Guaranteeing Deterministic QoS in a Multipoint-to-point Network

Guillaume Urvoy-Keller [a] and Gérard Hébuterne [b]

[a]Institut Eurecom,
2229, route des Crêtes, 06904 Sophia-Antipolis, France

[b]Institut National des Télécommunications,
9, rue Charles Fourier, 91011 Evry cedex, France

The need to guarantee QoS to multimedia sources leads to a convergence between the routing and forwarding functions in the Internet. MPLS provides a solution to this integration. In the context of MPLS, multipoint-to-point (m-t-p) architectures appear as key architectures since they provide a cheaper way to connect edge nodes than point-to-point connections, in an Internet Service Provider backbone. Th m-t-p architecture has been mainly studied for its ability to balance load. In the present work, we go a step further and evaluate two traffic management schemes that provide deterministic QoS guarantees for multimedia constrained sources in a m-t-p network. The first scheme is based on the FIFO policy and manages the multimedia sources as variable bit rate sources. The second scheme shapes the sources before they enter the network so as to manage them as constant bit rate sources in the network. We prove that the second scheme performs better in terms of scalability and resource requirements.

1. Introduction

Provisioning of Quality of Service (QoS) in high-speed networks has received much attention in the last decade. The ATM community advocated for a solution purely based on switching while the Internet community advocated for a routing solution. The current trend is a combination of these solutions as most routers in backbones comprise an underlying ATM switch. ATM switches provide an high-speed and low cost per port solution for the Internet. However, they are not universally deployed. MPLS, Multiprotocol Label Switching [10], has been developed to offer a universal forwarding layer to the Internet. MPLS may inter-operate adequately with ATM [9] or Frame Relay [5] or provide an ad-hoc forwarding service.
A major requirement for a QoS service is the ability to balance the load inside the network. In the traditional Internet, this is achieved with metric-based routing. The network administrator adjusts link metrics to achieve the required balancing. This ad-hoc solution is not satisfying in the context of QoS provisioning and there exists a need for a tighter control of the traffic flows. Explicit route-based control provides such a tight control. It may be achieved using MPLS. An Internet Service Provider (ISP) may use MPLS to establish a set of routes between its ingress nodes and its egress nodes. If point-to-point (p-t-p) routes are used and there are n edges, then $O(n^2)$ routes are required to connect the nodes. A less resource consuming way to cover the network is to use multipoint-to-point (m-t-p) connections rooted at the egress nodes. With m-t-p

connections, only $O(n)$ routes are required. Such a solution has been investigated in [1] in the context of IP over ATM and in [11] in the context of IP over MPLS. In [11], Saito et al. propose a traffic engineering scheme for Internet backbones, which aims at providing an optimal and reliable load balancing. Their proposed scheme uses multiple m-t-p Label Switch Paths (LSPs) between each ingress/egress pair to achieve load balancing and reliability. The traffic demand is expressed as service rates. Sources are thus implicitly assumed to be constant bit rate sources.
The multipoint-to-point architecture will become a typical architecture in the future Internet that combines routing and forwarding functions. In this paper, we propose and evaluate two traffic management schemes for a multipoint-to-point network that guarantee a deterministic QoS to variable bit rate sources. Sources are assumed to be leaky bucket constrained with a maximal end-to-end delay requirement. We assume a fluid model that closely approximates the behavior of a packet network with a small packet size compared to the service rate of the servers. It enables us to concentrate on the central issues. The first traffic management scheme is based on a deterministic multiplexing of the variable bit rate sources. It is called DMS for Deterministic Multiplexing Scheme. The second traffic management scheme shapes the sources before they enter the network. It is called DSS for Deterministic Shaping Scheme.
The remainder of the paper is organized as follows. In Section 2, we provide the results on which DMS relies, i.e. end-to-end delay bounds and a corresponding admission control procedure. In Section 3, we describe the two traffic management schemes, DMS and DSS. In Section 4, we compare DMS and DSS in terms of scalability of the admission control procedure and resource requirements. In Section 5, we conclude and provide some insights for future work.

2. Preliminaries on Multipoint-to-point Network

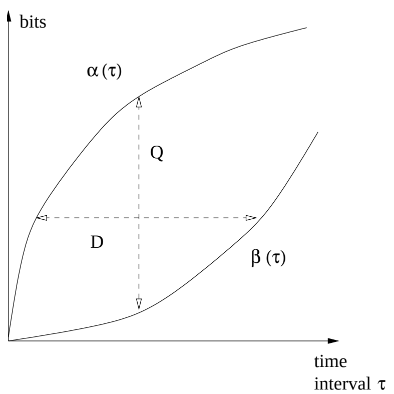
In the paper, we consider multipoint-to-point (m-t-p) networks where a source may enter the network at any node but exit at the root node only. Sources are assumed to be leaky bucket constrained. In [13], we provide an accurate upper bound on the end-to-end delay for m-t-p networks where the scheduling policy is FIFO. In [12], a corresponding admission procedure is derived. These results are obtained using the Network Calculus [7,8,2,4]. We first briefly recall the bases of Network Calculus and discuss the end-to-end delay bound and the corresponding admission control procedure.

