

Deterministic End-to-End Delay Bounds in an Accumulation Network

Guillaume Urvoy¹, Gérard Hébuterne¹, and Yves Dallery²

¹ Institut National des Télécommunications,
Département RST, 9, rue Charles, Fourier 91011 Evry, France

² Laboratoire d'Informatique de Paris 6,
8, rue Capitaine Scott, 75015 Paris, France

Abstract. In this paper, we focus on the determination of end-to-end delay bounds for FIFO accumulation networks with leaky bucket constrained sources, including IP and ATM network cases. We determine an upper bound for the overall end-to-end delay and prove its accuracy to approximate the exact maximum end-to-end delay for accumulation networks of any size. This is achieved through an original trajectory analysis approach. Numerical studies further illustrate this point. This extends our previous results for the two-server and proves that this bound may be used as an accurate criterion for a CAC algorithm providing a deterministic QoS guaranteed service.

1 Introduction

The design and operation of multi-service networks providing Quality of Service (QoS) is a challenge in which Call Admission Control (CAC) algorithms are the key issue for non-adaptive traffic sources. Invoking the CAC procedure consists mainly in two basic procedures : determining a bound (on loss rate or delay for instance) on the path of the new source and checking the non-violation of the QoS of already established connections.

In this paper, we focus on the determination of deterministic end-to-end delay bounds in a FIFO accumulation network. Note that by doing so, we do not address the entire CAC problem, as the latter may not be reduced to the determination of some bounds. Most of the studies avoid this problem because they model the network through a single server (refer to the algorithms presented in [9,8]) whereas CAC is an end-to-end issue. An accumulation or concentration network is a tree network where sources may enter at any node but exit at the root node only. Though not the case in general, this topology is not only of theoretical interest. For instance, a Multipoint-to-Point Tree (MPT) is considered in [1] for the routing problem of IP over ATM networks.

We assume a fluid flow model and leaky bucket constrained sources. This model perfectly fits ATM networks (for VBR sources) due to the small cell size compared to the servers' rates. The results we obtain in the present work may also be applied to IP networks (for sources declared via a TSpec [4]) with some more

caution due to the variable size of the packets (see for instance [7] for the translation between a fluid and a packet model).

The remainder of this paper is organized as follows. In Sect. 2, we review the related work in the field of deterministic analysis. In this context, Network Calculus [5,2] provides a useful approach, based on service curves, to obtain deterministic bounds. We evaluate this technique for the present problem in Sect. 3. In Sect. 4, we present and discuss an upper bound on the end-to-end delay. The related additive property is studied in Sect. 5. In Sect. 6 and 7, we generalize the results of [10] obtained in the two-server case. In Sect. 8, we further generalize these results to cover the case of general accumulation networks. Conclusion and hints for future works are eventually given in Sect. 9.

2 Related Work

The FIFO discipline is not able to offer a differentiated service. This is why deterministic studies have focused mainly on the design and evaluation of new service disciplines such as Packet Generalized Processor Sharing [6] which provides end-to-end delay bounds for leaky bucket sources.

Cruz [5] provides results concerning the burstiness characterization of flows inside a FIFO network. However, the bounds derived are not tight enough since they are obtained through summation of local worst-cases. The service curve and network service curve paradigms (see Sect. 3) enable us to obtain tighter bounds. A recent and major work in the field of FIFO networks is [3]. The authors show that if the peak rates of sources in a general FIFO network are constrained by a certain value (related to the number of flows that the source meets on its path), then the network is stable and bounds on end-to-end delays and backlogs are obtained. This is a major result since it applies to FIFO networks with a general topology, but with strictly deterministic sources. By contrast, we focus, in this work, on variable bit rate sources but for the more restrictive class of accumulation networks.

3 A Service Curve Approach

Network Calculus provides a straightforward way to model sources and network elements, through an arrival and a service curve respectively. It formulates theorems to derive bounds on backlog and delays. The arrival curve of a source represents, intuitively, an upper bound on the amount of traffic the source can send on any time interval. The service curve represents a lower bound on the service the source may expect.

