delay constraint for the QoS sensitive session i is met, a minimum amount of service $N_i(0, t)$ must be provided by the GPS server to session i over the interval $(0, t]$, where

$$N_i(0, t) = \begin{cases} \sigma_i + \rho_i(t - D_i) & t \geq D_i \\ 0 & t < D_i \end{cases} \tag{3}$$

That is the actual amount of service (work) $W_i(0, t)$ provided by the GPS server to session i over the interval $(0, t]$ must satisfy:

$$W_i(0, t) \geq N_i(0, t), \forall t \geq 0 \tag{4}$$

The function $N_i(0, t)^1$ is referred to as session i requirements.

B. Description of the system and problem formulation

The Best Effort Traffic Aware (BETA) GPS system is depicted in Figure 1. The BETA-GPS server capacity C_G is assumed to be shared by N QoS sensitive sessions with descriptors $(\sigma_i, \rho_i, D_i), i = 1, \ldots, N$ 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. Quantities associated with a QoS sensitive session (best effort session) will be identified by a subscript i (be), $i = 1, \ldots, N$. To avoid degenerate cases and be consistent with the GPS definitions it is assumed that $\sigma_i \rho_i D_i \neq 0, i = 1, \ldots, N$ and that the ϕ assignment of the BETA-GPS scheduler to a session can not be zero $(\phi_i > 0, \quad i = 1, \ldots, N, be)$.

Generally, the task of CAC is to determine whether the network can accept a new session without causing QoS requirement violations. In the case of a GPS scheduler it

[1] This function is treated as right continuous to simplify the presentation. Since $A_i(0, t)$ is considered left continuous it is implied that QoS sensitive session i delay requirement is considered to be D_i^-.

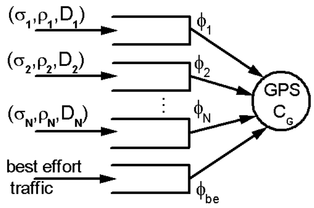


Fig. 1. Functional Diagram of the BETA-GPS system.

should also provide the server with the weight assignment which will be used in the actual service of the admitted calls. A CAC scheme for a GPS server is considered to be optimal if its incapability to admit a specific set of sessions implies that no ϕ assignment exists under which the server could serve this set of sessions (respecting all QoS requirements even under the worst case arrival scenario (all greedy system)). In addition, an optimal CAC scheme for the BETA-GPS system should seek to maximize the amount of service provided to the (traffic unlimited) best effort session under any arrival scenario and over any time horizon, while satisfying the QoS requirement of the (traffic limited) QoS sensitive sessions. That is, it should seek to maximize the normalized[2] weight assigned to the best effort traffic (ϕ_{be}), while satisfying the QoS requirement of QoS sensitive sessions.

Obviously, maximizing the weight assigned to the best effort traffic is equivalent to minimizing the sum of weights assigned to the QoS sensitive sessions. In view of this discussion, the following definition may be provided.

Definition 1:
(a) The optimal CAC scheme for the BETA-GPS system is the one that is based on the optimal ϕ assignment for the BETA-GPS system.
(b) The optimal ϕ assignment for the BETA-GPS system is the one that allows the QoS sensitive sessions to meet their QoS requirements - provided that it is possible - and achieves: $\max\{\phi_{be}\} = \max\{1 - \sum_{i=1}^{N} \phi_i\}$ or, equivalently, $\min\{\sum_{i=1}^{N} \phi_i\}$, where $\phi_i \in \mathbb{R}_+^*$, $i = 1, \ldots, N, be$, according to the definition of GPS.

In an all greedy system 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, that is, $\mathcal{B}(t) = \{i : e_i \geq t,\ i = 1, \ldots, N\}$, $\mathcal{E}(t) = \{i : e_i < t,\ \ i = 1, \ldots, N\}$, where e_i is defined in (2). Each session $k \in \mathcal{E}(t)$ requires a rate equal to ρ_k. Consequently, the bandwidth that can be considered to be available for allocation to the sessions i, $i \in \mathcal{B}(t)$ is equal to $(C_G - \sum_{k\in\mathcal{E}(t)} \rho_k)$. Session $i \in \mathcal{B}(t)$ will be allocated a share of that bandwidth equal to $\phi_i(1 - \sum_{k\in\mathcal{E}(t)} \phi_k)^{-1}$ and will be served with a rate $\phi_i(C_G - \sum_{k\in\mathcal{E}(t)} \rho_k)(1 - \sum_{k\in\mathcal{E}(t)} \phi_k)^{-1}$. Let

$$\hat{C}(t) \triangleq \frac{C_G - \sum_{j\in\mathcal{E}(t)} \rho_j}{1 - \sum_{j\in\mathcal{E}(t)} \phi_j} \tag{5}$$

be referred to as the Normalized Backlogged Sessions Allocated (NBSA) bandwidth (this quantity is called "universal slope" in [1]). Clearly $\hat{C}(t)$ changes value each time a session empties its backlog and remains constant between two consecutive backlog clearing times. Thus, $\hat{C}(t)$ is a piecewise constant function with the discontinuity points coinciding with the backlog clearing times of the sessions.

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.[3] 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

$$\hat{C}(t) = \frac{C_G - \sum_{k\in\mathcal{E}(\mathrm{b}_j)} \rho_k}{1 - \sum_{k\in\mathcal{E}(\mathrm{b}_j)} \phi_k} \quad \forall t \in (\mathrm{b}_{j-1}, \mathrm{b}_j] \tag{6}$$

$\hat{C}(t)$ is an increasing function of time since it preserves a constant value between two consecutive backlog clearing times and $\hat{C}(\mathrm{b}_j) < \hat{C}(\mathrm{b}_j^+)$ for a backlog clearing time b_j[4].

The amount of scheduler's work that is shared among the backlogged sessions $i \in \mathcal{B}(\mathrm{b}_j)$ over the time interval $(\mathrm{b}_{j-1}, \mathrm{b}_j]$ is equal to $(C_G - \sum_{i\in\mathcal{E}(\mathrm{b}_j)} \rho_i)(\mathrm{b}_j - \mathrm{b}_{j-1})$.

Let

$$\hat{W}(\mathrm{b}_{j-1}, \mathrm{b}_j) \triangleq \hat{C}(\mathrm{b}_{j-1}^+)(\mathrm{b}_j - \mathrm{b}_{j-1}) \tag{7}$$

be referred to as the Normalized Backlogged Sessions Allocated (NBSA) work. Then - in view of (6) and (7) -, 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 $(\mathrm{b}_{j-1}, \mathrm{b}_j]$.

In the next section a process is presented that converts any acceptable policy to the optimal policy, that is the policy which assigns the maximum possible weight to the best effort traffic. Although subsection III-A addresses a complete problem in itself, it is used more as an intermediate step of the proof that the algorithm presented in subsection III-C is optimal. In particular, the optimality of the presented algorithm follows from the properties of the optimization process presented in section III-A.

III. Optimal Call Admission Control for the BETA-GPS system

A. Optimizing an acceptable ϕ assignment

In this section a process that converts an acceptable ϕ assignment into a more efficient acceptable one is developed. **An acceptable ϕ assignment** is one which is feasible (that is $\sum_{i=1}^{N} \phi_i < 1$[5]) and delivers the required QoS to

[2]Without loss of generality, it is assumed that $\sum_{i=1}^{N} \phi_i + \phi_{be} = 1$

[3]Notice that e_i refers to the backlog clearing time of session i while b_j refers to the j^{th} of the ordered backlog clearing times.

[4]**Proof:** Assume, without loss of generality, that only one session, session k, empties its backlog at b_j. The fact that session k empties its backlog at b_j implies that it was served with a rate greater than ρ_k at $t = \mathrm{b}_j^-$, i.e.: $\rho_k < \phi_k \hat{C}(\mathrm{b}_j^-) \Leftrightarrow \rho_k \overline{\mathcal{P}_j} < \phi_k \overline{\mathcal{R}_j} \Leftrightarrow \overline{\mathcal{P}_j}\ \overline{\mathcal{R}_j} - \rho_k \overline{\mathcal{P}_j} > \overline{\mathcal{P}_j}\ \overline{\mathcal{R}_j} - \phi_k \overline{\mathcal{R}_j} \Leftrightarrow \overline{\mathcal{P}_j}(\overline{\mathcal{R}_j} - \rho_k) > (\overline{\mathcal{P}_j} - \phi_k)\overline{\mathcal{R}_j} \Leftrightarrow \overline{\mathcal{P}_j}\,\overline{\mathcal{R}_j^+} > \overline{\mathcal{P}_j^+}\,\overline{\mathcal{R}_j} \Leftrightarrow \hat{C}(\mathrm{b}_j) = \hat{C}(\mathrm{b}_j^-) < \hat{C}(\mathrm{b}_j^+)$
where $\overline{\mathcal{R}_j} = C_G - \sum_{i\in\mathcal{E}(\mathrm{b}_j)} \rho_i$, $\overline{\mathcal{P}_j} = 1 - \sum_{i\in\mathcal{E}(\mathrm{b}_j)} \phi_i$,
$\overline{\mathcal{R}_j^+} = C_G - \sum_{i\in\mathcal{E}(\mathrm{b}_j^+)} \rho_i$, $\overline{\mathcal{P}_j^+} = 1 - \sum_{i\in\mathcal{E}(\mathrm{b}_j^+)} \phi_i$

[5]Strict inequality is assumed to avoid the degenerate case which under equality would not leave any remaining ϕ to be assigned to the best effort traffic.

each of the supported QoS sensitive sessions. **A ϕ assignment is more efficient than another** if the sum of ϕ's $\sum_{i=1}^{N}\phi_i$ under the former assignment is smaller than that under the latter.

The aforementioned process will be referred to as the XMF (eXpand Minimum busy period First) process. According to the XMF process each QoS sensitive session's busy period is expanded as much as its QoS would permit, starting from the set of QoS sensitive sessions that empty their backlog first in order. A very important property of the XMF process is that it converts any acceptable ϕ assignment into the optimal one. That is, the process generates the ϕ assignment that maximizes the weight assigned to the best effort traffic (or, minimizes $\sum_{i=1}^{N}\phi_i$, for the N QoS sensitive sessions).

Let Π denote the set of acceptable policies (or equivalently, ϕ assignments) and let $\pi_a \in \Pi$. The application of the XMF process to π_a results in an acceptable policy $\pi_o = XMF(\pi_a)$, which is not less efficient than π_a; XMF(π) denotes a policy that is generated by applying the XMF process to π. In particular, it will be shown that π_o is unique and more efficient than π_a, except for the case in which $\pi_a = \pi_o$.

Let ${}^{a}\mathrm{I}_k$ denote the set of QoS sensitive sessions that empty their backlog k-th in order under π_a and let ${}^{a}\mathrm{b}_k$ be the time instant when this happens. Let ${}^{a}\mathrm{I}_k^P = \bigcup_{s=1}^{k-1} {}^{a}\mathrm{I}_s$ and ${}^{a}\mathrm{I}_k^F = \bigcup_{s\geq k+1} {}^{a}\mathrm{I}_s$ denote the sets of sessions that empty their backlog before (past) and after (future) ${}^{a}\mathrm{b}_k$, respectively. The following definitions will be needed:

Definition 2:

(a) A session i is compressed (decompressed) in ϕ- space if its weight is decreased (increased).

(b) A session i is decompressed in t- space, or its busy period is expanded, if its backlog clearing time is increased.

(c) Sessions in ${}^{a}\mathrm{I}_k$ are uniformly decompressed in t- space, or their busy periods are uniformly expanded, if their backlog clearing times are equally increased.

(d) A session i preserves its position in ϕ-(t-) space if its weight (backlog clearing time) remains unchanged.

(e) A set $A \subseteq {}^{a}\mathrm{I}_k$ is compressible in ϕ- space if $\forall i \in A$ $d\phi_i > 0$ exists, such that sessions $i \in A$ do not violate their delay bounds when they are assigned a weight $\phi_i - d\phi_i$, under the conditions: a) sessions in ${}^{a}\mathrm{I}_k^P \cup \{{}^{a}\mathrm{I}_k \setminus A\}$ preserve their position in ϕ- space; b) sessions which emptied their backlog after sessions in A still empty their backlog after sessions in A.

A.1 Description of the XMF process.

The XMF process applied to an acceptable policy $\pi_a \in \Pi$ is described next (its flowchart is depicted in Figure 2). Throughout the description no reference is made to the best effort traffic. Only the treatment of the QoS sensitive sessions is considered, and this is sufficient since the weight assigned to best effort traffic is given by $1-\sum_{i=1}^{N}\phi_i^{\pi}$ under a policy π.

At this point the following should be noted. XMF is a conceptual process which is not directly applicable at a computational level. The weights assigned to sessions and their busy periods change in a continuous way under this process. In addition, and in order to keep the presentation clear and simple, each time that the process modifies the policy it is applied to, the process is presented to be reapplied in its entirety to the modified policy, although not necessary. The rationale for this approach is that since XMF is a conceptual process it is not a concern how many times it will be applied, as long as it terminates (generates results) after a finite number of steps.

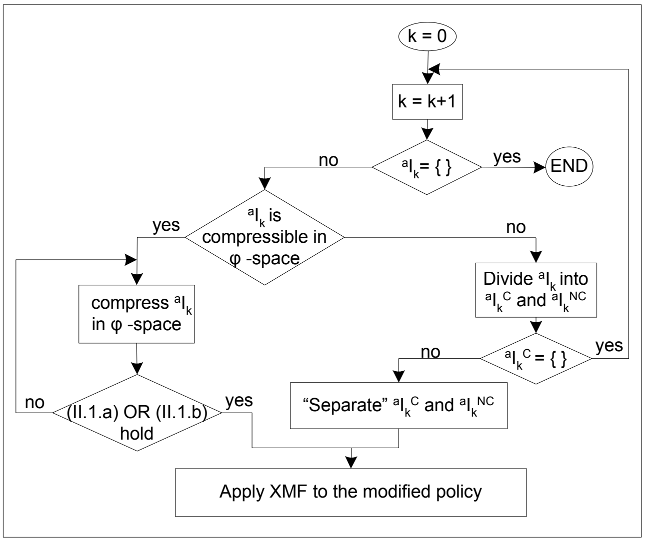


Fig. 2. Flowchart of the XMF process

(0) Initially, $k = 0$.