2.1. Network Calculus

Network Calculus provides deterministic bounds on end-to-end delays and backlogs. A source is modeled through an arrival curve that represents an upper bound on the volume of traffic it can send during any interval of time. In the case of a leaky bucket constrained source S with parameters (p, R, M) an arrival curve α is (see [2]):

$$\alpha(\tau) = \min\left(p\tau, R\tau + M\frac{p-R}{p}\right),\ \tau \geq 0 \tag{1}$$

Servers are modeled through a service curve representing a lower bound on the service they are able to provide during some time intervals. For instance, a service curve β for a FIFO server with a service rate C is $\beta(\tau) = C \cdot \tau$ (the time intervals, here, can be the backlog periods). Consider now a system with an input flow characterized by an arrival curve α and a server with a service curve β. An upper bound D on delay (resp. Q on backlog) is given by the maximal



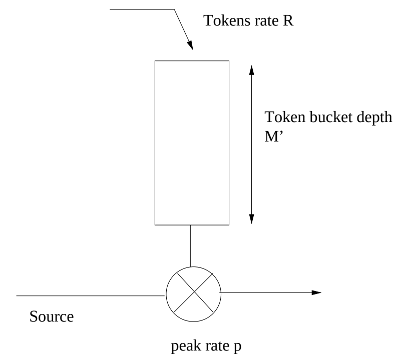


Figure 1. Upper bound on backlog and delay Figure 2. Leaky bucket controller

horizontal (resp. vertical) distance (see Figure 1) between the arrival curve and the service curve of the system (Theorems 1 and 2 of [6]). Knowledge of the arrival and service curves allows also to derive an arrival curve α^* for the flow once it has crossed the server. In the more general case, α^* may be expressed as a min-plus convolution between α and β (see [2]):

$$\alpha^*(\tau) = \sup_{v \geq 0}(\alpha(\tau + v) - \beta(v)) \tag{2}$$

In the more restrictive case of a FIFO (or work-conserving) server and a leaky bucket constrained input flow, a simpler characterization may be used:

$$\alpha^*(t) = \min(\alpha(\tau), \beta(\tau)) \tag{3}$$

Equation (3) indicates that a leaky bucket constrained flow crossing a work-conserving server is still leaky bucket constrained with the same leaky bucket parameters and an additional constraint on its peak rate.

2.2. End-to-end Delay Bound

Consider a given path $\mathcal{P}$ in a m-t-p network with K servers. Let $I = \{1, 2, \ldots, K\}$ be the set of indices of the servers in $\mathcal{P}$ (K is the root node). All the input sources are leaky bucket constrained. An arrival curve for a multiplex of leaky bucket constrained sources is the sum of each individual arrival (see [4]). Using this property and the result from equation (3), we prove in [13] that it is possible to derive an upper bound on the delay $\mathcal{D}_j$ at server $j \in J$. We further prove that these bounds, $(\mathcal{D}_j)_{j \in I}$, are in fact maximum delays through a trajectory analysis. The sum $\mathcal{D}_{\mathcal{P}} = \sum_{j \in I} \mathcal{D}_j$ of the these local maximum delays represents an upper bound on the end-to-end delay for path $\mathcal{P}$. The main result of [13] is that $\mathcal{D}_{\mathcal{P}}$ is an accurate approximation of the maximum end-to-end delay on path $\mathcal{P}$. The delay bound obtained is called the additive bound since it is obtained through summation of local maximum delays.

2.3. Admission Control Procedure

In [12], we derive two admission control algorithms based on the additive bound: a centralized one and a distributed one. The centralized algorithm works as follows (we assume n

sources are already in the network and we have to take a decision for source $n+1$ which follows path $\mathcal{P}$): (i) computation of the new value of the local maximum delay $\mathcal{D}_j$ for $j \in I$, (ii) checking of the non-violation of the QoS constraint (maximum delay) of all the sources that have at least one node in common with $n+1$.