A given flow crossing n network elements offering $(\beta_i)_{i \in \{1,n\}}$ as service curves, may consider the network as a unique element with a (network) service curve which is the convolution of the individual service curves. The interest of the network service curve is that it provides a tighter bound on the end-to-end delay than summing the local bound that may be obtained at each server. To apply

this technique in the case of FIFO networks, a first step is to derive the service curve received by a given flow sharing a FIFO server with another flow, as illustrated next.

3.1 Service Curve in a FIFO Environment

Consider 2 sources, S_1 and S_2 (constrained by α_1 and α_2) and one FIFO server of capacity C . Let λ_C be the function such that : $\forall t \geq 0$, $\lambda_C(t) = Ct$, and R_i (resp. R_i^*) the cumulative rate function of S_i at the input (resp. output) of the server. For a given time t , let us denote s_0 the last time there was no backlog ($s_0 \leq t$). Thus, $R_1^*(s_0) = R_1(s_0)$ and $R_2^*(s_0) = R_2(s_0)$. Since the server is work conserving, this yields :

$$R_1^*(t) - R_1^*(s_0) + R_2^*(t) - R_2^*(s_0) = C(t - s_0) . \quad (1)$$

Causality also implies that $\forall t$, $R_2^*(t) \leq R_2(t)$. Thus :

$$R_2^*(t) - R_2^*(s_0) \leq R_2(t) - R_2(s_0) . \quad (2)$$

Now, since S_2 is constrained by α_2 and the server rate, we obtain :

$$R_2^*(t) - R_2^*(s_0) \leq \min(C(t - s_0), \alpha_2(t - s_0)) . \quad (3)$$

Mixing equation (1) and (3), we obtain :

$$R_1^*(t) - R_1(s_0) \geq C(t - s_0) - \min(C(t - s_0), \alpha_2(t - s_0)) . \quad (4)$$

Let us define $(x)^+$ as $\max(0, x)$. A service curve for S_1 is thus $\beta_1 = (\lambda_C - \alpha_2)^+$.

3.2 Discussion

The service curve obtained is thus conservative. Indeed, if S_2 were preemptive over S_1 , the service curve would be the same since, in this case, S_1 receives only the remaining capacity unused by S_2 . Besides, assume that S_1 and S_2 transit in a second server where they mix with a third source. To derive a service curve for S_1 in the second server, an arrival curve for S_2 at the input of the second server is required. Network Calculus provides a way to derive this arrival curve from the arrival curve of S_2 at the input of server 1 and its service curve in server 1. But since the service curve for S_2 at server 1 is pessimistic, the arrival curve for S_2 at the second server will also be pessimistic. Thus, the conservative aspect of the result increases with the size of the network. This approach leads inevitably to pessimistic results. For instance, consider a single server and assume S_1 and S_2 have the same traffic descriptor, namely (p, R, M) , where p is the peak rate of the source, R the leak rate of the bucket and M its depth. Then, the following relation exists between the bound on delay D_{SC} obtained with the service curve approach and the exact value of the maximum delay D_{max} : $D_{max} = \frac{C-R}{C} D_{SC}$. Thus, when $R \rightarrow \frac{C}{2}$ (stability requires that $C > 2R$), $D_{SC} \rightarrow 2D_{max}$.

The drawbacks of this method lead us to envisage a new approach presented in the next section. Note, however, that we have not proved that it was not possible to find a better service curve. It remains an open problem.

4 The Additive Bound

Consider first a tandem accumulation network with p servers and n sources. A superposition of leaky bucket constrained sources may be seen as a single multi-leaky bucket constrained source with arrival curve being the sum of the individual arrival curves. We can thus group the sources entering the network at a given node and consider a network with p servers and p sources.