(I) k=k+1.[6] If ${}^{a}\mathrm{I}_k = \emptyset$ goto (III).

(II) Sessions in ${}^{a}\mathrm{I}_k$ are considered. Sessions in ${}^{a}\mathrm{I}_k^P$ preserve their position in ϕ- space. This implies that the sessions in ${}^{a}\mathrm{I}_k^P$ preserve their position in t- space as well[7].

(II.1) If ${}^{a}\mathrm{I}_k$ is compressible the sessions in ${}^{a}\mathrm{I}_k$ are compressed in ϕ- space in such a way that their busy periods are uniformly expanded. At the same time the sessions in ${}^{a}\mathrm{I}_k^F$ are decompressed in ϕ- space in such a way (under the condition) that they receive the same amount of work up to the end of the (modified) busy periods of sessions in ${}^{a}\mathrm{I}_k$ as they did under π_a[8]. This "conditional exchange of weights" is possible[7] and does not alter the backlog clearing times of the sessions in ${}^{a}\mathrm{I}_k^F$, that is the sessions in ${}^{a}\mathrm{I}_k^F$ preserve their position in t- space[7]. The "conditional exchange of weights" between sessions in ${}^{a}\mathrm{I}_k$ and sessions in ${}^{a}\mathrm{I}_k^F$ takes place continuously until one of the following happen:

(II.1.a) The backlog clearing time of sessions in ${}^{a}\mathrm{I}_k$ becomes equal to the backlog clearing time of sessions which empty their backlog k+1-th in order under π_a (sessions in

[6] Since policy $\pi_a \in \Pi$ is known, the sets of sessions, ${}^{a}\mathrm{I}_k$, that empty their backlog k^{th} in order under policy π_a, $k \geq 1$ (in an all greedy system) are well defined and can be considered to be known.

[7] see proof of Proposition 1 in the Appendix.

[8] This increase of their weights is necessary for sessions in ${}^{a}\mathrm{I}_k^F$ in order to receive the same amount of work up to the end of the (modified) busy periods of sessions in ${}^{a}\mathrm{I}_k$, since the expansion of the busy periods of sessions in ${}^{a}\mathrm{I}_k$ leads to a reduction of the NBSA bandwidth in the interval between the original and the modified backlog clearing time of sessions in ${}^{a}\mathrm{I}_k$.

${}^a\mathrm{I}_{k+1}$), or

(II.1.b) sessions in ${}^a\mathrm{I}_k$ can not be compressed any further in ϕ-space (their busy periods be uniformly expanded), because some session will miss its delay bound.

At the end of step (II.1) new ϕ's will have been assigned to (all) streams in ${}^a\mathrm{I}_k$ and ${}^a\mathrm{I}_k^F$ while streams in ${}^a\mathrm{I}_k^P$ will have maintained their original ϕ's under π_a. Thus, a new policy π_b is defined in terms of the new ϕ's which is shown[7] to be acceptable and more efficient than π_a[9]. At the end of step (II.1), the XMF process is applied to policy π_b (modified π_a) from the beginning (from step (0)).

(II.2) Else i f ${}^a\mathrm{I}_k$ is not compressible then a uniform expansion of the busy period of sessions in ${}^a\mathrm{I}_k$ is not feasible. In this case, ${}^a\mathrm{I}_k$ is divided into two subsets, that is ${}^a\mathrm{I}_k = {}^a\mathrm{I}_k^C \cup {}^a\mathrm{I}_k^{NC}$, where: ${}^a\mathrm{I}_k^C$ is the maximum subset of ${}^a\mathrm{I}_k$ which is compressible in ϕ- space and ${}^a\mathrm{I}_k^{NC} = {}^a\mathrm{I}_k \setminus {}^a\mathrm{I}_k^C$.

(II.2.a) If ${}^a\mathrm{I}_k^C \neq \emptyset$ then the two sets ${}^a\mathrm{I}_k^C$ and ${}^a\mathrm{I}_k^{NC}$ are separated by performing an infinitesimal uniform expansion of the busy periods of sessions in ${}^a\mathrm{I}_k^C$ (at the same time the weights of sessions in ${}^a\mathrm{I}_k^F$ are increased in such a way that they receive the same amount of work up to the end of the (modified) busy periods of sessions in ${}^a\mathrm{I}_k^C$ as they did under π_a (as at step (II.1))), resulting in a new policy π_b.[10] After step (II.2.a), the XMF process is applied to policy π_b (modified π_a) from the beginning (from step (0)).

(II.2.b) If ${}^a\mathrm{I}_k^C = \emptyset$ then no stream in ${}^a\mathrm{I}_k$ may be compressed in ϕ- space any further and the next set ${}^a\mathrm{I}_{k+1}$ needs to be considered. Thus the XMF process continues from step (I).

(III) End of the XMF process. At this step the unique optimal policy πo (see Proposition 3) has been determined, that is the original acceptable policy π_a has been optimized. Under the resulting policy π_o the QoS sensitive sessions are assigned some weights $\phi_i^{\pi_o}, i = 1, \ldots, N$ and the best effort traffic is assigned weight $\phi_{be}^{\pi_o} = 1 - \sum_{i=1}^N \phi_i^{\pi_o}$.

A.2 Properties of the XMF process

The XMF process forces all sessions to empty their backlog as late as possible. The only parameters that impose an upper limit on the expansion of the busy periods are the sessions' delay bounds, which do not depend on π_a. Thus, one could expect π_o not to depend on π_a. The following propositions hold. Their proofs may be found in the Appendix.

Proposition 1: The (intermediate) policy π_b that is defined at the end of steps (II.1.a), (II.1.b) or (II.2.a) is acceptable and more efficient than π_a.

[9]Under π_b ${}^b\mathrm{I}_k = {}^a\mathrm{I}_k \cup {}^a\mathrm{I}_{k+1}$ and ${}^b\mathrm{b}_k > {}^a\mathrm{b}_k$ if step (II.1.a) is followed, or ${}^b\mathrm{I}_k = {}^a\mathrm{I}_k$ but ${}^b\mathrm{b}_k > {}^a\mathrm{b}_k$ if step (II.1.b) is followed.

[10]Under π_b ${}^b\mathrm{I}_k = {}^a\mathrm{I}_k^{NC}$, ${}^b\mathrm{b}_k = {}^a\mathrm{b}_k$ and ${}^b\mathrm{I}_{k+1} = {}^a\mathrm{I}_k^C$, ${}^a\mathrm{b}_{k+1} > {}^b\mathrm{b}_{k+1} = {}^a\mathrm{b}_k + dt > {}^a\mathrm{b}_k$, $dt > 0$ arbitrary but such that (a) the inequality holds and (b) ${}^b\mathrm{I}_{k+1} = {}^a\mathrm{I}_k^C$ is still compressible. This formulation is used to express the fact that sessions in ${}^a\mathrm{I}_k^C$ do not violate their requirements during the infinitesimal uniform expansion of their busy periods and that the infinitesimal "conditional exchange of weights" between sessions in ${}^a\mathrm{I}_k^C$ and ${}^a\mathrm{I}_k^F$ is used just to separate sets ${}^a\mathrm{I}_k^C$ and ${}^a\mathrm{I}_k^{NC}$.

Proposition 2: The final policy that results when the application of the XMF process to an arbitrary original acceptable policy is terminated, is acceptable and does not depend on the original policy. That is, $\forall \pi_{a1}, \pi_{a2} \in \Pi, \quad XMF(\pi_{a1}) = XMF(\pi_{a2}) = \pi_o, \pi_o \in \Pi$.

From Propositions 1 and 2 it is easily concluded that a) π_o is the only policy that remains unchanged under the XMF process; that is, $\forall \pi_a \in \Pi, XMF(\pi_a) = \pi_a \Leftrightarrow \pi_a = \pi_o$ and b) for any $\pi_a \in \Pi$, XMF(π_a)$= \pi_o$ is more efficient than π_a, except the case where $\pi_a = \pi_o$. In view of the above the following proposition is self-evident.

Proposition 3: Policy π_o, $\pi_o = XMF(\pi)$ for any $\pi \in \Pi$ is optimal and unique.

B. Properties of the optimal ϕ assignment

In this section some properties of the optimal policy π_o are provided. These properties help establish in the next section that the proposed CAC algorithm is optimal. It is shown that in order to determine the optimal policy it is sufficient to observe the all greedy system at certain time instances, which coincide with either the delay bound or the backlog clearing time of some session. For this reason the notion of the checkpoints is introduced.

Definition 3: Let $\tau_0 = 0$, that is τ_0 coincides with the beginning of the system busy period of an all greedy system. 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, $m = 0, \ldots, M$ will be referred to as the m^{th} ordered checkpoint[11].

Definition 4: Let $d_i = \{m : \tau_m = D_i\}$. That is, checkpoint τ_{d_i} coincides with the time instant at which the deadline of session i expires. Then the following quantities are defined for session $i, i = 1, \ldots, N$ at all checkpoints τ_m such that $D_i \leq \tau_m < e_i$ (that is, checkpoints at which session i is still backlogged and its requirements (as defined in equation (3)) are greater than zero.

$$\phi_i^-(\tau_m) = \frac{N_i(0, \tau_m)}{\hat{W}(0, \tau_m)} \tag{8}$$

$$\phi_i^+(\tau_m) = \frac{\rho_i}{\hat{C}(\tau_m^+)} \tag{9}$$

where $\hat{W}(0, \tau_m) = \sum_{k=1}^m \hat{W}(\tau_{k-1}, \tau_k)$.

The quantity $\phi_i^-(\tau_m)$ is expressed in terms of the values of the associated quantities (session 's requirements, NBSA work) left of τ_m and refers to the evolution of the system for t less than τ_m. On the other hand, the quantity $\phi_i^+(\tau_m)$ is expressed in terms of the value of the NBSA bandwidth right of τ_m.

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

[11]Equation (6) which holds for time instances $\in \{\mathrm{b}_m\}_{m=1}^L$ does hold for all time instances in $\{\tau_m\}_{m=1}^M \supseteq \{\mathrm{b}_m\}_{m=1}^L$ as well, with the observation that the time instances in $\{\tau_m\}_{m=1}^M \setminus \{\mathrm{b}_m\}_{m=1}^L$ are degenerate points of discontinuity of the NBSA bandwidth.

backlog before time τ_m[12]. In particular, the denominator of the right hand side of equation (8) is the total amount of NBSA work which is assigned to backlogged sessions up to time τ_m, including sessions which cleared their backlog earlier and are no longer backlogged at time τ_m. Each session i, which is still backlogged at τ_m gets a fraction of this work equal to ϕ_i, that is, it is assigned work equal to $\phi_i \cdot \sum_{k=1}^{m} \hat{W}(\tau_{k-1}, \tau_k)$.

The quantity $\phi_i^+(\tau_m)$ represents the fraction of the NBSA bandwidth just after τ_m that must be assigned to session i in order for session i to be served with rate equal to ρ_i. It is easily seen that 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.

The usefulness of these quantities follows from Proposition 5.

Proposition 4: Under π_o, $e_i > D_i$, $\forall i \in QoS$ ($QoS \triangleq \{1, 2, \ldots, N\}$). That is, each QoS sensitive session empties its backlog after checkpoint $\tau_{d_i} = D_i$.

Proof: For $t < D_i$ session i requirements have zero value and, thus, they do not impose a restriction on the expansion of session i busy period. This means that if $e_i < D_i$ session i would be compressible in ϕ- space and, thus, π_o would not remain unchanged under XMF. If $e_i = D_i$, since for $D_i \rho_i \neq 0$ $A_i(0, D_i) - N_i(0, D_i) > 0$, it is implied that an infinitesimal decrease of session 's weight would be possible (session i would be compressible in ϕ- space). ■

Proposition 5: Under π_o, QoS sensitive session i is assigned weight:

$$\phi_i^{\pi_o} = \begin{cases} \phi_i^-(\tau_k) \text{ if } \exists k \text{ such that} \\ k = \min\{m : \phi_i^-(\tau_m) \geq \phi_i^+(\tau_m)\} \\ \phi_i^+(\tau_k), k = \max\{n : \tau_n < \infty\}, \text{ otherwise} \end{cases} \tag{10}$$

Proof: As proved in section III-A.2 the optimal policy for the BETA-GPS system (π_o) is the unique acceptable policy that remains "unchanged" under the XMF process. This implies that the weight $\phi_i^{\pi_o}$ assigned to QoS sensitive session i under π_o is such that the following two conditions are fulfilled.

(C1) QoS requirements of session i are met. In particular, for an arbitrary time instant t:

(C1a) QoS requirements of session i up to time t are met.

(C1b) QoS requirements of session i $\forall t' > t$ are met.

(C2) session i is not compressible in ϕ-space.

Let τ_f denote the last checkpoint with finite value, that is $\tau_f = \max\{\tau_m : \tau_m < \infty\}$. In order for condition (C1a) to hold for $t = \tau_f$ it is sufficient:

$$\phi_i^{\pi_o} \geq \max_{\tau_m \leq \tau_f} \{\phi_i^-(\tau_m)\} \tag{11}$$

In particular (11) implies that for two consecutive checkpoints τ_{j-1} and τ_j with finite value: (a) $W_i(0, \tau_{j-1}) \geq N_i(0, \tau_{j-1})$ and (b) $W_i(0, \tau_j) \geq N_i(0, \tau_j)$. Since $N_i(0, t)$ and $W_i(0, t)$ are both linear in $[\tau_{j-1}, \tau_j]$, (a) and (b) imply that $W_i(0, t) \geq N_i(0, t)$, $\forall t \in [\tau_{j-1}, \tau_j]$.

[12]It should be emphasized that the previous does not imply that session 's requirements are satisfied for all $t < \tau_m$ (see Prop. 5).

Now it is easily seen that:

$$\phi_i^{\pi_o} = \max\{\max_{\tau_m \leq \tau_f} \{\phi_i^-(\tau_m)\}, \phi_i^+(\tau_f)\} \tag{12}$$

where the "external" max takes into account condition (C1b) (session 's requirements must be met not only up to τ_f but also for $t \geq \tau_f$) and equality holds due to condition (C2)(that session i is not compressible in ϕ-space under π_o).