In the distributed admission control procedure, each server needs to store the safety margin of the sources it serves. The safety margin of a source is the difference between the delay constraint of the source and the additive bound along the path of the source. The admission procedure proceeds in two phases. In a first phase, the additive bound $D_{\mathcal{P}}$ is computed and potential QoS violations are detected, using the safety margins. This iterative procedure starts from node 1 and moves down to node K, following the route $\mathcal{P}$ of source $n+1$. At the end of the first phase, the new value of $\mathcal{D}_{\mathcal{P}}$ is computed and we can check whether any QoS constraint is violated. We can thus make a decision for source $n+1$. However, the values of the safety margin stored by the servers along the route of a given source are no more consistent. This is why a second phase is required which consists in a flooding of the network with the new values of the safety margins of all the sources (they are all affected by $n+1$ since they share at least one server: the root node of the m-t-p network).

3. Traffic Engineering in a Multipoint-to-point Network

3.1. Traffic Demand

We consider sources that are leaky bucket constrained with an additional constraint on their peak rate. A traffic descriptor for a given source S comprises three parameters (p, R, M), which are the peak rate p, the mean rate R and the maximum burst size M of the source. Such a source is able to pass though the leaky bucket controller depicted in Figure 2 without experiencing any loss. Note that the size of the token bucket is $M' = M\frac{p}{p-R} > M$ since the peak rate of the source is finite.

The traffic management schemes that we discuss provide deterministic QoS guarantees. A given source S has thus one QoS constraint, which is its maximum end-to-end delay requirement.

3.2. Deterministic Multiplexing Scheme

The deterministic multiplexing scheme, DMS, is based on the additive bound presented in Section 2.2 and the associated distributed admission procedure (Section 2.3). Each server $j \in \{1, \ldots, K\}$ implements the FIFO scheduling policy and stores in a table the values of the safety margins of each source it serves. Each server j must also store the current value of the local maximum delay $\mathcal{D}_j$.

Another important issue is the buffer space requirement. Since we want to design a service that provides deterministic guarantees, the buffer space at each server must be large enough so that no loss occurs. Since $\mathcal{D}_j$ is achievable at server j (see Section 2.2), server j must grant the sources with a buffer space of at least $\mathcal{D}_j \cdot C_j$, where C_j is the service rate of server j.

3.3. Deterministic Shaping Scheme

3.3.1. Deterministic Effective Bandwidth

Central to DSS is the concept of *deterministic effective bandwidth* that has been first introduced by Le Boudec [2] in the context of the Network Calculus [6]. It makes use of the theorems by Cruz on delays and backlogs (see Section 2.1). The deterministic effective bandwidth $e_d(\alpha)$ associated to a given source S for a given arrival curve α and a delay constraint d is the minimum service rate that ensures to this source a delay smaller than d. The following result for $e_d(\alpha)$ is given in [2]:

$$e_d(\alpha) = \sup_{s \geq 0} \Big(\frac{\alpha(s)}{s+d} \Big) \tag{4}$$

In the special case of a leaky bucket constrained source S with an arrival curve given by equation (1) and a delay requirement d, equation (4) may be re-written as (see Figure 3):

$$e_d(\alpha) = \max \Big(R, \frac{M}{d + \frac{M}{p}} \Big) = \begin{cases} \frac{M}{d+\frac{M}{p}} & \text{if } 0 \leq d \leq M(\frac{1}{R} - \frac{1}{p}) \\ R & \text{if } d \geq M(\frac{1}{R} - \frac{1}{p}) \end{cases} \tag{5}$$