In [10], we show that if the input of a server is leaky bucket constrained, so is the output. When applied to node j of the considered accumulation networks, we obtain that the source seen at the input of this node is multi-leaky bucket constrained. Application of Network Calculus gives that the worst-case source at node j (which generates the maximum local delay) is obtained when the multi-leaky bucket source is greedy (a leaky bucket constrained source is greedy when it emits its bits as soon as possible : it thus first emits its maximum burst size at its peak rate p and then emits at its mean rate R). A recursion from server j to server 1 proves that if all the sources crossing node j are greedy and synchronous, i.e. they begin their emission at the same time, then the aggregated source at the input of node j is the worst-case source. We thus obtain the local maximum delay at node $k \in [1, j]$ is achieved when all the sources entering the network before at node $k \in [1, j]$ are greedy and synchronous. The sum of these maximum local delays provides an upper-bound on the maximum end-to-end delay in the system. We call it the Additive Bound. We have only considered tandem networks so far but the results also hold for a tree network : the maximum local delays are obtained when all the sources entering the network are greedy and synchronous. Since the aggregated source seen at the output of a subtree of the accumulation network is multi-leaky bucket constrained, we only have to test the accuracy of the Additive Bound for tandem accumulation networks.

5 Additivity Property and Networks Classification

5.1 Definition

We term Additivity the following property : “There exists a trajectory of the system such that a bit experiences an end-to-end delay equal to the Additive Bound”. By extension, we say that a network is additive if it exhibits the Additivity property. It is non-additive otherwise.

5.2 Intrinsic Parameters (t_{max}, d_{max}):

Let us denote greedy trajectory, the trajectory of the system where all the sources are greedy and synchronous. This yields the maximum delay on each node (see

Sect. 4). For this greedy trajectory and for each node j , we use the following parameters : $d_{j \max}$, the maximum local delay and $t_{j \max}$, the time where the bit experiencing $d_{j \max}$ arrives at node j (see Fig. 1).

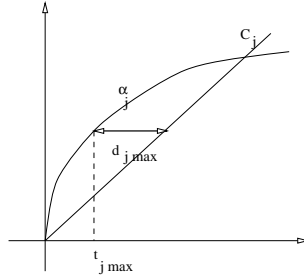


Fig. 1. Intrinsic Parameters

5.3 Networks Typology

In [10], we proved that a two-stage network is additive if and only if $t_{2 \max} \leq t_{1 \max} + d_{1 \max}$. This condition can be verified easily with the greedy trajectory of the system. We also characterize the “additive trajectory” : S_1 and S_2 greedy respectively from times $\theta_1 = 0$ and $\theta_2 = t_{1 \max} + d_{1 \max} - t_{2 \max}$.

By extension, any general tandem accumulation network can be partitioned in a set of subnetworks for which the following property either holds or not :

Property 1. For all adjacent servers j and $j+1$, $t_{(j+1) \max} \leq t_{j \max} + d_{j \max}$.

6 Additive Networks

We generalize here the result obtained for the two-server case. Consider an accumulation network with p servers for which Prop. 1 holds. Let us define $(\theta_j)_{j \in \{1, p\}}$ as follow :

1. $\theta_1 = 0$
2. $\theta_{j+1} = \theta_j + (t_{j \max} + d_{j \max} - t_{(j+1) \max})$, $j \in [1, p-1]$

If S_j is greedy from $t = \theta_j$, (note that $\theta_{j+1} \geq \theta_j$), the bit experiencing $d_{1 \max}$ in the first server experiences $d_{j \max}$ at node j for all $j \in \{1, p\}$. Thus its end-to-end delay is : $D_{\max} = \sum_{j=1}^p d_{j \max}$. An accumulation network for which Prop. 1 holds is thus additive. Besides, since the only way for a bit to experience $\sum_{j=1}^p d_{j \max}$ is to experience $d_{j \max}$ at server j , for all $j \in \{1, p\}$, it follows that a network that does not fulfill Property 1 is not additive.

7 Non-additive Networks

In this section, we generalize the lower bound building method developed for a two-server network to accumulation networks of any size and use it to test the accuracy of the Additive Bound in the case of non-additive networks. A direct generalization would hide the difficulty of the construction of the trajectory. We thus first present the three-server case.