In order to proceed the following must be proved:

Claim 1: For two consecutive checkpoints τ_{j-1} and τ_j with finite value ($\tau_j \leq \tau_f$) $\phi_i^+(\tau_j) \leq \phi_i^+(\tau_{j-1})$.[13]

Claim 2: For two consecutive checkpoints τ_{j-1} and τ_j with finite value ($\tau_j \leq \tau_f$), $\phi_i^-(\tau_{j-1}) < \phi_i^+(\tau_{j-1}) \Rightarrow \phi_i^-(\tau_{j-1}) < \phi_i^-(\tau_j)$.[14]

Claim 1 implies that:

$$\phi_i^+(\tau_f) = \min_{\tau_m \leq \tau_f} \{\phi_i^+(\tau_m)\} \tag{13}$$

- If $\exists k$ such that $k = \min\{m : \phi_i^-(\tau_m) \geq \phi_i^+(\tau_m)\}$ Claim 2 holds for every $\tau_{j-1}, \tau_j, \tau_j \leq \tau_k$ and implies that $\phi_i^-(\tau_k) = \max_{\tau_m \leq \tau_k} \{\phi_i^-(\tau_m)\}$. Obviously $\phi_i^-(\tau_k) \geq \phi_i^+(\tau_k)$ implies that $\phi_i^-(\tau_k) \geq \phi_i^-(\tau_j), k \leq j \leq f$, since $\phi_i^-(\tau_k)$ is sufficient to ensure that session's QoS requirements are met for $t > \tau_k$. So $\phi_i^-(\tau_k) = \max_{\tau_m \leq \tau_f} \{\phi_i^-(\tau_m)\}$ and (12), in conjunction with (13)), implies that $\phi_i^{\pi_o} = \phi_i^-(\tau_k)$. This means that Proposition 5 holds.
- If $\nexists k$ such that $k = \min\{m : \phi_i^-(\tau_m) \geq \phi_i^+(\tau_m)\}$ Claim 2 holds for every τ_{j-1}, τ_j with $\tau_j \leq \tau_f$ and implies that $\phi_i^-(\tau_f) = \max_{\tau_m \leq \tau_f} \{\phi_i^-(\tau_m)\}$. Since $\phi_i^-(\tau_f) < \phi_i^+(\tau_f)$ (12) implies that $\phi_i^{\pi_o} = \phi_i^+(\tau_f)$ and Proposition 5 holds.

■

Proposition 6: Assume that $\tau_k, \hat{C}(\tau_k^+)$, $\forall k \leq j-1$, are known. τ_j is given by [15]:

$$\tau_j = \min(\min_{i \in Nex_{j-1}} D_i \,, \min_{i \in Phi_{j-1}} vct_i(\tau_{j-1})) \tag{14}$$

where Nex_{j-1} is the set of sessions with delay bound greater than τ_{j-1}, Phi_{j-1} is the set of sessions for which $\exists k \leq j-1 : \phi_i^-(\tau_k) \geq \phi_i^+(\tau_k)$ and have not cleared their backlog up to time τ_{j-1} and

$$vct_i(\tau_{j-1}) = \tau_{j-1} + \frac{A_i(0, \tau_{j-1}) - W_i(0, \tau_{j-1})}{\phi_i^{\pi_o} \cdot \hat{C}(\tau_{j-1}^+) - \rho_i} \tag{15}$$

[13]**Proof:** $\hat{C}(\tau_{j-1}) \leq \hat{C}(\tau_j) \Leftrightarrow \phi_i^+(\tau_{j-1}) = \frac{\rho_i}{\hat{C}(\tau_{j-1})} \geq \frac{\rho_i}{\hat{C}(\tau_j)} = \phi_i^+(\tau_j)$

[14]**Proof:** $\phi_i^-(\tau_{j-1}) < \phi_i^+(\tau_{j-1}) \Leftrightarrow \frac{N_i^{j-1}}{\mathcal{W}_1^{j-1}} < \frac{\rho_i}{\mathcal{C}_{j-1}^+} \Leftrightarrow$ $N_i^{j-1} \mathcal{C}_{j-1}^+ \tau_{j-1}^j + N_i^{j-1} \mathcal{W}_1^{j-1} < \tau_{j-1}^j \rho_i \mathcal{W}_1^{j-1} + N_i^{j-1} \mathcal{W}_1^{j-1} \Leftrightarrow$ $N_i^{j-1} \left(\mathcal{C}_{j-1}^+ \tau_{j-1}^j + \mathcal{W}_1^{j-1} \right) < \left(N_i^{j-1} + \tau_{j-1}^j \rho_i \right) \mathcal{W}_1^{j-1} \Leftrightarrow$ $N_i^{j-1} \mathcal{W}_1^j < \left(N_i^{j-1} + \tau_{j-1}^j \rho_i \right) \mathcal{W}_1^{j-1} \Leftrightarrow N_i^{j-1} \mathcal{W}_1^j < N_i^j \mathcal{W}_1^{j-1} \Leftrightarrow$ $\phi_i^-(\tau_{j-1}) < \phi_i^-(\tau_j)$ where $N_i^x = N_i(0, \tau_x)$, $\tau_{x-1}^x = \tau_x - \tau_{x-1}$, $\mathcal{C}_x^+ = \hat{C}(\tau_x^+)$, $\mathcal{W}_1^x = \sum_{k=1}^{x} \hat{W}(\tau_{k-1}, \tau_k)$.

[15]It is noted that in order to avoid unnecessary complexity it is assumed that $\min(\emptyset) = \infty$, that is the min function returns ∞ when applied to an empty set.

is the virtual clearing time of session i at τ_{j-1}, that is the backlog clearing time of session i assuming that no other session is going to empty its backlog before session i does.

Proof: Let the QoS sensitive sessions be divided in the following disjoint sets at τ_{j-1} :

- $Empty_{j-1}$: contains sessions which have emptied their backlog at the current or a previous checkpoint, i.e. $Empty_{j-1} = \{i \in QoS : e_i \leq \tau_{j-1}\}$.
- Nex_{j-1}: contains sessions with a delay bound greater than τ_{j-1}, i.e. $Nex_{j-1} = \{i \in QoS : D_i > \tau_{j-1}\}$.
- Phi_{j-1}: contains sessions for which $\exists k \leq j-1 : \phi_i^-(\tau_k) \geq \phi_i^+(\tau_k)$ and have not cleared their backlog up to time τ_j.
- $Trans_{j-1}$: contains sessions for which $\nexists k \leq j-1 : \phi_i^-(\tau_k) \geq \phi_i^+(\tau_k)$ and have $D_i \leq \tau_{j-1}$.

According to the definition of checkpoints τ_j coincides with either the delay bound or the backlog clearing time of some session.

Obviously no future checkpoint is associated with sessions $i \in Empty_{j-1}$, since $e_i \leq \tau_{j-1}$ and according to Proposition 4, $D_i < e_i \; \forall i \in Empty_{j-1}$.

According to Proposition 4 all sessions $i \in Nex_{j-1}$ have $e_i > \min_{i \in Nex_{j-1}} D_i$. It is implied that if τ_j is associated with a session in Nex_{j-1} it will have to be the session k, $k : D_k = \min_{i \in Nex_{j-1}} D_i$. This "justifies" the first term of equation.

Claim 3: For two consecutive checkpoints τ_{j-1} and τ_j with finite value ($\tau_j \leq \tau_f$), $\phi_i^-(\tau_{j-1}) < \phi_i^+(\tau_{j-1}) \Rightarrow \phi_i^+(\tau_{j-1}) > \phi_i^-(\tau_j)$.[16]

From Proposition 5, in conjunction with Claim 3, it is easily concluded that sessions in $Trans_{j-1}$ are assigned weight less than or equal to $\phi_i^+(\tau_{j-1})$ under π_o , i.e. they are served with a rate less than ρ_i in the interval $(\tau_{j-1}, \tau_j]$. This means that the slope of their service curve is less than the slope of their (greedy) arrival curve. Both curves are linear in $(\tau_{j-1}, \tau_j]$ and it is obvious that these two lines can not cross each other, i.e. sessions in $Trans_{j-1}$ do not empty their backlog at τ_j.

All sessions in Phi_{j-1} are served with a rate greater than or equal to ρ_i for $t > \tau_{j-1}$ so they could empty their backlog at τ_j. For every session $i \in Phi_{j-1}$ the line with slope $\phi_i^{\pi_o}\hat{C}^+(\tau_{j-1})$ which passes from point $(\tau_{j-1}, W_i(0, \tau_{j-1})$ represents $W_i(0,t)$ for $t \geq \tau_{j-1}$ and has the form $y_1(t) = \phi_i^{\pi_o}\hat{C}^+(\tau_{j-1})(t-\tau_{j-1}) + W_i(0,\tau_{j-1})$. The line $y_2(t) = \sigma_i + \rho_i t$ represents the arrivals of the greedy session. These two lines cross each other at $t = \frac{\sigma_i + \phi_i^{\pi_o}\hat{C}^+(\tau_{j-1})\,\tau_{j-1} - W_i(0,\tau_{j-1})}{\phi_i^{\pi_o}\hat{C}^+(\tau_{j-1}) - \rho_i}$ $= \tau_{j-1} + \frac{A_i(0,\tau_{j-1}) - W_i(0,\tau_{j-1})}{\phi_i^{\pi_o}\cdot\hat{C}^+(\tau_{j-1}) - \rho_i} = vct_i(\tau_{j-1})$. It is obvious that the only sessions for which their virtual clearing time represents their real backlog clearing time are the sessions whose virtual clearing time is equal to the minimum virtual clearing time. ∎

[16]**Proof:** $\phi_i^-(\tau_{j-1}) < \phi_i^+(\tau_{j-1}) \Leftrightarrow \frac{N_i^{j-1}}{\mathcal{W}_1^{j-1}} < \frac{\rho_i}{\mathcal{C}_{j-1}^+} \Leftrightarrow N_i^{j-1}\mathcal{C}_{j-1}^+ + \rho_i\tau_{j-1}^j\mathcal{C}_{j-1}^+ < \rho_i\mathcal{W}_1^{j-1} + \rho_i\tau_{j-1}^j\mathcal{C}_{j-1}^+ \Leftrightarrow \mathcal{C}_{j-1}^+\left(N_i^{j-1} + \rho_i\tau_{j-1}^j\right) < \rho_i\left(\mathcal{W}_1^{j-1} + \tau_{j-1}^j\mathcal{C}_{j-1}^+\right) \Leftrightarrow \mathcal{C}_{j-1}^+ N_i^j < \rho_i\mathcal{W}_1^j \Leftrightarrow \phi_i^+(\tau_{j-1}) > \phi_i^-(\tau_j)$
The notations are the same as in proof of Claim 2.

C. Optimal Call Admission Control Algorithm

The CAC algorithm presented in this section determines progressively the optimal policy π_o, based on Propositions 5 and 6. It includes two conceptually distinct functions. One which (based on Proposition 5) examines whether the optimal weights of the QoS sensitive sessions can be determined at a specific checkpoint and another which (based on Proposition 6) determines the next checkpoint.

Assume that all the checkpoints (τ_k) and the corresponding values of the NBSA bandwidth ($\hat{\mathrm{C}}(\tau_k^+)$) were known. Proposition 5 could be used in order to determine the optimal weights of the QoS sensitive sessions. A sequential consideration of all the checkpoints would be required since, according to (the first case of) Proposition 5, for each QoS sensitive session i the first checkpoint τ_j at which the condition $\phi_i^-(\tau_j) \geq \phi_i^+(\tau_j)$ would hold should be determined in order to determine the optimal weight of session i; if τ_m denotes this particular checkpoint for session i the optimal weight of session i is $\phi_i^{\pi_o} = \phi_i^-(\tau_m)$. The weights of the QoS sensitive sessions not fulfilling the aforementioned condition at any checkpoint (with a finite value) would be determined at the last checkpoint with a finite value, as prescribed by the second case of Proposition 5; if τ_m denotes the maximum checkpoint, the optimal weights of such sessions i are $\phi_i^{\pi_o} = \phi_i^+(\tau_m)$.

In the sequel, it is said that *session i is examined at τ_j*, when it is checked whether the condition $\phi_i^-(\tau_j) \geq \phi_i^+(\tau_j)$ (in the first case of Proposition 5), referred to as the *condition*, holds for session i at τ_j. The interval $[D_i, e_i)$ is referred to as *the examination interval of session i*; notice that the quantities $\phi_i^-(\tau_j)$, $\phi_i^+(\tau_j)$ used in the *condition* are defined for session i at τ_j only if $D_i \leq \tau_j < e_i$ (see Definition 4). In addition, it is said that *session i is assigned an appropriate weight*, when session i meets the *condition* at τ_j and is assigned the optimal weight $\phi_i^{\pi_o} = \phi_i^-(\tau_j)$.

The algorithm presented in this section employs Proposition 5 in a similar way as that described above in order to determine the optimal weights of the QoS sensitive sessions. That is, the checkpoints are considered sequentially and *each session i is examined at the checkpoints* that are within *the examination interval of session i*. At the checkpoint that the *condition* is met for session i for the first time, *session i is assigned an appropriate weight.* At the last checkpoint with a finite value, sessions that have not been assigned a weight are assigned a weight equal to that prescribed by the second case of Proposition 5. However, in the framework of the algorithm the checkpoints (τ_k) and the corresponding values of the NBSA bandwidth ($\hat{\mathrm{C}}(\tau_k^+)$) are determined on the fly (by using Proposition 6), since they are not a priori known; only the first checkpoint ($\tau_0 = 0$) and the initial value of the NBSA bandwidth ($\hat{\mathrm{C}}(\tau_0^+) = C_G$) are known.