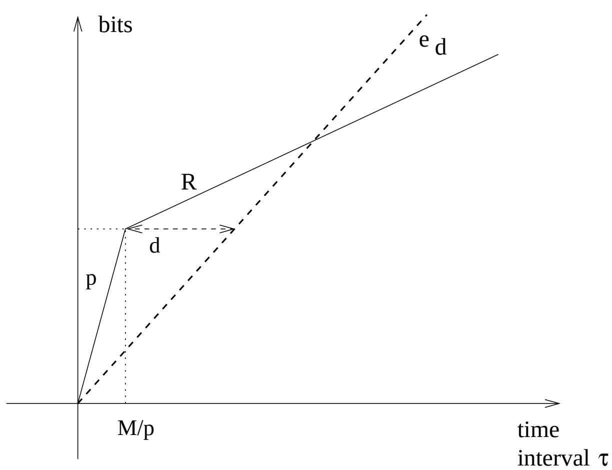


Figure 3. Deterministic effective bandwidth of a leaky bucket constrained source

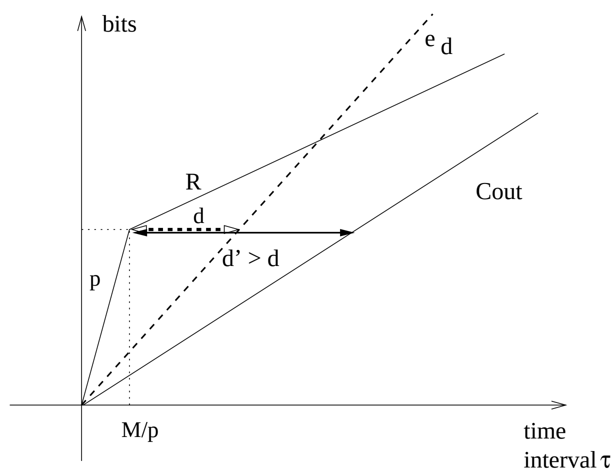


Figure 4. Choice of the shaping rate

Note that $R \leq e_d(\alpha) \leq p$.

3.3.2. QoS Provisioning Scheme

With DSS, sources are shaped, prior to their entrance in the network, at a rate equal to their deterministic effective bandwidth. This operation may be done using a spacer [3].

Theorem 1 *The choice of the deterministic effective bandwidth as a shaping rate is optimal in the sense that any smaller shaping rate can lead to a QoS violation.*

Proof:

Let us prove the result by contradiction. Suppose that we choose a shaping rate $C_{out} < e_d(\alpha)$. We show that there is at least one trajectory of the source such that one bit experiences a delay strictly larger than d. Consider the greedy trajectory of S, which is the trajectory where S consumes its tokens as soon as they are available. The cumulative arrival rate of this trajectory corresponds to α and the service rate offered by the spacer is $\beta(t) = C_{out} \cdot t$. As a consequence, the bit emitted at time $t = \frac{M}{p}$ experiments a delay d' strictly larger than d (see Figure 4). This proves the result. $\square$

Let us now focus on the behavior of the servers. Since, with DSS, a source may incur a delay equal to its delay constraint in the shaper, DSS must guarantee a delay equal to zero in the network (in the fluid model context) in order to provide a deterministic QoS service.To guarantee a delay equal to zero, DSS allocates to each source a service rate equal to its peak rate, i.e. its deterministic effective bandwidth, on each server along its route. This may be done using a dynamic Complete Partitioning (CP) scheduling policy. CP allows to partition the bandwidth of the server among different sources. Here, we consider two classes of sources: the QoS class comprising the sources with QoS constraints and the BE class comprising the best-effort sources. The QoS class will be granted a service rate C_j^{QoS} at server j and the best-effort class will be granted C_j^{BE}. We require $C_j = C_j^{QoS} + C_j^{BE}$, where C_j is the service rate of server j. Let $(S_i)_{i \in I_j}$ (resp. $(e_{d_i}(\alpha_i))_{i \in I_j}$) be the set of sources crossing server j (resp. their deterministic effective bandwidths). We set:

$$C_j^{QoS} = \sum_{i \in I} e_{d_i}(\alpha_i) \tag{6}$$

$$C_j^{BE} = C_j - \sum_{i \in I} e_{d_i}(\alpha_i) \tag{7}$$

Each source of the QoS class is thus granted a service rate equal to its deterministic effective bandwidth at each server j along its route. Thus, globally, with DSS, the network guarantees to each source of the QoS class a service rate equal to its deterministic effective bandwidth. This insures that the source experiences a delay equal to zero in the network.
Let us focus now on the buffer space requirement. Since the resource allocation is peak rate based, no buffer space needs to be allocated inside the nodes of the network. This is a major advantage as compared to DMS where buffer space should be allocated at each server.

3.3.3. Admission Control Procedure

The admission control algorithm is triggered upon the reception of two events: (i) if a new source is to be accepted, to reserve resources, (ii) when a connection ends, to free the resources attached to this connection. We study these two cases successively.

3.3.4. Admitting a New Source

For a given server j with a service rate C_j, let $C_j^{QoS} = \sum_{i \in I} e_{d_i}(\alpha_i)$ be the sum of the deterministic effective bandwidths of all the sources in the QoS class. Consider the admission process of a new source S with a deterministic effective bandwidth $e_d(\alpha)$. The admission algorithm consists in checking the following condition for every node j on the route of the source:

$$C_j^{QoS} + e_d(\alpha) \leq C_j \tag{8}$$

If the source request is to be accepted, then every node k along the route of the source has to update the value of the variable C_k^{QoS} from C_k^{QoS} to $C_k^{QoS} + e_d(\alpha)$.

3.3.5. Session Termination

Since the servers store only the total service rate allocated to the QoS class, the admission control algorithm must ensure that when a session characterized by $e_d(\alpha)$ ends, each server along the path of this source receives a message indicating the corresponding deterministic effective bandwidth $e_d(\alpha)$. Each server j must then update its parameter C_j^{QoS} from C_j^{QoS} to $C_j^{QoS} - e_d(\alpha)$.