7.1 Three-Server Case

Lower Bound. Consider a two-stage network. If it is non-additive, this means intuitively that the burst necessary to obtain $d_{1 \max}$ is not sufficient to obtain $d_{2 \max}$ in the second server (considering the greedy trajectory of the system), since when all the bits of this burst have reached the second server, the delay that can be obtained is less than $d_{2 \max}$. The idea is then to postpone the emission of this burst so as to synchronize the local maximum delays, knowing that the delay on the second server will necessarily be less than the delay in the greedy synchronous case.

Consider now a network with three servers and three sources $(S_i)_{i \in \{1, \dots, 3\}}$ (S_i entering at node i). The trajectories of the sources are built so as to maximize the amount of bits in buffer j when the reference bit, i.e. the one experiencing $d_{1 \max}$ in the first server, arrives.

Trajectory of Sources. Consider the greedy trajectory of S_1 , as given in Fig. 2. It can be divided into three parts. The first part corresponds to the part of the trajectory necessary to achieve the local maximum delay $d_{1 \max}$. The second part corresponds to the time necessary for the last bucket of the sources composing S_1 to empty. In the last part, all the sources composing S_1 have reached their mean rate.

Now consider the trajectory of S_1 given in Fig. 3. S_1 is an aggregation of n_1 sources, with each source controlled by a single leaky bucket with parameters (p_k, R_k, M_k) for $k \in \{1, \dots, n_1\}$. With the greedy trajectory of the system, the source with index k emits at its peak rate p_k during $[0, \frac{M_k}{p_k}]$ and then emits at its mean rate R_k . Let us define :

$$T_{\frac{M}{p}}^{S_1} = \max_{k \in \{1, \dots, n_{S_1}\}} \left(\frac{M_k}{p_k} \right) . \quad (5)$$

$T_{\frac{M}{p}}^{S_1}$ corresponds to the beginning of the third part defined in Fig. 2. The modified trajectory is built by changing the beginning of emission of the sources composing S_1 as follows :

1. if $\frac{M_k}{p_k} \leq t_{1 \max}$ then the source :
 - (a) emits at its mean rate during $[0, T_{\frac{M}{p}}^{S_1} - t_{1 \max}]$,
 - (b) becomes greedy for $t \geq T_{\frac{M}{p}}^{S_1} - t_{1 \max}$; this is possible since its bucket is still full at this time.

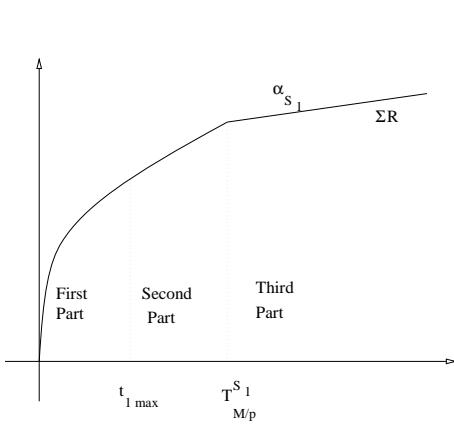


Fig. 2. S_1 initial trajectory

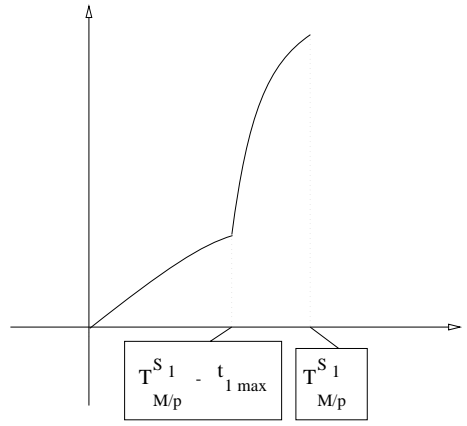


Fig. 3. S_1 modified trajectory

2. if $\frac{M_k}{p_k} \geq t_{l \max}$, then the source :
 - (a) emits at its mean rate during $[0, T_{M/p}^{S_1} - \frac{M_k}{p_k}]$,
 - (b) becomes greedy for $t \geq T_{M/p}^{S_1} - \frac{M_k}{p_k}$.