The pseudocode of the algorithm is provided in Figure 3, while Figure 4 illustrates the flowchart of the algorithm. C_G is the bandwidth controlled by the GPS scheduler and QoS is the set of QoS sensitive sessions under investigation. The algorithm iterates over the checkpoints starting at $\tau_0 = 0$; index j runs over the checkpoints, while index

i runs over the QoS sensitive sessions. At each iteration a checkpoint (τ_j) is considered, and each QoS sensitive session i belongs to one of the following disjoint sets Phi_j, $Empty_j$, Nex_j, $Trans_j$. These are the same sets of sessions as those defined in the proof of Proposition 6. However, in the framework of the algorithm their conceptual interpretation is slightly different. More specifically:

- Set Phi_j contains sessions that fulfil the *condition* at τ_j or some previous checkpoint and have not emptied their backlog yet (up to τ_j). In the framework of the algorithm these are the sessions whose weights have been determined (since, according to the algorithm, when the *condition* is met for a session at a checkpoint this session *is assigned an appropriate weight*).
- Set $Empty_j$ contains sessions which have emptied their backlog at (or before) τ_j. Their weights are known, since (as explained in the sequel, at *Step B.3* of the description) sessions that empty their backlog at τ_j belonged in Phi_k for some $0 < k \leq j-1$, that is their weights have been determined at some previous checkpoint.
- Set Nex_j contains sessions whose delay bound is less than τ_j. In the framework of the algorithm this set is also referred to as the set of not examined sessions, since these sessions need not to *be examined at* τ_j; τ_j is not within the *examination interval of any session* in Nex_j and, thus, the *condition* is definitely not met by any of them at τ_j.
- Set $Trans_j$ contains the rest of the sessions, that is sessions that have been examined at a previous checkpoint, but their weights have not been determined.

Next a detailed description of the algorithm is provided. In the description it is assumed that all the checkpoints up to τ_{j-1} are known, as well as the sets of sessions Phi_k, $Empty_k$, Nex_k, $Trans_k$, for $k \leq j-1$ and the weights of the sessions in Phi_{j-1} and in $Empty_{j-1}$ (these weights are determined during previous iterations). Notice that this holds for $j = 1$, since $\tau_0 = 0$ and $C(\tau_0^+) = C_G$; at τ_0 all QoS sensitive sessions belong to Nex_0 (since their delay bounds are greater than $\tau_0 = 0$), while all other sets are empty.

Step A: Initialization of the algorithm. Initially, the checkpoint which corresponds to the beginning of the system busy period of the all greedy system (that is, $\tau_0 = 0$) is considered. At τ_0 steps B.2-B.6 of the algorithm are not executed, since the sets of sessions are already known (as explained, at the beginning all sets are empty except Nex_0).

Step B Main loop (B1.-B6.) is executed until the weights of all QoS sensitive sessions are determined.

Step B1 Using Proposition 6 the value of time at the current checkpoint τ_j is computed. The weights of sessions in Phi_{j-1}, which are needed in order to determine τ_j, have been determined using Proposition 5 at some previous checkpoint, since Phi_{j-1} contains sessions which fulfill the *condition* at some previous checkpoint. Thus, the quantities needed to determine the next checkpoint are known or can be computed[17].

Step B2 If $\tau_j = \infty$, the previous checkpoint (τ_{j-1}) was the one with the maximum finite value of time and according to the second case of Proposition 5 the weights of the sessions whose weight has not been determined yet must be determined at τ_{j-1}. Thus, sessions i in $Trans_{j-1}$ are assigned a weight equal to $\phi_i^+(\tau_{j-1})$. Such sessions (whose optimal weight is determined according to the second case of Proposition 5) have a backlog clearing time which tends to infinity. If the sum of the weights assigned to QoS sensitive sessions is greater or equal to 1 then the sessions are not schedulable and the algorithm terminates.[18]

Step B3 Sessions which empty their backlog at the current checkpoint are moved to set $Empty_j$. This is necessary in order to compute the value of the NBSA bandwidth at τ_j ($\hat{C}(\tau_j^+)$); according to equation (5) the set of sessions that empty their backlog up to τ_j and their weights must be known.

As shown in the proof of Proposition 6, only sessions in Phi_{j-1} may empty their backlog at τ_j, and from the sessions in Phi_{j-1} these whose virtual clearing time at τ_{j-1} is equal to τ_j empty their backlog at τ_j. Thus, set $PE_j \subseteq Phi_{j-1}$ corresponds to the set of sessions which have been examined at a previous checkpoint, their weights have been determined at a previous checkpoint and empty their backlog at τ_j.

Step B4 As mentioned, the algorithm uses Proposition 5 in order to determine the optimal weights of the QoS sensitive sessions. According to Proposition 5, if τ_j is the first checkpoint, within *the examination interval of session* i, at which the *condition* is met for session i (that is, $\phi_i^-(\tau_j) \geq \phi_i^+(\tau_j)$ holds) then the optimal weight for session i is $\phi_i^{\pi_o} = \phi_i^-(\tau_j)$. At this step the algorithm checks whether the *condition* is met for two kind of sessions:

- Sessions which have already been examined at some previous checkpoint but did not fulfil the *condition* (that is, sessions in $Trans_{j-1}$). Set $TP_j \subseteq Trans_{j-1}$ contains the sessions that fulfil the *condition* for the first time at τ_j. TP_j is moved into set Phi_j at step **B.6**; the rest of the sessions in $Trans_{j-1}$ remain in set $Trans_j$.
- Sessions which are examined for the first time at τ_j (sessions for which $\tau_j = D_i$, that is τ_j is the beginning of their *examination interval*). From these sessions, those fulfilling the *condition* form the set $NP_j \subseteq Nex_{j-1}$ (sessions which are examined for the first time at τ_j and their weights can be determined at τ_j since they meet the *condition* at τ_j), while the rest of these sessions form the set $NT_j \subseteq Nex_{j-1}$ (sessions which are examined for the first time at the current checkpoint (τ_j) but their weights cannot determined at τ_j). At steps **B.5** and **B.6** the main sets are properly updated.

Step B5 The sets Nex_j and $Trans_j$ are determined by properly updating the corresponding sets at τ_{j-1}. More specifically, all sessions in Nex_{j-1} that *are examined at* τ_j (sets NP_j and NT_j) are removed from Nex_{j-1}; sessions

[17] More specifically, $A_i(0,\tau_{j-1}) = \sigma_i + \rho_i \tau_{j-1}$, $W_i(0,\tau_{j-1}) = \phi_i^{\pi_o} \hat{W}(0,\tau_{j-1})$, $\hat{W}(0,\tau_{j-1}) = \sum_{k=1}^{j-1} \hat{W}(\tau_{k-1},\tau_k)$, $\hat{W}(\tau_{k-1},\tau_k) = \hat{C}(\tau_{k-1}^+)(\tau_k - \tau_{k-1})$.

[18] All ϕ's are considered to be initially undefined. $\sum_{i \in QoS} \phi_i$ denotes the summation over all sessions that have been assigned weight by the algorithm.

Determine_$\pi_o(C_G, QoS)$
A. $j{=}0, Trans_0{=}Phi_0{=}Empty_0{=}\emptyset$, $Nex_0 = QoS$, $\tau_0 = 0$
B. repeat (B1.-B6.) until ($\phi_i = \phi_i^{\pi_o} \quad \forall i \in Nex_0$)
B1. $j = j+1$, $\tau_j = \min\{\min_{i \in Nex_{j-1}} D_i\,,\ \min_{i \in Phi_{j-1}} vct_i(\tau_{j-1})\}$
B2. If ($\tau_j = \infty$) then
{ $\forall i \in Trans_{j-1}$
{$\phi_i = \phi_i^{\pi_o} = \phi_i^{+}(\tau_{j-1})$, if $\sum_{i \in QoS} \phi_i \geq 1$ then error},
goto *C.*
}
B3. $PE_j = \{i \in Phi_{j-1} : vct_i(\tau_{j-1}) = \tau_j\}$,
$Empty_j = Empty_{j-1} \cup PE_j$
B4. $TP_j = \{i \in Trans_{j-1} :\ \phi_i^{-}(\tau_j) \geq\ \phi_i^{+}(\tau_j)\}$,
$NP_j = \{i \in Nex_{j-1} : D_i = \tau_j,\ \phi_i^{-}(\tau_j) \geq\ \phi_i^{+}(\tau_j)\}$,
$NT_j = \{i \in Nex_{j-1} : D_i = \tau_j,\ \phi_i^{-}(\tau_j) <\ \phi_i^{+}(\tau_j)\}$
B5. $Nex_j = Nex_{j-1} \backslash (NP_j \cup NT_j)$,
$Trans_j = (Trans_{j-1} \backslash TP_j) \cup NT_j$
B6. $\forall i \in NP_j \cup TP_j$
{$\phi_i = \phi_i^{\pi_o} =\ \phi_i^{-}(\tau_j)$, if $\sum_{i \in QoS} \phi_i \geq 1$ then error}
$Phi_j = (Phi_{j-1} \backslash PE_j) \cup NP_j \cup TP_j$
C. $\phi_{be} = 1 - \sum_{i \in QoS} \phi_i$

Fig. 3. Pseudocode of the optimal CAC algorithm. At step *A.* the algorithm initializes. Proposition 6 is applied at step *B1.* in order to determine the next checkpoint. At step *B2.* it is checked whether the second case of Proposition 5 is applicable. At step *B3.* the set of sessions that empty their backlog up to the current checkpoint is determined. At steps *B4.-B6.* it is checked for which sessions the first case of Proposition 5 applies (and thus, their optimal weights are determined). In addition, the sets used by the algorithm are properly updated. At step **C.** the weight of the best effort traffic session is computed.

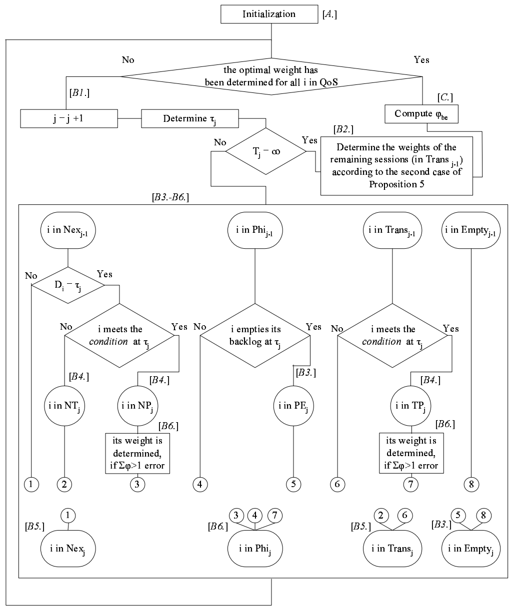


Fig. 4. Flowchart of the optimal CAC algorithm. The steps of the pseudocode at which the corresponding actions take place are illustrated within square brackets.

in $Trans_{j-1}$ that fulfill the *condition* at τ_j (set TP_j) are removed from $Trans_{j-1}$, while sessions in Nex_{j-1} that *are examined at* τ_j and do not meet the *condition* at τ_j (set NT_j) are added to $Trans_{j-1}$.

Step B6 At this step the weight assignment for the sessions whose weight is determined at τ_j (those fulfilling the *condition* for the first time at τ_j, that is, sessions in NP_j and sessions in NT_j) is performed. If the sum of the weights assigned to QoS sensitive sessions is greater or equal to 1 then the sessions are not schedulable and the algorithm terminates. In addition, set Phi_j is determined by removing from Phi_{j-1} the sessions that empty their backlog at τ_j (set PE_j) and by adding the sessions whose weights are determined at τ_j (sets NP_j and TP_j).

Step C The weight of the best effort traffic session ($\phi_{be}^{\pi_o}$) is computed.

At this point the following should be noted. For the sake of simplicity in the presentation, all checkpoints are treated "equally" in the pseudocode of the algorithm, in the sense that all sets of sessions are examined at all checkpoints. However, if the current checkpoint τ_j coincides with the delay bound of some session, without coinciding with the backlog clearing time of some session (that is $\tau_j = \min_{i \in Nex_{j-1}} D_i$, $\tau_j \neq \min_{i \in Phi_{j-1}} vct_i(\tau_{j-1})$), then:

- Sessions $i \in Trans_{j-1}$ need not be examined at τ_j, since TP_j is an empty set, that is, $\phi_i^{-}(\tau_j) \geq \phi_i^{+}(\tau_j)$ cannot hold for any session $i \in Trans_{j-1}$. By definition (construction) $\phi_i^{+}(\tau_{j-1}) > \phi_i^{-}(\tau_{j-1})$, $\forall i \in Trans_{j-1}$ holds, and from Claim 3 it is implied that $\phi_i^{+}(\tau_{j-1}) > \phi_i^{-}(\tau_j)$, $\forall i \in Trans_{j-1}$. In addition – and because no session empties its backlog at τ_j – $\hat{C}(\tau_j^{+}) = \hat{C}(\tau_{j-1}^{+})$ holds, implying that $\phi_i^{+}(\tau_j) = \phi_i^{+}(\tau_{j-1})$, $\forall i \in Trans_{j-1}$. Thus, $\phi_i^{+}(\tau_j) > \phi_i^{-}(\tau_j)$, $\forall i \in Trans_{j-1}$ and TP_j is an empty set.
- It is not necessary to compute the virtual clearing time of sessions in Phi_{j-1} since they are exactly the same as those computed at τ_{j-1}. (As explained $\hat{C}(\tau_j^{+}) = \hat{C}(\tau_{j-1}^{+})$ holds and given this condition it is straightforward to conclude from equation (15) that $vct_i(\tau_{j-1}) = vct_i(\tau_j)$, $\forall i \in Phi_{j-1}$.) Thus, in order to determine the minimum virtual clearing time at τ_j, it is sufficient to compute the virtual clearing time of sessions in NP_j and compare it with the minimum virtual clearing time of sessions in Phi_{j-1} as computed at τ_{j-1}. The minimum of these quantities is the minimum virtual clearing time at τ_j.

Thus, when τ_j does not coincide with the backlog clearing time of some session, only sessions i for which $\tau_j = D_i$ need to be examined and to be included in the proper set. (These sessions have not been examined up to that point by the algorithm.)

Finally it is noted that best effort traffic will not only be granted $\phi_{be}^{\pi_o} C_G$ but will receive more service on larger time scales, as QoS sensitive sessions empty their backlog.

IV. Pure QoS System

In a system where only QoS sensitive sessions are present the existence of an extra session, denoted as "dummy", may be assumed and the presented algorithm be applied. The modified algorithm for the pure QoS system is referred to as Modified Optimal CAC Algorithm (MOCA).

The MOCA is exactly the same as the optimal CAC algorithm for the BETA-GPS system except the check of the sum of the weights assigned to QoS sensitive sessions at steps B.2 and B.6, which should be replaced by ..."if ($\sum_{i \in QoS} \phi_i > 1$ or $\phi_i = 0$[19]) then error"... , since the

[19] according to the definition of GPS.