4. Comparison of DMS and DSS

4.1. Admission Control Procedure

The main concern with the admission control procedure is the scalability. While this concept is widely used, there exists no precise definition of scalability. For us, *a service is scalable if the effort to provide this service does not depend on the number of admitted connections.*
Clearly, DMS, which requires each server to keep track of the safety margin of the sources it serves, suffers from a lack of scalability. This is emphasized by the m-t-p architecture which causes all previously accepted connections to be involved in the admission process of a new source (see Section 2.3). The second phase of the distributed algorithm results then in a flooding of the m-t-p network.
With DSS, the admission procedure for a new source, does not result in any flooding of the network: only the servers along the path of the source are affected. Moreover, DSS does not suffer from lack of scalability during the data-transfer phase since the servers using DSS manage only two classes, QoS and BE, whatever the number of connection in each class is.

4.2. Bandwidth Requirement

Another important factor to take into account when evaluating DMS and DSS is the resource requirements. Indeed, even if DSS goes beyond a trivial peak rate allocation (by using the deterministic effective bandwidth), we have to determine whether its scalability properties does not come at the expense of a more important bandwidth requirement than DMS. We will prove that it is not a case, and that DSS requires less bandwidth than DMS.
Let us first note that for a given network topology and a given set of leaky bucket constrained sources, we can determine the minimum required bandwidth $C^j_{\min}$ at each server j when DSS is used. Indeed, the minimum service rate $C^j_{\min}$ of server j serving n connections with respective deterministic effective bandwidth $(e_i)_{i\in\{1,\ldots,n\}}$ is:

$$C_{\min} = \sum_{i=1}^{n} e_i \tag{9}$$

No such simple result exists for DMS. With DSS, we must compute the additive bound associated to each source and check if this additive bound is smaller than the delay requirement of the source. For this reason the following method is applied to compare the two QoS provisioning schemes:

1. Choose an m-t-p network and a set of sources with their characteristics.
2. Compute the rates of the servers if DSS were in use, with equation (9).
3. Compute the additive bound for each source and check if it is smaller than the delay requirement of the source.

This method allows to check on sample configurations whether DMS can provide the sources with their QoS requirement, with the same amount of bandwidth as DSS. We now discuss the following points: (i) the network architecture, (ii) the parameters of the sources, (iii) the performance parameters used.

4.3. Architecture of multipoint-to-point networks

Consider a general m-t-p network $\mathcal{N}$. As explained in Section 2, a leaky bucket constrained source crossing a server is still leaky bucket constrained. Moreover, the multiplex of such sources is also leaky-bucket constrained. As a consequence, a flow leaving a given subtree of $\mathcal{N}$ is leaky bucket constrained. Thus, we can restrict our study to the case of a single path in $\mathcal{N}$, i.e. a tandem network where sources may enter at any node but exit at the root node only.

4.4. Parameters of the Sources

The leaky bucket parameters of the sources are drawn randomly from Table 1. All sources have the same peak rate. The parameters R (mean rate) and M vary proportionally, which allows to describe the burstiness of a source by a single parameter, M.

Table 1
Connection request parameters

Peak Rate p	*Mean Rate* R	*Burstiness* M
1000	1	10
1000	10	100
1000	100	1000

A source is also characterized by its delay requirement d. Equation (5) indicates that a delay requirement is significant only if it is in the interval $\left[0, M(\frac{1}{R} - \frac{1}{p})\right]$. If d is larger than $M(\frac{1}{R} - \frac{1}{p})$, then the admission control algorithm has only to ensure that the service rate of the source is equal to its mean rate R, and the result has little interest. For a given source S with leaky bucket parameters (p, R, M), let d^{max} be this maximum delay value, i.e. $M(\frac{1}{R} - \frac{1}{p})$. For a given set of n sources with parameters $(d_i^{max})_{i \in \{1,\ldots,n\}}$, the delay requirements are chosen from the following interval: $[\min_{i \in \{1,\ldots,n\}} d_i^{max}, \max_{i \in \{1,\ldots,n\}} d_i^{max}]$.

We propose three scenarios, corresponding to different ways of choosing delay requirements:

First scenario: all the sources have the same delay requirement. We perform a set of tests with the delay requirement varying in the interval $[\min_{i \in \{1,\ldots,n\}} d_i^{max}, \max_{i \in \{1,\ldots,n\}} d_i^{max}]$.