The modified trajectory has two parts (see Fig. 3) :

1. the first part where some sources emit at their peak rate whereas others emit at their mean rate. This part corresponds to the second part of the initial greedy trajectory with a slight modification : if a source emits at its peak rate during τ_1 and then at its mean rate during τ_2 in the initial trajectory, then, in the modified trajectory, it first emits at its mean rate during τ_2 and then at its peak rate during τ_1 . Due to this inversion between τ_1 and τ_2 , we call inverted part this part of modified trajectory.
2. the second part is strictly equivalent to the first part of the initial trajectory.

Note that, as with the initial trajectory, the last bucket empties at time $t = T_{M/p}^{S_1}$. A modified trajectory for S_2 and S_3 is built using the same method. We now fix the synchronization parameters.

Synchronization of Sources. With the modified trajectory given above for S_1 , the last bit of the burst (reference bit) experiences a delay $d_{1 \max}$. S_2 is synchronized in such a way that the end of its burst corresponds to the arrival of the reference bit. This bit will then experience $d_2 \leq d_{2 \max}$ in the second server. Since, a priori, $T_{M/p}^{S_1} \neq T_{M/p}^{S_2}$, using the previous synchronization method leads one of the sources to start emitting before the other. Assume S_2 starts its emission before S_1 . To maximize the number of bits backlogged in server 2 at the time where the reference bit arrives, it is possible to modify the trajectory

of S_1 such that it emits at its mean rate before the beginning of the modified trajectory, in an interval of length $T_{\frac{M}{p}}^{S_2} - T_{\frac{M}{p}}^{S_1}$. This trajectory of S_1 is valid with respect to its leaky bucket constraint. The same method is applied to synchronize S_3 , as shown in Fig. 4.

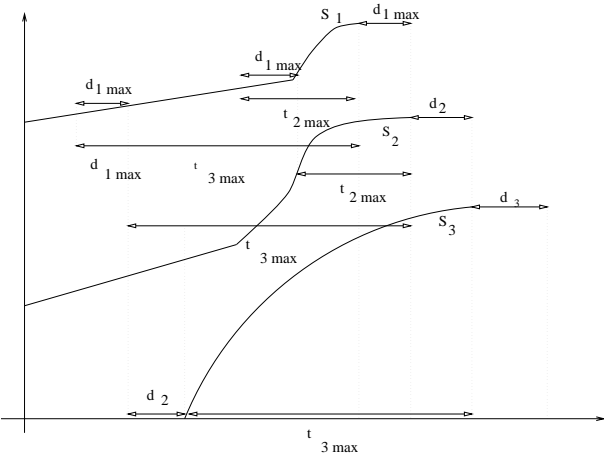


Fig. 4. Sources synchronization

Result for Delay. The lower bound on the maximum end-to-end delay is obtained as the end-to-end delay of the reference bit in the modified trajectory. Since all the sources are leaky bucket constrained, the initial and modified trajectories described above correspond to piece-wise linear curves. Computation of the intrinsic parameters as well as the delay of the reference bit is thus straightforward from the algorithmic point of view.

7.2 Numerical Results

Our aim is to estimate the accuracy of the Additive Bound in a non-additive network with the lower bound presented above. Accuracy means here the relative difference between the Additive Bound and this lower bound. A key problem is the choice of the non-additive networks.

Networks Building Method.