"dummy" session can be assigned a weight equal to zero. In addition, at step C. ϕ_{be} should be replaced by ϕ_d, d for "dummy".

The input of the MOCA is a traffic mix consisting only of QoS sensitive sessions i, $i = 1, \ldots, N$. If the MOCA finds a solution it returns as output a ϕ assignment ($\phi_1, \phi_2, ..., \phi_N$, ϕ_d). Obviously the QoS sensitive sessions can be admitted being assigned weights $(\phi_1, \phi_2, ..., \phi_N)$(or normalized to sum to one $(\phi_1, \phi_2, ..., \phi_N) \cdot (1 - \phi_d)^{-1}$). If the MOCA does not find a solution then this implies that a solution does not exist, since the MOCA minimizes $\sum_{i=1}^{N} \phi_i$. In this sense the MOCA can be considered as optimal for the pure QoS system.

A. Minimal bandwidth requirement

Another capability that could be required by a CAC scheme for the pure QoS system is to be able to compute the minimum capacity of the GPS server $C_{G(min)}$ required to support the N QoS sensitive sessions [20] (and the appropriate ϕ assignment). In this case the following proposition is useful.

Proposition 7: Suppose that an acceptable policy is computed by the MOCA in a pure QoS system where the GPS scheduler controls capacity C_G. QoS sensitive sessions, assigned the computed weights and being served by a GPS scheduler (of a pure QoS system) controlling capacity $C_G(1 - \phi_d)$, do meet their QoS requirements.

Proof: Assume that an acceptable policy ($\phi_1, \phi_2, ...,$ ϕ_N, ϕ_d) has been computed by the MOCA. If $\phi_d = 0$ the proposition holds. If $\phi_d \neq 0$ any (σ_d, ρ_d, D_d) can be chosen to describe the "dummy" session as long as it does conform to the assumption that the "dummy" session empties its backlog last in order and to the stability condition ($\sum_{i \in QoS \cup d} \rho_i \leq 1$). Let $\rho_d = \phi_d C_G$ and let σ_d and D_d be arbitrary but properly selected so that the "dummy" session empties its backlog last in order.

Since the greedy system corresponds to the worst case arrival scenario resulting in the maximum delay experienced by the QoS sensitive sessions, their requirements are satisfied under any other arrival scenario including the following: Consider the "dummy" session not starting greedy at the beginning of the system busy period but with $\sigma_d = 0$ and transmitting with a constant rate equal to $\rho_d = \phi_d \cdot C_G$. It is easily seen that this scenario is equivalent to a pure QoS system where the GPS server controls capacity equal to $C_G(1 - \phi_d)$. ■

Proposition 7 states that the QoS sensitive sessions would meet their requirements even in a more stringent environment, where the capacity controlled by the GPS server is equal to $C_G(1 - \phi_d)$. This indicates that if it is desirable to compute the minimum capacity of the GPS server ($C_{G(min)}$) required to support the N QoS sensitive sessions the following recursive process can be followed, which amounts to cutting slices of size $\phi_{d(n)} C_n$ from the capacity C_n (with $C_1 = C_G$) controlled by the GPS scheduler at the n-th iteration, until the last cutted slice becomes smaller than an arbitrary predefined small quantity ϵ.

$compute\_C_{G(min)}(Nex_0, C_G)$
$n = 1, C_{(1)} = C_G$
$for(;;)$
$\{\phi_{d(n)} = \phi_d(MOCA(Nex_0, C_{(n)}))$
$C_{(n+1)} = C_{(n)}(1 - \phi_{d(n)})$
if $(\phi_{d(n)} C_{(n)} < \epsilon)$ then $\{C_{G(min)} = C_{(n+1)}$, exit$\}$
$n = n + 1\}$

where Nex_0 is the traffic mix (consisting only of QoS sensitive sessions) and ϵ is an arbitrary small positive number. $\phi_{d(n)}$ is the weight assigned to the "dummy" session by the MOCA, assuming that the GPS scheduler controls capacity $C_{(n)}$. The process stops when $\phi_{d(n)} C_{(n)}$ becomes less than a predefined quantity ϵ. The feasibility of this process and the fact that it can approximate as closely as desired the minimum GPS capacity required to support the N QoS sensitive sessions can be easily concluded. In particular, let $\phi_{d(1)}$ be the weight assigned to the "dummy" session at the first iteration of the process. The weight assigned to the "dummy" session is the weight not needed by the QoS sensitive sessions. If $\phi_{d(1)} = 0$ then obviously the optimal solution is determined, since the MOCA minimizes $\sum_{i=1}^{N} \phi_i$. If $\phi_{d(1)} \neq 0$ then according to Proposition 7 bandwidth $(1 - \phi_{d(1)}) C_G$ is sufficient to satisfy QoS sensitive sessions requirements.

Applying MOCA at the second iteration leads to the computation of the maximum ϕ not needed by the Qos sensitive sessions for the more stringent system ($\phi_{d(2)}, \phi_{d(2)} \leq \phi_{d(1)}$ [21]). If $\phi_{d(2)} = 0$ the optimal solution is determined, else the process is repeated.

Due to the fact that $\phi_{d(n)}$ becomes smaller at each iteration the aforementioned process approximates as closely as desired the solution to the problem of minimization of the required capacity to admit the QoS sensitive sessions.

V. Discussion - Numerical results

A. Relation to the effective bandwidth-based CAC

Deterministic effective bandwidth ([9]) can be used in a straightforward way to give a simple and elegant CAC scheme. A similar approach is followed in [4] for the deterministic part of their analysis. The deterministic effective bandwidth of a (σ_i, ρ_i, D_i) session is given by $w_i^{\text{eff}} = \max\{\rho_i, \frac{\sigma_i}{D_i}\}$. It is easy to see that the requirements of the QoS sensitive sessions are satisfied if they are assigned weights such that $\phi_i C_G = w_i^{\text{eff}}$ ($\frac{\phi_i}{\phi_j} = \frac{w_i^{\text{eff}}}{w_j^{\text{eff}}}, \forall i, j \in QoS$). In this section the presented algorithm is compared with

[20] The capacity not used by the GPS server could be used by another server. The scheme under which such a partitioning could be realized is out of the scope of this paper.

[21] It is easy to prove that $\phi_{d(2)} \leq \phi_{d(1)}$ by contradiction. In particular, suppose that $\phi_{d(2)} > \phi_{d(1)}$. Since QoS sensitive sessions meet their delay bounds under $(\phi_{1(2)}, \ldots, \phi_{N(2)})$, $\sum_{i=1}^{N} \phi_{i(2)} = 1 - \phi_{d(2)}$ when the GPS server controls capacity $C_G(1 - \phi_{d(1)})$ they will not miss their delay bound when the GPS server controls capacity C_G, that is the policy $(\phi_{1(2)}, \ldots, \phi_{N(2)})$ is an acceptable policy for the case where the GPS server controls capacity C_G. This implies that $(\phi_{1(1)}, \ldots, \phi_{N(1)})$, $\sum_{i=1}^{N} \phi_{i(1)} = 1 - \phi_{d(1)}$ computed by MOCA at the first iteration was not the policy with the minimum $\sum_{i=1}^{N} \phi_i$ (contradiction).

the effective bandwidth-based CAC scheme. The effective bandwidth-based CAC is tighter than the CAC which is based on rate proportional weighting ($\frac{\phi_i}{\phi_j} = \frac{\rho_i}{\rho_j}$), which is used for comparison in [3] (see also [11]).

In addition, it is noted that the effective bandwidth-based CAC scheme can be considered as a special case of the presented algorithm. If the presented algorithm is denied the ability to "remember" which sessions and when they empty their backlog, the NBSA bandwidth, as computed by the algorithm, has a constant value which is equal to C_G and each checkpoint coincides with some session 's delay bound. It is easily seen that in this special case the algorithm would assign to session i weight equal to $\max\{\phi_i^-(\tau_{d_i}), \phi_i^+(\tau_{d_i})\}$, since the algorithm would be completely unable to keep track of the evolution of the greedy system. This implies [22] that $\phi_i C_G = \max\{\rho_i, \frac{\sigma_i}{Di}\} = w_i^{\text{eff}}$, that is the algorithm would assign to each session the same weight as the effective bandwidth-based CAC would.

B. Graphical interpretation

In this section an intuitive examination of the algorithm is attempted. The examination is based on the graphical representation of the mean values of the associated quantities.

Since the GPS scheduler is work conserving, $C_G \cdot t = \sum_{i \in TF} W_i(0,t)$,where $TF = QoS \cup be$ in the best effort aware system and $TF = QoS$ in the pure QoS system, holds for arbitrary t in the system busy period. This implies that:

$$C_G = \sum\nolimits_{i \in TF} \overline{W}_i(0,t) \tag{16}$$

where $\overline{W}_i(0,t) = W_i(0,t)/t$ is the mean work assigned to session i in $(0,t]$. The mean requirements of session i in the interval $(0,t]$ are:

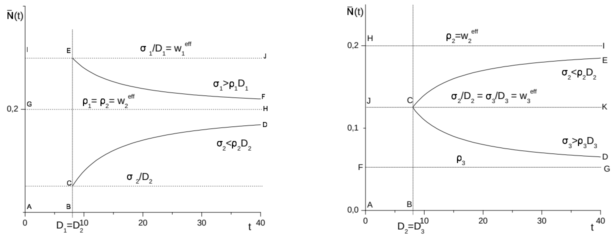
$$\overline{N}_i(0,t) = \begin{cases} \frac{\sigma_i + \rho_i(t-D_i)}{t} & t \geq D_i \\ 0 & t < D_i \end{cases} \tag{17}$$

which implies:

$$\partial \overline{N}_i(0,t)/\partial t = (\rho_i D_i - \sigma_i)/t^2 \quad t \geq D_i \tag{18}$$

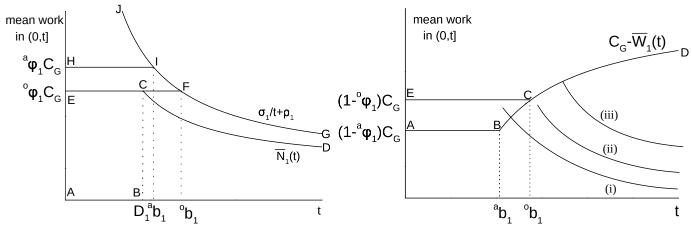
Although session i requirements are always an increasing function of time (for $t \geq D_i$), the mean requirements of a session are an increasing (decreasing) function of time if $\rho_i D_i > \sigma_i$ ($\rho_i D_i < \sigma_i$).

In Figure 5(a) the mean requirements of two sessions with $(\sigma_1, \rho_1, D_1) = (2, 0.2, 8)$ and $(\sigma_2, \rho_2, D_2) = (1, 0.2, 8)$ (arbitrary units) and in Figure 5(b) the mean requirements of two sessions with $(\sigma_2, \rho_2, D_2) = (1, 0.2, 8)$ and (σ_3, ρ_3, D_3) $= (1, 0.05, 8)$ are shown. It is evident that in the framework of the deterministic effective bandwidth-based CAC scheme, CAC decisions are based on the maximum (or more precisely, the supremum) of the mean requirements of the sessions, since the dependencies among the sessions are not considered, that is, the fact that more bandwidth becomes available to the still backlogged sessions as some sessions empty their backlog. This is the main strength (due to the implied simplicity) and at the same time the main weakness (due to resource under-utilization) of this scheme.



(a) (b)

Fig. 5. Mean requirements and deterministic effective bandwidth of (a) two sessions with $(\sigma_1, \rho_1, D_1) = (2, 0.2, 8)$ and $(\sigma_2, \rho_2, D_2) = (1, 0.2, 8)$ (b) two sessions with $(\sigma_2, \rho_2, D_2) = (1, 0.2, 8)$ and $(\sigma_3, \rho_3, D_3) = (1, 0.05, 8)$.



(I.) (II.)

Fig. 6. (I.) Mean service received by the session which empties its backlog 1st in order under the optimal policy π_o and under the policy π_a. (II.) Mean work assigned to other sessions under the optimal policy π_o and under some other policy π_a.

Condition $W_i(0,t) \geq N_i(0,t)$, $t \geq D_i$, which must hold in order for QoS sensitive session i requirements to be satisfied, implies that $\overline{W}_i(0,t) \geq \overline{N}_i(0,t)$, $t \geq D_i$. For $t \geq e_i$, where e_i is the backlog clearing time of session i, session i is served with a constant rate (equal to ρ_i). Thus,

$$\overline{W}_i(0,t) = \frac{\sigma_i + \rho_i t}{t} = \frac{\sigma_i}{t} + \rho_i, \quad t \geq e_i \tag{19}$$

In Figure 6-(I.) the mean service received by the QoS sensitive session which empties its backlog first in order (it is assumed that it is only one for simplicity and will be referred to as session 1) under the optimal policy π_o (curve EFG) and under some other policy, π_a (curve HIG), are shown[23].

$\overline{W}_1(0,t)$ is constant (linear parts EF and HI for policies π_o and π_a respectively), as long as the session is backlogged. According to equation (19) the service curve of the session coincides with the curve $\frac{\sigma_1}{t} + \rho_1$ for $t \geq e_1 = \mathrm{b}_1$. This implies that the backlog clearing time of the session corresponds to point F under π_o and I under π_a. From Figure 6-(I.) it is evident that session 's requirements (curve ABCD) are satisfied under both policies.

[22] From Definition 4 it is concluded that in this special case (where the NBSA bandwidth is considered equal to C_G) $\phi_i^-(\tau_{d_i}) = \frac{\sigma_i}{C_G D_i}$ and $\phi_i^+(\tau_{d_i}) = \frac{\rho_i}{C_G}$ for all QoS sensitive sessions .

[23] It is noted that it is straightforward to prove, using the properties of π_o, that under the optimal policy for the BETA-GPS system, the session which empties its backlog first in order has mean requirements which are a decreasing function of time for $t \geq D_i$ (given that such a session is part of the traffic mix).