Second scenario: the delay requirement of each source is chosen according to its d^{max} parameter only. More precisely, for a given source S, the delay requirement d is such that $d = \frac{d^{max}}{\alpha}$ with α randomly drawn in the following set of values $\{1.1, 1.5, 2.0\}$.

Third scenario: if an operator were to propose a service based on DMS, a reasonable solution would be to propose a fixed and relatively small number of delay classes rather than a whole continuous set of values. In this scenario, we thus assume that there are three delay classes, namely $\mathcal{D}^1 = \frac{\min_{i \in \{1,\ldots,n\}} d_i^{max}}{10}$, $\mathcal{D}^2 = \min_{i \in \{1,\ldots,n\}} d_i^{max}$, $\mathcal{D}^3 = \max_{i \in \{1,\ldots,n\}} d_i^{max}$.

4.5. Performance Parameters

The main performance parameter is the ratio of networks for which DMS outperforms DSS. Since the considered networks are dimensioned with the minimal service rate for DSS at each node, DMS outperforms DSS (for a given network) when it is able to meet all the sources' requirements. We call this first performance parameter the the failure ratio – since in this case DSS performs worse than DMS. For the cases where DMS outperforms DSS, it is interesting to evaluate how better DMS is over DSS. An evaluation in terms of bandwidth is not possible since there exists no formula giving the minimum bandwidth requirement for DMS. We thus use two performance parameters, called the *mean deviation* E and the *minimum delay deviation* E_{min}, defined as follows:

$$E = \frac{1}{n} \sum_{k \in \{1,\ldots,K\}, i \in I_k} \Big(\frac{d_i - \mathcal{D}_{\mathcal{P}_k}}{d_i}\Big) \tag{10}$$

$$E_{min} = \min_{k \in \{1,\ldots,K\}} \max_{i \in I_k} \Big(\frac{d_i - \mathcal{D}_{\mathcal{P}_k}}{d_i}\Big) \tag{11}$$

where n is the total number of sources, k is the index of the server, i is the index of the source, d_i its delay constraint, I_k is the set of indices of the sources crossing server k, $\mathcal{D}_{\mathcal{P}_k}$ the value of the additive bound along the route of source i, i.e between server k and the root server K.
E is the mean relative deficit, in terms of delay between the two schemes, and E_{min} is the minimum relative difference between the required and provided delays. If E_{min} is equal to zero, then DMS would not be able to serve any extra source since this would lead to a delay constraint violation for the source corresponding to $E_{min} = 0$.

4.6. Scenarios

4.6.1. Identical Delays

Comparison between DMS and DSS for network sizes ranging from 5 to 20 nodes are presented in Figures 6, 5, 7 and 8. In each figure, the x-axis represents the delay multiplicative factor, i.e for a given value x, the delay requirement of all the sources is $d = x \cdot \min_i d_i^{max}$. x varies from 0 to 10, since $\max_{i \in \{1,\ldots,n\}} d_i^{max} \simeq 10 \min_{i \in \{1,\ldots,n\}} d_i^{max}$ with the values of the descriptors of Table 1. Each point corresponds to 10000 networks randomly generated.
A first important conclusion is that the failure ratio is generally small. It is significant only in the interval $[0, \min_{i \in \{1,\ldots,n\}} d_i^{max}]$. But as soon as the delay bound is larger than $\min_{i \in \{1,\ldots,n\}} d_i^{max}$, the failure ratio decreases toward 0. The slope of the decrease increases with the network size. For instance, for a delay bound equal to $\min_{i \in \{1,\ldots,n\}} d_i^{max}$, the failure ratio decreases from 27.5% for a network with 5 servers to 1% for a network with 20 servers. Thus, DSS is perfectly suited for large networks with moderate delay bounds.
It is possible to explain why the failure ratio always increases as the delay bound goes to zero by noting that the deterministic effective bandwidth of the sources converges to their peak rate in this case. This is illustrated by the curve corresponding to: $f(e, p, R) = \frac{1}{n} \sum_i \big(\frac{e_{d_i} - R_i}{p_i - R_i}\big)$. Since the deterministic effective bandwidth is close to the peak rate of the sources, equation (9) indicates that the servers rates tend to the sum of the peak rate of the sources. As a consequence, any service policy, and thus FIFO, becomes able to provide the sources with their QoS requirement.
For configurations where DSS fails at outperforming DMS, we can gain additional information

by using the performance parameters, E (mean deviation) and E_{min} (minimum deviation), previously defined. E_{min} has noticeable values only when the number of servers in the network is small. For instance, for a network with 5 servers, it varies from 15 to 30% whereas for a network with 20 servers, it is almost always less than 8%. Thus, the extra amount of traffic that DMS would be able to serve with the same network configuration is relatively small and decreases with the size of the network. As for the mean deviation, it is always important, ranging from 60 to 70%. This is expected since all the sources have the same delay requirement and if DMS succeeds in serving the sources, it means that the end-to-end delay bound for the longest path is smaller than the delay bound, and thus DMS will also be able to serve all the other sources.
We can conclude that, for this first scenario, DSS almost always outperforms DMS. Moreover DMS, when better than DSS, would almost always fail in serving more sources.