We consider accumulation networks with $p = \{4, 5, 8, 10\}$ servers. For each server, we draw the number of sources entering at this stage in a uniform fashion in the interval $[1, 5]$. Then, we draw the characteristics of the sources in Table 7.2. We now have to choose the capacities of the servers. A necessary condition

Table 1. Sources descriptors

<i>Peak rate p</i>	<i>Mean rate M</i>	<i>Burstiness M</i>
10	0.1	10
100	1	100
1000	10	1000

for a network to be additive is that the rate of the servers increases. Conversely, if capacities decrease, the network is non-additive (sufficient but not necessary). We choose to give every server the same rate which is the sum of the mean rate of all the sources times $\gamma = 1.01$ (γ is used to ensure stability). This sum represents the minimum capacity of the last server in the case of accumulation networks . Doing so, the most important part of the end-to-end delay is concentrated at the end of the network. To obtain some significant results, we choose to calculate the relative range, which is the difference between the lower bound and the Additive Bound divided by the Additive Bound, for this initial system, i.e. a particular random generation of the sources descriptors and capacities of servers. We next modify the network by enforcing some of the sources to modify their entering node. This is achieved through the following algorithm :

1. Step 1 : one computes the relative range for the initial network.
2. Step 2-9 : each source is removed from node j to node $j - 1$ with probability 0.1.

Applying this algorithm, the accumulation network heuristically “worsens” and thus the relative range should increase.

Results. The results, presented in Fig. 5, are obtained for 10000 successive random generations of networks. The x-axis is indexed following the steps of the algorithm. For each step and for the different network sizes, we compute the mean relative range.

Discussion. For non-additive accumulation networks, we have an upper bound on the end-to-end delay, the Additive Bound, and a heuristically obtained lower bound. The maximum, exact, end-to-end delay over all possible trajectories of the system is thus between these two bounds and gives full meaning for considering the relative range as a performance parameter.

The results obtained confirm the good accuracy of the lower bound. The mean relative ranges remain reasonable even for large size of networks. The maximum error, not presented here is no more than 67%. It thus remains within the same order of magnitude. We now address the case of general accumulation networks which are neither strictly additive nor strictly non-additive.

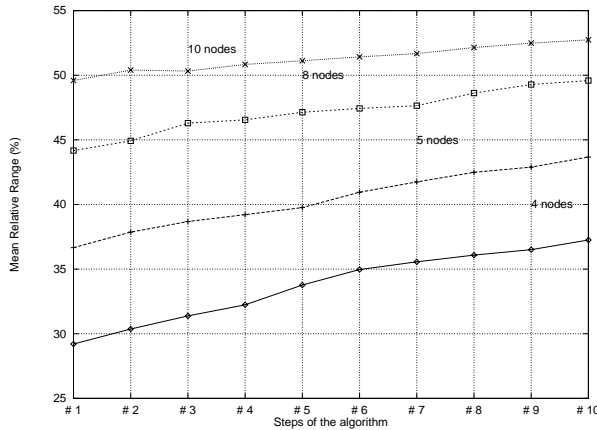


Fig. 5. Results with network of different sizes

8 General Accumulation Networks

Our expectation is that the Additive Bound always represents an accurate upper bound on the maximum end-to-end delay for accumulation networks. Proving such a statement requires an exhaustive study, which is not possible. We restrict our study to a specific class of accumulation networks, that we term well-formed accumulation networks. A well-formed accumulation network is an accumulation network where the following rule applies : capacities of the servers follow an increasing fashion from the leaves to the root of the tree.

We extend here the previous results for well-formed tandem accumulation networks using the same lower bound as in the non-additive case. Indeed, the method used to build the trajectory leading to the lower bound is based solely on the set of intrinsic parameters $(t_{j \max}, d_{j \max})$ and do not rely on any assumption concerning the additivity of the network. It may thus be applied in the case of well-formed accumulation networks.

8.1 Networks Building Method

The method used to generate a well-formed network is the following :

1. for each server, the number of sources (between 1 and 5) entering at this node and their characteristics are drawn (see Table 7.2).
2. the rate of each server is then computed as the sum of the mean rate of the sources crossing this node times a coefficient α . α can take one of the three following values $\{1.1, 1.5, 2.0\}$ which, for each set of sources, leads to three different networks.