TABLE I
Sessions under investigation

Case 1	s_1	s_2		s_3
Case 2	s_1		s_2	s_3
σ_i	0.04	0.16	0.04	0.64
ρ_i	0.01	0.01	0.04	0.01
D_i	1	4	4	16
w_i^{eff}	0.04	0.04	0.04	0.04

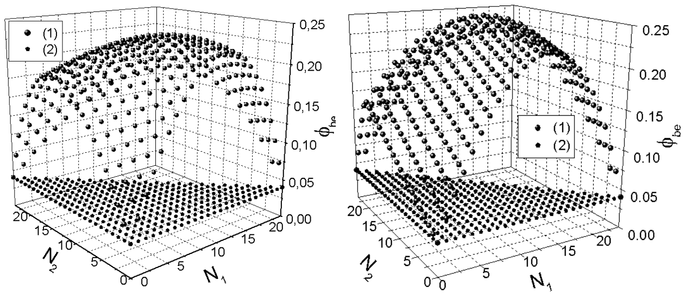
In Figure 6-(II.) the quantity $C_G - \overline{W}_1(0,t)$ (curve ABCD under π_a, curve ECD under π_o), which represents the mean work available to other sessions, is shown. On this figure the following observations can be made.
1. Under π_o the requirements of a session with mean requirements of the form (i) can be satisfied, while under π_a they can not.
2. Under both policies a sessions with mean requirements of the form (ii) can be admitted. In addition, under both policies the best effort traffic is assigned the same fraction of $C_G - \overline{W}_1(t)$ for $t >{}^{o}\mathrm{b}_1$ but this corresponds to a greater weight under π_o. The former observations illustrated the optimality of the presented algorithm.
3. It may also be observed that under both policies a session with mean requirements of the form (iii) can be admitted. In this example the minimum required server capacity has the same value under both policies and $\sum_{i=1}^{2} \phi_i = 1$. This illustrates that the optimal solution for the pure QoS system is, in general, not unique, since there exist more than one policies under which $\sum_{i=1}^{N} \phi_i$ is minimized.

Finally, it should be mentioned that no session with a deterministic effective bandwidth greater than that corresponding to the straight line $AB\ldots$ in Figure 6-(II.) would be admitted along with session 1 (whose deterministic effective bandwidth corresponds to the straight line $ECF\ldots$ in Figure 6-(I.)).

C. Numerical results

In this section some numerical results are presented for the BETA-GPS system and for the pure QoS system.

BETA-GPS system: Although the algorithm can support an arbitrary number of delay classes the numerical investigation is limited to the case of three delay classes. The analysis for this case is simple enough to follow and provide some insight. Two cases are investigated.
Case 1: The traffic mix consists only of QoS sensitive sessions whose mean requirements are a decreasing function of time for $t \geq D_i, \forall i$. The sessions under investigation for this case are shown in Table I. All quantities are considered normalized with respect to the link capacity C.
In order to compare the presented algorithm with the effective bandwidth-based CAC scheme the following scenario is considered. The effective bandwidth-based CAC scheme admits the maximum number of sessions under the constraint that a nonzero weight remains to be assigned to best effort traffic. From Table I it can be seen that the effective bandwidth of each QoS sensitive session is 1/25 of the server's capacity (which is considered to be equal to the link capacity ($C_G = C$)), implying that for the BETA-GPS system at most 24 QoS sensitive sessions can be admitted under the effective bandwidth-based CAC scheme. This means that $N_1 + N_2 + N_3 = 24$ must hold and that the best effort traffic is assigned weight equal to 0.04 for each such triplet (N_1, N_2 and N_3 denote the number of admitted sessions of type s_1,s_2 and s_3 respectively).



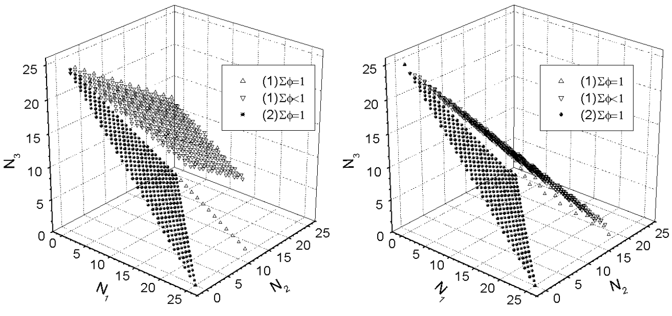
Case 1 *Case 2*

Fig. 7. Weight assigned to the best effort traffic according to the (1) optimal BETA-GPS CAC (2) effective bandwidth-based CAC scheme, both under the constraint $N_1 + N_2 + N_3 = 24$. The minimum guaranteed rate to the best effort traffic is $\phi_{be} C_G$

For each triplet (N_1, N_2, N_3), $N_1 + N_2 + N_3 = 24$, the weight assigned to the best effort traffic by the optimal BETA-GPS CAC scheme is computed. The results are illustrated in Figure 7.
It is easily seen that the improvement achieved by the optimal algorithm depends on the diversity of the traffic mix. For heterogeneous traffic mixes a significant improvement is achieved. On the other hand for pure homogeneous traffic mixes (only one type of session) the optimal algorithm can not result in any improvement.
Case 2*:* The traffic mix consists of both types of QoS sensitive sessions (with increasing and decreasing mean requirements). To demonstrate this case session s_2 is replaced by a session with the same effective bandwidth but with mean requirements which are an increasing function of time (see Table I) under the deterministic effective bandwidth-based CAC scheme.
The same scenario as in *Case 1* is followed. The results are illustrated in Figure 7. The achieved improvement is less than in *Case 1*, in particular when sessions of type s_1 are a minor part of the traffic mix.
The above results suggest that the presented algorithm for the BETA-GPS system may achieve a significant improvement, compared to the effective bandwidth-based CAC scheme, when the traffic mix is heterogeneous and consists of bursty sessions[24], that is sessions whose mean requirements are a decreasing function of time.

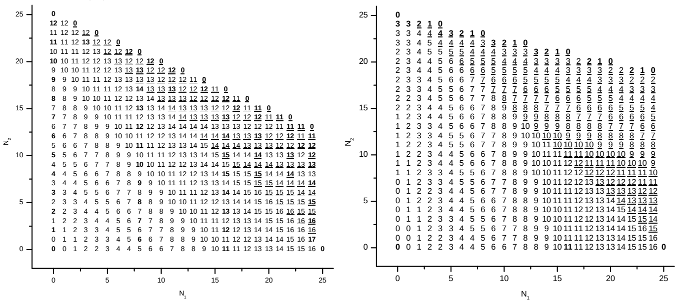
Pure QoS system: For this system the comparison is

[24] At this point it should be noted that sessions are characterized as "burtsy" and "non-bursty" from the perspective of the required service rate (and not from the perspective of the arrival rate); as a threshold the mean rate of the session is used. More specifically, the service rate of a session whose mean requirements are a decreasing function of time must exceed (for some time interval) its mean rate in order for the QoS requirements of the sessions to be met and, thus, some kind of burstiness is expected to occur in the session's service rate; such sessions are characterized as bursty.



Case 1 *Case 2*

Fig. 8. Maximum number of s_3 sessions that can be admitted as a function of the number of s_1 and s_2 sessions according to (1) MOCA and (2) the effective bandwidth-based CAC scheme.



Case 1 *Case 2*

Fig. 9. Difference of the maximum number of s_3 sessions that can be admitted according to MOCA minus the maximum number of s_3 sessions that can be admitted according to the effective bandwidth-based CAC, as a function of the number of s_1 and s_2 sessions. The underlined numbers indicate cases in which the effective bandwidth-based CAC scheme would not accept even the corresponding number of s_1 and s_2 sessions; in such cases the underlined numbers are the maximum number of s_3 sessions that can be admitted according to MOCA. Bolded numbers correspond to the triplets for which $\sum_{i \in QoS} \phi_i = 1$ according to MOCA.

based on the maximum number of sessions that can be admitted for a given capacity of the server. For the traffic mix the same cases as for the BETA-GPS system are considered. It is easily seen from Table I that for the pure QoS system the maximum number of sessions that can be admitted by the effective bandwidth-based CAC scheme is 25, i.e.: $N_1 + N_2 + N_3 = 25$ holds.

Case 1*:* In Figure 8 the maximum number of s_3 sessions (N_3) that can be admitted are shown as a function of the number of s_1 sessions (N_1) and s_2 sessions (N_2) according to the presented algorithm (1) and according to the effective bandwidth-based CAC (2). For some triplets (N_1, N_2, N_3) the sum of the weights assigned to QoS sensitive sessions is equal to one implying that the best effort traffic can not be assigned a nonzero weight. Such triplets are valid for a pure QoS system but not for a BETA-GPS system. Any of the (N_1-1, N_2, N_3), (N_1, N_2-1, N_3), (N_1, N_2, N_3-1) triplets would be valid for a BETA-GPS system. It is noted that all triplets for the effective bandwidth-based CAC are such ($\sum_{i \in QoS} \phi_i = 1$) and that the triplets for which $\sum_{i \in QoS} \phi_i = 1$ according to the presented algorithm are denoted with $\triangle$.

Case 2*:* The acceptable triplets (N_1, N_2, N_3) for this case are shown in Figure 8. It is obvious that the achieved improvement is less due to the less bursty nature of s_2 sessions.

Figure 9 provides an alternative illustration of the data presented in Figure 8. More specifically, it depicts the difference of the maximum number of s_3 sessions that can be admitted according to MOCA minus the maximum number of s_3 sessions that can be admitted according to the effective bandwidth-based CAC, as a function of the number of s_1 and s_2 sessions. The underlined numbers indicate cases in which the effective bandwidth-based CAC scheme would not accept even the corresponding number of s_1 and s_2 sessions; in such cases the underlined numbers are the maximum number of s_3 sessions that can be admitted according to MOCA. Bolded numbers correspond to the triplets for which $\sum_{i \in QoS} \phi_i = 1$ according to MOCA.

Finally, it is noted that the case where the traffic mix consists of QoS sensitive session whose mean requirements are an increasing function of time for $t \geq D_i, \forall i$ is not considered, since in this special case the problem of CAC is trivial, that is the stability condition is sufficient to ensure that the QoS sensitive sessions do not violate their delay bounds and obviously the described optimal CAC algorithm does not improve the resource utilization.

D. Complexity of the algorithm

Consider the following simplistic implementation of the algorithm, which is in direct correspondence with the pseudocode of the algorithm. The state of the QoS sensitive sessions is kept in a $N \times 6$ matrix.[25] Each row of the matrix corresponds to a QoS sensitive session $i \sim (\sigma_i, \rho_i, D_i)$ and includes the following fields: σ_i, ρ_i, D_i, ϕ_i, $vct_i(\tau_{j-1})$, $set_i(\tau_{j-1})$, $set_i(\tau_j)$, where ϕ_i is the weight assigned to session i, $vct_i(\tau_{j-1})$ is the virtual clearing time of session i at the $j-1$ iteration and $set_i(\tau_{j-1})$ ($set_i(\tau_j)$) indicates the set that session i belongs to at the $j-1$ (j) iteration. In addition, assume that the values of the quantities τ_j, $\hat{C}(\tau_j^+)$, $\hat{W}(0, \tau_j)$, $\sum_{i \in QoS} \phi_i$ for each checkpoint are stored.

Initially, the sessions are sorted with respect to delay in ascending order, incurring a complexity of $O(N \log N)$. At each iteration of the algorithm a checkpoint is examined. It can easily be verified that each step of each iteration is of complexity O(N) (or less). More specifically, computing the quantities $\phi_i^-(\tau_j)$, $\phi_i^+(\tau_j)$, $vct_i(\tau_{j-1})$ for a session i can be done in O(1) and, thus, computing these quantities for all sessions examined at an iteration can be done in O(N). The update of a set corresponds to properly adjusting the values of $set_i(\tau_{j-1})$ and $set_i(\tau_j)$ and is of complexity O(N). Finding the minimums at step B.1 and the update of $\hat{C}(\tau_j^+)$ (which can also take place during step B.3), are of complexity O(N), while the update of $\hat{W}(0, \tau_j)$ is an O(1) operation.

Thus, the update of the system at each checkpoint is performed in $O(N)$ complexity. There are at most $2N$ checkpoints, implying that the computational complexity of the presented algorithm is $O(N^2)$.

The increased complexity of the proposed algorithm as compared to the deterministic effective bandwidth-based

[25] A more realistic implementation would be based on linked lists and would execute some steps concurrently and not sequentially (for example, steps B.4-B.6).

CAC scheme (complexity of $O(N)$), although not prohibiting in practice (especially for a small number of sessions N), makes necessary to further investigate the tradeoff between the cost in terms of complexity and the gain in terms of performance.

Finally, it should be noted that the problem of the optimal weight assignment in the multiple node case is expected to be far more complex. This is mainly due to the fact that in order to retain optimality the ϕ assignment must be modified every time a new connection arrives to the system.

VI. Conclusions

In this paper the problem of allocating optimal weights to sessions being served according to the Generalized Processor Sharing scheduling discipline at a single node has been addressed. The derivation of the optimal solution is based on a procedure capable of transforming any acceptable weight allocation to the optimal one; based on this procedure, a CAC algorithm that computes the optimal weight assignment directly has been derived.

Since the proposed CAC scheme is optimal it bounds the resource utilization that can be achieved by a GPS scheduler at a single node and, thus, helps in comparing the efficiency of other simpler CAC schemes and the comparison of GPS with other scheduling disciplines. For example it can be used to investigate how much worse GPS is as compared to Earliest Deadline First (EDF), which is known to be the optimal scheduler for the single node case.

Apart from its value as a performance bound, whether the $O(N^2)$ optimal algorithm retains enough gain to be used in practice in place of $O(N)$ deterministic effective bandwidth-based algorithm, remains to be investigated.

In addition, it should be noted that in order to be able to sustain an optimal performance in a dynamic environment (new sessions arriving, other ending) the algorithm has to be re-executed upon any change in the traffic mix in order to recompute the optimal weights. This requirement emphasizes an inherent drawback of the GPS scheduling discipline, which is that it cannot fully exploit the available bandwidth under a static allocation of weights.

References

[1] A. Parekh and R. 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.

[2] A. Parekh and R. 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.

[3] R. Szabó, P. Barta, F.Németh, and J.Biró. Call admission control in generalized processor sharing (gps) schedulers using non-rate proportional weighting of sessions. *Proceedings of IEEE INFOCOM*, 3:1243–1252, March 2000.