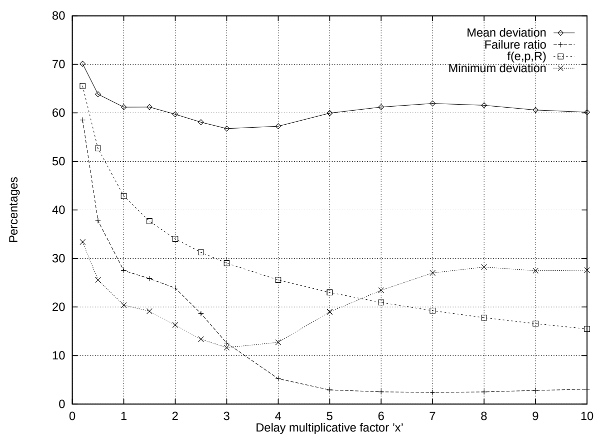


Figure 5. $K = 5$ servers

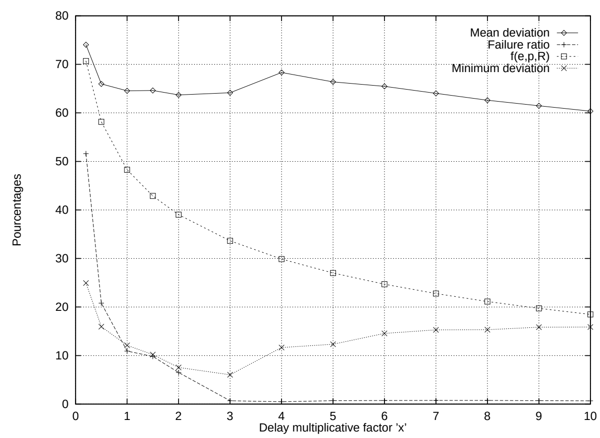


Figure 6. $K = 10$ servers

4.6.2. Delay Bounds Based on the Characteristics of the Sources

When the delay requirement for each source is chosen only according to its own characteristic (as a function of d_i^{max} for source i) the failure ratio is *always* equal to zero. Thus, for this scenario, DSS always outperforms DMS. However, such a complete freedom in the choice of the bound may be unrealistic from a service provider viewpoint. This is why the third scenario assumes only three delay classes.

4.6.3. Three Delay Classes

In this scenario, we assume that a service provider offers three classes of delays, namely $\mathcal{D}^1 = \frac{\min_{i\in\{1,\ldots,n\}} d_i^{max}}{10}$, $\mathcal{D}^2 = \min_{i\in\{1,\ldots,n\}} d_i^{max}$, $\mathcal{D}^3 = \max_{i\in\{1,\ldots,n\}} d_i^{max}$. Note that there is at least one order of magnitude between two delay requirement values. The reason is that it is known that multimedia applications may have very different delay requirements.
Here, once again, whatever the network size, DMS is never able to serve the sources: the failure ratio is *always* equal to zero.
We can conclude from these three scenarios that DSS is a more attractive candidate for QoS