8.2 Results

We present hereafter the numerical results obtained for networks of various sizes (from 3 to 20 servers). For each network size, 10000 networks are drawn. The performance parameter computed for each network is the relative range between the lower bound and the Additive Bound. The average relative ranges are presented in Table 8.2.

Table 2. Average relative ranges (in %) with networks of different sizes

	<i>Size=3</i>	<i>Size=5</i>	<i>Size=10</i>	<i>Size=15</i>	<i>Size=20</i>
$\alpha = 1.1$	0.72	1.29	2.03	2.89	3.87
$\alpha = 1.5$	1.97	3.36	5.67	8.42	11.23
$\alpha = 2.0$	1.96	3.13	5.06	7.64	10.18

8.3 Discussion

The results shown in Table 8.2 strongly confirm our claim : the Additive Bound represents a good criterion for a deterministic CAC (a CAC based on a deterministic delay bound). They are also interesting since the way well-formed accumulation networks are built here is close to a real dimensioning process. Indeed, α^{-1} , which is the rate of the server divided by the sum of the average rates of the sources it serves, represents the average activity rate of the servers and tuning activity rates of servers at a given rate is a common procedure for networks dimensioning.

Compared to the results obtained in the previous section, the relative ranges obtained here are significantly smaller : for instance, for a network with 10 servers, the relative range was close to 50% whereas here it is close to 5%. This is due to the method the server rates are assigned in each case. In the previous section, all the servers had the same capacity which lead to a strictly non-additive network, whereas here, the rates increase from one server to another, which is a necessary (though not sufficient) condition to obtain additive networks.

9 Conclusion

In this paper, we focused on the problem of determining an end-to-end delay bound in an accumulation network. We first evaluate the direct service curve approach provided by the Network Calculus theory and stress the over-conservatism of the obtained results. We then propose a new approach, based on a trajectory analysis and the Additivity property introduced for a two-server case in [10]. We show that the Additive Bound (sum of the local maximum delays) represents a good approximation for the maximum end-to-end delay.

Future work will address the problem of designing a CAC procedure based on the Additive Bound. Preliminary results underline the problem of checking the non-violation of the QoS of the already established connections in a network environment. It seems that a heavy updating procedure needs to be performed after each new source acceptance. Extension of the network topologies is also an important point to consider. Up to now, we have addressed accumulation networks. We now wish to investigate the case of general networks topologies. However, the Additive Bound may be less accurate due to the fact that sources mixing with the reference source may leave this source before its exiting node. Preliminary results indicate that the computation of the local maximum delay is a very difficult task in a general FIFO network with leaky bucket constrained sources.

References

1. H. Ahmed, R. Callon, A. Malis, and J. Moy. Ip switching for scalable ip services. *Proceedings of the IEEE*, 85(12), 1997.
2. J.-Y. Le Boudec. An application of network calculus to guaranteed service networks. *IEEE/ACM Transactions on Information Theory*, May 1998.
3. I. Chlamtac, A. Faragó, and H. Zhang. A deterministic approach to the end-to-end analysis of packet flows in connection-oriented networks. *IEEE Trans. on Networking*, August 1998.
4. D. D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network : Architecture and mechanism. *ACM SIGCOM'92*, 1992.
5. R.L. Cruz. A calculus for network delay, part 1 : network elements in isolation. *IEEE Transaction on Information Theory*, January 1991.
6. A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks : The single-node case. *IEEE/ACM Transactions on Networking*, June 1993.
7. A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks : The multiple-node case. *IEEE/ACM Transactions on Networking*, Mars 1994.
8. H. G. Perros and K. H. Elsayed. Call admission control schemes : A review. *IEEE Magazine on Communications*, Nov. 1996.
9. K. Shiimoto, N. Yamanaka, and T. Takahashi. Overview of measurement-based connection admission control methods in atm networks. *IEEE Communications Surveys*, 1999.
10. G. Urvoy, G. Hébuterne, and Y. Dallery. Delay-constrained vbr sources in a network environment. In *ITC'16*, 1999.