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

[5] Z. Zhang, D. Towsley, and J. Kurose. Statistical analysis of the generalized processor sharing scheduling discipline. *in Proc. of ACM SIGCOMM*, September 1994.

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

[7] Z. 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 (Special Issue on Scheduling)*, 1(4), 1998.

[8] R. Szabo, P. Barta, J.Biró, F.Németh, and C-G. Perntz. Non-rate proportional weighting of genaralized processor sharring schedulers. *in Proc. of GLOBECOM*, 2:1334–1339, December 1999.

[9] J.-Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory*, 44(3):1087–1096, May 1998.

[10] L. Georgiadis, R. Guérin, V. Peris, and K. Sivarajan. Efficient network qos provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, 4(4):482–501, August 1996.

[11] G. Urvoy, G. Hébuterne. An efficient weight assignment process for gps servers. *in Proc. of ECUMN 2000*, 171–178, 2000.

[12] O. Yaron and M. Sidi. Generalized Processor Sharing Networks with Exponentially Bounded Burstiness Arrivals. *in Proc. of INFOCOM '94*, 628-634, Jan 1996.

[13] G. de Veciana and G. Kesidis. Bandwidth Allocation for Multiple Qualities of Service using Generalized Processor Sharing. *IEEE Transactions on Information Theory* , 42(1):268-72, Jan 1996.

[14] K. Kumaran, G.E. Margrave, D. Mitra, and K.R. Stanley. Novel techniques for the design and control of Generalized Processor Sharing schedulers for multiple QoS classes. *in Proc. of IEEE INFOCOM 2000* , 2000.

[15] Nandita D., Joy Kuri, and H. S. Jamadagni Optimal Call Admission Control in Generalized Processor Sharing (GPS) Schedulers. *in Proc. of IEEE INFOCOM 2001*, 2001.

[16] Antonis Panagakis and Ioannis Stavrakakis. Optimal call admission control under generalized processor sharing scheduling. *in Proc. of IWQoS2001*, 2001.

VII. Appendix

Due to space limitations the following notations are used in the proof of Proposition 1.

$$ {}^{y}_{x}\mathcal{P}^{w}_{z} = \sum_{i\in {}^{y}\mathrm{I}^{w}_{z}} {}^{x}\phi_i, {}^{y}_{x}\mathcal{F}^{w}_{z} = \sum_{i\in {}^{y}\mathrm{I}^{w}_{z}} {}^{x}\mathrm{f}_i,\ {}^{y}\mathcal{R}^{w}_{z} = \sum_{i\in {}^{y}\mathrm{I}^{w}_{z}} \rho_i $$

$$ \overline{{}^{y}_{x}\mathcal{P}^{w}_{z}} = 1 - {}^{y}_{x}\mathcal{P}^{w}_{z},\ \overline{{}^{y}_{x}\mathcal{P}^{w}_{z} + {}^{q}_{p}\mathcal{P}^{r}_{s}} = 1 - {}^{y}_{x}\mathcal{P}^{w}_{z} - {}^{q}_{p}\mathcal{P}^{r}_{s} $$

$$ \overline{{}^{y}\mathcal{R}^{w}_{z}} = C_G - {}^{y}\mathcal{R}^{w}_{z},\ \overline{{}^{y}\mathcal{R}^{w}_{z} + {}^{q}\mathcal{R}^{r}_{s}} = C_G - {}^{y}\mathcal{R}^{w}_{z} - {}^{q}\mathcal{R}^{r}_{s} $$

$$ {}^{y}_{x}\mathcal{C}^{w}_{z} = {}^{y}\hat{\mathrm{C}}({}^{x}\mathrm{b}^{w}_{z}),\ l^{z,x}_{w,k} = {}^{z}\mathrm{b}_w - {}^{x}\mathrm{b}_k $$

$$ {}^{y}\hat{\mathrm{W}}^{x,k}_{z,w} = {}^{y}\hat{\mathrm{W}}({}^{z}\mathrm{b}_w, {}^{x}\mathrm{b}_k),\ {}^{y}_{x}\mathcal{W}^{w}_{z} = \sum_{m=z}^{w} {}^{y}\hat{\mathrm{W}}^{x,m}_{x,m-1} $$

Proof of Proposition 1 (Section III-A)

Only the proof for steps (II.1.a) and (II.1.b) is provided. The proof for step (II.2.a) is similar.
First it is proved that the "conditional exchange of weights" between sessions in ${}^{a}\mathrm{I}_k$ and sessions in ${}^{a}\mathrm{I}^{F}_{k}$ is feasible, that is, the total decrease of the weights of sessions in ${}^{a}\mathrm{I}_k$ is sufficient to balance the necessary total increase of the weights of sessions in ${}^{a}\mathrm{I}^{F}_{k}$ so that the condition (see (II.1)) is met. It is also shown that the best effort traffic is assigned a greater weight under π_b, that is π_b is more efficient than π_a. Then it is shown that the QoS sensitive sessions do not miss their delay bound under π_b.

Assume that the described "conditional exchange of weights" between sessions in ${}^{a}\mathrm{I}_k$ and sessions in ${}^{a}\mathrm{I}^{F}_{k}$ is feasible. Let ${}^{a}\phi_i$ denote the weight assigned to session i under π_a. Under π_b sessions in ${}^{a}\mathrm{I}^{P}_{k}$ preserve their position in ϕ- space, sessions in ${}^{a}\mathrm{I}_k$ are compressed in ϕ- space and sessions in

${}^{a}\mathrm{I}_k^F$ are decompressed in ϕ- space, i.e.:

$$
\begin{aligned}
&{}^{b}\phi_i = {}^{a}\phi_i \quad ,i \in {}^{a}\mathrm{I}_k^P; \quad {}^{b}\phi_i = {}^{a}\phi_i - {}^{b}\mathrm{f}_i \quad ,i \in {}^{a}\mathrm{I}_k;\\
&{}^{b}\phi_i = {}^{a}\phi_i + {}^{b}\mathrm{f}_i \quad ,i \in {}^{a}\mathrm{I}_k^F; \quad {}^{b}\mathrm{f}_i > 0 \quad \forall i \in QoS
\end{aligned} \tag{20}
$$

–Sessions in ${}^{a}\mathrm{I}_k \cup {}^{a}\mathrm{I}_k^F$ are backlogged at ${}^{a}\mathrm{b}_{k-1}$ under π_b

If step (II.1.a) is followed ${}^{b}\mathrm{I}_k = {}^{a}\mathrm{I}_k \cup {}^{a}\mathrm{I}_{k+1}$ and ${}^{b}\mathrm{b}_k > {}^{a}\mathrm{b}_k$ holds for the (intermediate) policy π_b. If step (II.1.b) is followed ${}^{a}\mathrm{I}_k = {}^{b}\mathrm{I}_k$ but ${}^{b}\mathrm{b}_k > {}^{a}\mathrm{b}_k$. In both cases sessions in ${}^{a}\mathrm{I}_k$ empty their backlog at ${}^{b}\mathrm{b}_k$ under π_b, ${}^{b}\mathrm{b}_k > {}^{a}\mathrm{b}_k$ and sessions in ${}^{a}\mathrm{I}_k^F$ are assigned the same amount of work under π_b and under π_a over the interval $(0, {}^{b}\mathrm{b}_k]$. This means that since sessions in ${}^{a}\mathrm{I}_k \cup {}^{a}\mathrm{I}_k^F$ are still backlogged at time ${}^{a}\mathrm{b}_{k-1}$ under π_a they are backlogged at time ${}^{a}\mathrm{b}_{k-1}$ (and $\forall t < {}^{a}\mathrm{b}_{k-1}$) under π_b too.

–Sessions in ${}^{a}\mathrm{I}_k^P$ preserve their position in t- space

Sessions in ${}^{a}\mathrm{I}_k^P$ preserve their position in ϕ-space (they are assigned the same weights under π_a and π_b) and all sessions in ${}^{a}\mathrm{I}_k \cup {}^{a}\mathrm{I}_k^F$ (and the best effort traffic) empty their backlog for $t \geq {}^{a}\mathrm{b}_k > {}^{a}\mathrm{b}_{k-1}$, under π_a and under π_b, by assumption. Obviously ${}^{a}\mathrm{b}_0 = {}^{b}\mathrm{b}_0 = 0$ and ${}^{b}\hat{\mathrm{C}}({}^{a}\mathrm{b}_0^+) = {}^{a}\hat{\mathrm{C}}({}^{a}\mathrm{b}_0^+) = C_G$. Since ${}^{b}\phi_i = {}^{a}\phi_i \quad ,\forall i \in {}^{a}\mathrm{I}_k^P$ it is implied that ${}^{b}\mathrm{b}_1 = {}^{a}\mathrm{b}_1$, ${}^{b}\mathrm{I}_1 = {}^{a}\mathrm{I}_1$ and ${}^{b}\hat{\mathrm{C}}({}^{a}\mathrm{b}_1^+) = {}^{a}\hat{\mathrm{C}}({}^{a}\mathrm{b}_1^+)$. Making similar thoughts it is easily concluded that:

$$
{}^{b}\mathrm{b}_m = {}^{a}\mathrm{b}_m \quad \wedge \quad {}^{b}\mathrm{I}_m = {}^{a}\mathrm{I}_m \quad ,m = 0, \ldots, k-1 \tag{21}
$$

$$
\text{and } {}^{b}_{b}\mathcal{C}_m^+ = {}^{a}_{a}\mathcal{C}_m^+ \quad ,m = 0, \ldots, k-1 \tag{22}
$$

This means that in both cases (step (II.1.a) or (II.1.b)) sessions in ${}^{a}\mathrm{I}_k^P$ preserve their position in t- space.

–Computation of weight differences (${}^{b}\mathrm{f}_i$s')

CASE A: ${}^{a}\mathrm{b}_{k+1} < \infty$. Sessions in ${}^{a}\mathrm{I}_k$ empty their backlog at ${}^{a}\mathrm{b}_k$ under π_a. This implies $W_i(0, {}^{a}\mathrm{b}_k) = A_i(0, {}^{a}\mathrm{b}_k)$ $\forall i \in {}^{a}\mathrm{I}_k$, i.e.:

$$
{}^{a}\phi_i \; {}^{a}_{a}\mathcal{W}_1^k = \sigma_i + \rho_i \, {}^{a}\mathrm{b}_k \quad , i \in {}^{a}\mathrm{I}_k \tag{23}
$$

Sessions in ${}^{b}\mathrm{I}_k$ empty their backlog at ${}^{b}\mathrm{b}_k$ under π_b, i.e.:

$$
{}^{b}\phi_i \; {}^{b}_{b}\mathcal{W}_1^k = \sigma_i + \rho_i \, {}^{b}\mathrm{b}_k \quad , i \in {}^{b}\mathrm{I}_k \tag{24}
$$

In both cases (steps (II.1.a) and (II.1.b)) under examination ${}^{a}\mathrm{I}_k \subseteq {}^{b}\mathrm{I}_k$, implying that equation (24) holds $\forall i \in {}^{a}\mathrm{I}_k$. Subtracting (23) from (24) implies:

$$
\text{Cond. 1:} \; {}^{b}\phi_i \; {}^{b}_{b}\mathcal{W}_1^k - {}^{a}\phi_i \; {}^{a}_{a}\mathcal{W}_1^k = \rho_i l_{k,k}^{a,b}, i \in {}^{a}\mathrm{I}_k \tag{25}
$$

Sessions in ${}^{a}\mathrm{I}_k^F$ are assigned the same amount of work up to time ${}^{b}\mathrm{b}_k$ under π_a and under π_b, i.e.:

$$
\text{Cond. 2:} \; {}^{b}\phi_i \; {}^{b}_{b}\mathcal{W}_1^k = {}^{a}\phi_i \left({}^{a}_{a}\mathcal{W}_1^k + {}^{a}\hat{\mathrm{W}}_{a,k}^{b,k} \right), i \in {}^{a}\mathrm{I}_k^F \tag{26}
$$

From (21) and (22) it is implied that:

$$
{}^{b}\hat{\mathrm{W}}_{b,m-1}^{b,m} = {}^{a}\hat{\mathrm{W}}_{a,m-1}^{a,m} \quad ,m = 1, \ldots, k-1 \tag{27}
$$

$$
\text{Using (22):} \; {}^{b}\hat{\mathrm{W}}_{b,k-1}^{b,k} = {}^{a}_{a}\mathcal{C}_{k-1}^+ \, l_{k,k}^{a,b} + {}^{a}\hat{\mathrm{W}}_{a,k-1}^{a,k} \tag{28}
$$

$$
\text{(27), (28) imply that:} \; {}^{b}_{b}\mathcal{W}_1^k = {}^{a}_{a}\mathcal{W}_1^k + {}^{a}_{a}\mathcal{C}_{k-1}^+ \, l_{k,k}^{a,b} \tag{29}
$$

$$
\text{In addition it is obvious that:} \; {}^{a}\hat{\mathrm{W}}_{a,k}^{b,k} = {}^{a}_{a}\mathcal{C}_k^+ \, l_{k,k}^{a,b} \tag{30}
$$

Conditions 1 and 2, using (29) and (30) become:

$$
{}^{b}\mathrm{f}_i \left({}^{a}_{a}\mathcal{W}_1^k + {}^{a}_{a}\mathcal{C}_{k-1}^+ \, l_{k,k}^{a,b} \right) = \left({}^{a}\phi_i \; {}^{a}_{a}\mathcal{C}_{k-1}^+ - \rho_i \right) l_{k,k}^{a,b}, i \in {}^{a}\mathrm{I}_k \tag{31}
$$

$$
{}^{b}\mathrm{f}_i \left({}^{a}_{a}\mathcal{W}_1^k + {}^{a}_{a}\mathcal{C}_{k-1}^+ \, l_{k,k}^{a,b} \right) = {}^{a}\phi_i \left({}^{a}_{a}\mathcal{C}_k^+ - {}^{a}_{a}\mathcal{C}_{k-1}^+ \right) l_{k,k}^{a,b}, i \in {}^{a}\mathrm{I}_k^F \tag{32}
$$

Conditions 1 and 2 (equations (31) and (32)) provide the necessary relations to determine the ${}^{b}\mathrm{f}_i$'s under the assumption that the "conditional exchange of weights" is feasible. In order to prove that the assumption holds it is sufficient to show that the total decrease of the weights of sessions in ${}^{a}\mathrm{I}_k$ is sufficient to balance the total increase of the weights of sessions in ${}^{a}\mathrm{I}_k^F$.