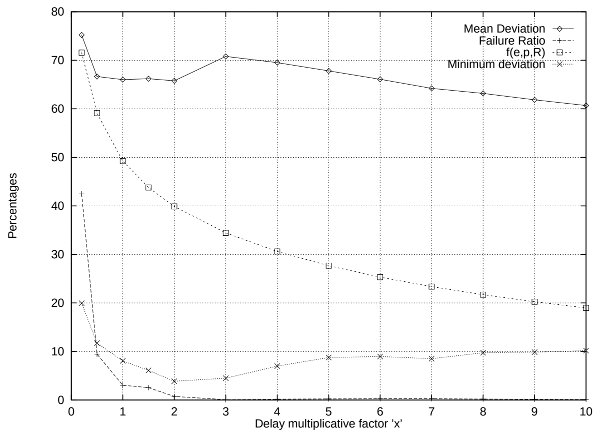


Figure 7. $K = 15$ servers

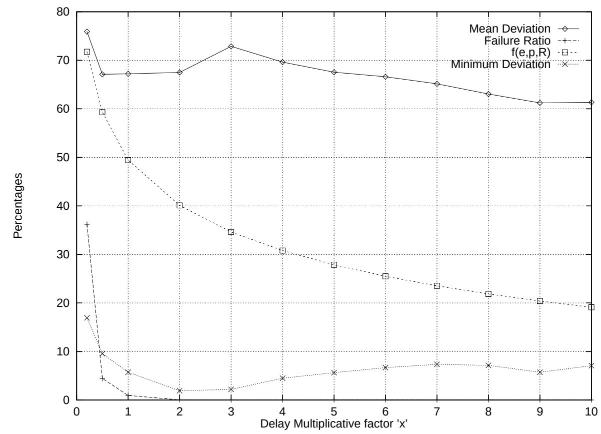


Figure 8. $K = 20$ servers

provisioning not only because of its scalability, but also because this scalability is not obtained at the expense of an over-provisioning of bandwidth.

5. Conclusion

In this paper, we have presented and discussed two QoS provisioning schemes, DMS and DSS, in the case of a multipoint-to-point network. The multipoint-to-point topology is of a great importance, in the Internet backbones, with the growing convergence of the routing and forwarding functions in the Internet through the Multiprotocol Label Switching protocol (MPLS). Studies of the m-t-p architecture have concentrated so far on the load balancing ability of these architectures when multiple m-t-p are used between each pair of ingress/egress nodes. These studies have been carried out with a simple characterization of the source demand through a single parameter representing the service rate required.

Our work further investigates how to establish a complete deterministic QoS scheme in the case of a single m-t-p network for variable bit rate sources. We consider two approaches corresponding to two schemes: (i) the Deterministic Multiplexing Scheme, DMS, which is based on the simple FIFO scheduling policy and (ii) the Deterministic Shaping Scheme, which first shapes the sources before they enter the network and then performs a peak rate allocation using the Complete Partitioning policy.

We compare DMS and DSS according to two criteria: the scalability of their admission control procedure and their resource requirements. It clearly appears that the admission procedure of DSS is scalable while the admission procedure of DMS is not. This is not surprising since, with DSS, we perform a peak rate allocation in the network. Sources are thus treated as constant bit rate sources. On the opposite, DMS has to manage variable bit rate sources, which is far more complicated. But what is more striking is that the scalability of DSS does not come at the expense of a resource over-allocation: it requires, most of the time, less bandwidth than DSS. This means that the shaping performed at the network entrance with DSS is a better option than multiplexing directly the sources as it is done with DMS. Indeed, in the latter case, as DMS uses the FIFO policy which does not differentiate among the sources, all the sources are treated as the one with the more stringent delay requirement. This obviously requires more bandwidth. Future work should study how to integrate DSS in the global context of MPLS where there are several m-t-p networks corresponding to different possible routes between a pair of ingress/egress nodes. We can guess that it should be ease by the fact that with DSS, a source is characterized by only one parameter: its deterministic effective bandwidth.

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. L. Boudec, "An Application of Network Calculus to Guaranteed Service Networks", *IEEE Transactions on Information Theory*, 44(3), May 1998.
3. P. Boyer, F. Guillemin, and M. Servel, "The Spacer-Controller : An Efficient UPC/NPC for ATM Networks", In *ISS'92*, 1992.
4. C.-S. Chang, *Performance Guarantees in Communication Networks*, Springer-Verlag, 2000.
5. A. Conta, P. Doolan, and A. Malis, "Use of Label Switching on Frame Relay Networks Specification", *Internet Request for Comments*, RFC 3034, January 2001.
6. R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks", *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1056, August 1995.
7. R. Cruz, "A calculus for Network Delay, Part I: Network Elements in Isolation", *IEEE Transactions On Information Theory*, 37(1):114–131, January 1991.
8. R. Cruz, "A calculus for Network Delay, Part II: Network Analysis", *IEEE Transactions On Information Theory*, 37(1):132–141, January 1991.
9. B. Davie, J. Lawrence, and K. McCloghrie, "MPLS using LDP and ATM VC Switching", *Internet Request for Comments*, RFC 3035, January 2001.
10. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture", *Internet Request for Comments*, RFC 3031, January 2001.
11. H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering Using Multiple Multipoint-to-Point LSPs", In *Proceedings of IEEE Infocom*, Tel Aviv, Israel, March 2000.
12. G. Urvoy, G. Hébuterne, and Y. Dallery, "Deterministic Call Admission Control in an Accumulation Network", In 5^{th} *Informs Conference in Telecommunications*, March 2000.
13. G. Urvoy, G. Hébuterne, and Y. Dallery, "Deterministic End-to-end Delay Bounds in an Accumulation Network", In *Networking'2000*, May 2000.