–The decrease of the weights of sessions in ${}^{a}\mathrm{I}_k$ is greater than the increase of the weights of sessions in ${}^{a}\mathrm{I}_k^F$

Summing over all $i \in {}^{a}\mathrm{I}_k$ equation (31) gives:

$$
{}^{a}_{b}\mathcal{F}_k \left({}^{a}_{a}\mathcal{W}_1^k + {}^{a}_{a}\mathcal{C}_{k-1}^+ \, l_{k,k}^{a,b} \right) = \left({}^{a}_{a}\mathcal{P}_k \; {}^{a}_{a}\mathcal{C}_{k-1}^+ - {}^{a}\mathcal{R}_k \right) l_{k,k}^{a,b} \tag{33}
$$

After some algebra the following equality can be verified:

$$
{}^{a}_{a}\mathcal{P}_k \; {}^{a}_{a}\mathcal{C}_{k-1}^+ - {}^{a}\mathcal{R}_k = \left({}^{a}_{a}\mathcal{C}_k^+ - {}^{a}_{a}\mathcal{C}_{k-1}^+ \right) \left(1 - {}^{a}_{a}\mathcal{P}_k - {}^{a}_{a}\mathcal{P}_k^P \right) \tag{34}
$$

Dividing (32) by (33), due to (34), results in:

$$
{}^{b}\mathrm{f}_i = {}^{a}\phi_i \; {}^{a}_{b}\mathcal{F}_k \left(1 - {}^{a}_{a}\mathcal{P}_k - {}^{a}_{a}\mathcal{P}_k^P \right)^{-1}, \quad i \in {}^{a}\mathrm{I}_k^F \tag{35}
$$

Equation (35) is the main result of this section. Summing over all sessions in ${}^{a}\mathrm{I}_k^F$ (and using $\sum_{i \in QoS} {}^{a}\phi_i + {}^{a}\phi_{be} = 1$):

$$
{}^{a}_{b}\mathcal{F}_k^F = {}^{a}_{a}\mathcal{P}_k^F \left({}^{a}\phi_{be} + {}^{a}_{a}\mathcal{P}_k^F \right)^{-1} \; {}^{a}_{b}\mathcal{F}_k \Leftrightarrow {}^{a}_{b}\mathcal{F}_k^F < {}^{a}_{b}\mathcal{F}_k \tag{36}
$$

This means that the total decrease of the weights of sessions in ${}^{a}\mathrm{I}_k$ is greater than the total increase of the weights of sessions in ${}^{a}\mathrm{I}_k^F$. So the described "conditional exchange of weights" is feasible. In addition the best effort traffic is assigned a greater weight under the intermediate policy π_b:

$$
{}^{b}\phi_{be} = 1 - \sum_{i \in QoS} {}^{b}\phi_i = 1 - {}^{b}_{b}\mathcal{P}_k^P - {}^{b}_{b}\mathcal{P}_k - {}^{b}_{b}\mathcal{P}_k^F =
$$

$$
\overline{{}^{a}_{a}\mathcal{P}_k^P} - {}^{a}_{a}\mathcal{P}_k + {}^{a}_{b}\mathcal{F}_k - {}^{a}_{a}\mathcal{P}_k^F - {}^{a}_{b}\mathcal{F}_k^F = {}^{a}\phi_{be} - {}^{a}_{b}\mathcal{F}_k^F + {}^{a}_{b}\mathcal{F}_k > {}^{a}\phi_{be} \tag{37}
$$

$$
\text{(35) imply that:} \; {}^{b}\mathrm{f}_i ({}^{b}\mathrm{f}_j)^{-1} = {}^{a}\phi_i ({}^{a}\phi_j)^{-1}, \quad i,j \in {}^{a}\mathrm{I}_k^F \tag{38}
$$

Under π_a and π_b the same sessions empty their backlog in $(0, {}^{b}\mathrm{b}_k]$ (implying that ${}^{b}\hat{\mathrm{C}}({}^{b}\mathrm{b}_k^+) = {}^{a}\hat{\mathrm{C}}({}^{b}\mathrm{b}_k^+)$) and sessions in ${}^{a}\mathrm{I}_k^F$ receive, by assumption, the same amount of work over $(0, {}^{b}\mathrm{b}_k]$ under both policies. In conjunction with equation (38) it can be concluded that sessions in ${}^{a}\mathrm{I}_k^F$ preserve their position in $t-$ space under π_b. It is easy to show that this holds for the best effort traffic too.

CASE B: ${}^{a}\mathrm{b}_{k+1} = \infty$. This case is examined only for step (II.1.a) (${}^{b}\mathrm{I}_k = {}^{a}\mathrm{I}_k \cup {}^{a}\mathrm{I}_{k+1}$ and ${}^{b}\mathrm{b}_k > {}^{a}\mathrm{b}_k$), since the proof for step (II.1.b) is exactly the same as in **CASE A**.

–Equation (35) holds for CASE B too

In **CASE B** (and assuming that step (II.1.a) was followed) the following equations hold under π_b and π_a, since ${}^b\mathrm{b}_k = \infty$ and ${}^a\mathrm{b}_{k+1} = \infty$.

$$ {}^b\phi_i \ {}^b_b\mathcal{C}^+_{k-1} = \rho_i \quad i \in {}^a\mathrm{I}_k \cup {}^a\mathrm{I}^F_k \tag{39}$$

$$ {}^a\phi_i \ {}^a_a\mathcal{C}^+_{k} = \rho_i \quad i \in {}^a\mathrm{I}^F_k \tag{40}$$

$$\text{From (21), (22):}\ {}^a\mathrm{b}_{k-1} = {}^b\mathrm{b}_{k-1} \ \wedge \ {}^a_a\mathcal{C}^+_{k-1} = {}^b_b\mathcal{C}^+_{k-1} \tag{41}$$

Subtracting (39) from (40) implies:

$$ {}^a\phi_i \left({}^a_a\mathcal{C}^+_k - {}^a_a\mathcal{C}^+_{k-1} \right) = {}^b\mathrm{f}_i \ {}^a_a\mathcal{C}^+_{k-1} \quad i \in {}^a\mathrm{I}^F_k \tag{42}$$

Summing over all $i \in {}^a\mathrm{I}_k$ equation (39) gives:

$$ {}^a_b\mathcal{F}_k \ {}^a_a\mathcal{C}^+_{k-1} = {}^a_a\mathcal{P}_k \ {}^a_a\mathcal{C}^+_{k-1} - {}^a\mathcal{R}_k \tag{43}$$

Dividing (42) by (43), due to (34), results in (35).

$-\pi_b$ is acceptable

It has been shown that the "conditional exchange of weights" between sessions in ${}^a\mathrm{I}_k$ and sessions in ${}^a\mathrm{I}^F_k$ is feasible and that π_b is more efficient than π_a. It is easily seen that the requirements of the QoS sensitive sessions are not violated under π_b (that is π_b is acceptable). In particular :

- For $t < {}^a\mathrm{b}_k$: The NBSA bandwidth of the system is the same under the two policies (see equation (22)). Sessions in ${}^a\mathrm{I}^P_k$ are assigned the same weight under π_b and π_a so they get the same service under both policies, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) = {}^a\hat{\mathrm{W}}_i(0,t)$, $\forall i \in {}^a\mathrm{I}^P_k$. Sessions in ${}^a\mathrm{I}^F_k$ are assigned a greater weight under π_b than under π_a so they get better service under π_b, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) > {}^a\hat{\mathrm{W}}_i(0,t)$, $\forall i \in {}^a\mathrm{I}^F_k$. Sessions in ${}^a\mathrm{I}_k$ are assigned a smaller weight under π_b than under π_a so they get worse service, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) < {}^a\hat{\mathrm{W}}_i(0,t)$ $\forall i \in {}^a\mathrm{I}_k$, but by assumption they do not violate their delay bound.
- For ${}^a\mathrm{b}_k < t < {}^b\mathrm{b}_k$: All sessions in ${}^a\mathrm{I}^P_k$ have cleared their backlog under π_a and π_b so they get the same service under both policies, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) = {}^a\hat{\mathrm{W}}_i(0,t)$, $\forall i \in {}^a\mathrm{I}^P_k$. Sessions in ${}^a\mathrm{I}_k$ are assigned a smaller weight under π_b than under π_a so they get worse service, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) < {}^a\hat{\mathrm{W}}_i(0,t)$ $\forall i \in {}^a\mathrm{I}_k$, but by assumption they do not violate their delay bound. Sessions in ${}^a\mathrm{I}^F_k$ are assigned a greater amount of work in $(0, {}^a\mathrm{b}_k]$ $({}^b\hat{\mathrm{W}}_i(0, {}^a\mathrm{b}_k) > {}^a\hat{\mathrm{W}}_i(0, {}^a\mathrm{b}_k))$ and at ${}^b\mathrm{b}_k$ the amount of work they have received is the same under both policies $({}^b\hat{\mathrm{W}}_i(0, {}^b\mathrm{b}_k) = {}^a\hat{\mathrm{W}}_i(0, {}^b\mathrm{b}_k))$. So ${}^b\hat{\mathrm{W}}_i(0,t) > {}^a\hat{\mathrm{W}}_i(0,t)$ holds $\forall t \in ({}^a\mathrm{b}_k, {}^b\mathrm{b}_k)$, $\forall i \in {}^a\mathrm{I}^F_k$.
- For ${}^b\mathrm{b}_k < t$: All sessions in ${}^a\mathrm{I}^P_k \cup {}^a\mathrm{I}_k$ have cleared their backlog under π_a and π_b so they get the same service under both policies, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) = {}^a\hat{\mathrm{W}}_i(0,t)$, $\forall i \in {}^a\mathrm{I}^P_k \cup {}^a\mathrm{I}_k$. The NBSA bandwidth of the system is the same under the two policies at ${}^b\mathrm{b}_k{}^+$. This implies, in conjunction with equation (38), that sessions in ${}^a\mathrm{I}^F_k$ are assigned the same amount of work under π_b and π_a, i.e. ${}^b\hat{\mathrm{W}}_i(0,t) = {}^a\hat{\mathrm{W}}_i(0,t)$, $\forall i \in {}^a\mathrm{I}^F_k$.

Finally it is noted that the best effort traffic is assigned a greater weight under π_b and the same conclusions as for the sessions in ${}^a\mathrm{I}^F_k$ hold for it.

Note: Since the exchange of ϕ 's takes place in a continuous way, it is sufficient to show that the final policy is acceptable in order to ensure that every intermediate policy is acceptable. ${}^b\mathrm{b}_k$ can be replaced by any t ${}^a\mathrm{b}_k > t > {}^b\mathrm{b}_k$ and the proof that the intermediate policy is acceptable and more efficient still holds.

Proof of Proposition 2 (Section III-A)

According to Proposition 1 each time that the XMF process modifies the original policy the resulting policy (π_b) is acceptable and more efficient. This implies that the generated by the XMF process policy is acceptable and more efficient than the original policy, except from the case where XMF does not modify the original policy.

Two policies are identical (all sessions are assigned the same weights under both policies) if and only if the backlog clearing time of each session in an all greedy system is the same under both policies. In particular, in the all greedy system ${}^a\mathrm{b}_0 = 0$ and ${}^a\hat{\mathrm{C}}({}^a\mathrm{b}_0^+) = C_G$ hold under any policy π_a. If ${}^b\mathrm{b}_1 = {}^a\mathrm{b}_1 = \mathrm{b}_1$ for two policies π_a and π_b, it is concluded that ${}^b\phi_i = {}^a\phi_i$ for all sessions i for which $e_i = \mathrm{b}_1$ (where e_i is defined as ${}^ae_i = {}^be_i = e_i$ for all sessions i for which ${}^ae_i = {}^be_i$). If the same set of sessions empty their backlog at b_1 under both policies ${}^a\hat{\mathrm{C}}(\mathrm{b}_1^+) = {}^b\hat{\mathrm{C}}(\mathrm{b}_1^+)$ (the NBSA bandwidth has the same value right of the first backlog clearing time under both policies). If ${}^a\mathrm{b}_2 = {}^a\mathrm{b}_2 = \mathrm{b}_2$, it is concluded that ${}^b\phi_i = {}^a\phi_i$ for all sessions i for which $e_i = \mathrm{b}_2$. If the same set of sessions empty their backlog at b_2 under both policies ${}^a\hat{\mathrm{C}}(\mathrm{b}_2^+) = {}^b\hat{\mathrm{C}}(\mathrm{b}_2^+)$ holds. Making similar thoughts for all backlog clearing times the validity of the claim becomes clear.

Now suppose that there were two policies, π_{a1} and π_{a2}, for which $XMF(\pi_{a1}) = \pi_{o1} \neq \pi_{o2} = XMF(\pi_{a2})$. Let ${}^{o1}\mathrm{I}_m$, ${}^{o2}\mathrm{I}_m$ denote the set of sessions that empty their backlog m^{th} in oder under π_{o1} and π_{o2}, respectively. Suppose further that the two policies coincide for the set of sessions in ${}^{o1}\mathrm{I}_n = {}^{o2}\mathrm{I}_n$, $n = 0, 1, \ldots, k-1$ and their first difference occurs for the sets ${}^{o1}\mathrm{I}_k$ ${}^{o2}\mathrm{I}_k, k \geq 1$, i.e. the first backlog clearing time on which the two policies differ is considered. The differentiation between the two policies can have one of the following forms:

- ${}^{o1}\mathrm{I}_k = {}^{o2}\mathrm{I}_k$, that is under both policies the same sessions empty their backlog k^{th} in order but they differ on the backlog clearing time of these sessions. Suppose ${}^{o1}\mathrm{b}_k < {}^{o2}\mathrm{b}_k$. This means that under π_{o1} sessions in ${}^{o1}\mathrm{I}_k$ are not compressible in ϕ- space (else the XMF process would have expanded their busy periods) and under π_{o2} they are (contradiction).
- ${}^{o1}\mathrm{I}_k \neq {}^{o2}\mathrm{I}_k$. It is easily seen that a session which empties its backlog at a different time instance under the two policies exists. Such a session is not compressible in ϕ-space under the one policy (the policy under which it empties its backlog first) and is compressible under the other (contradiction).