

## Research Article

# Network Provisioning Using Multimedia Aggregates

Enrique Hernández-Orallo and Joan Vila-Carbó

Departamento de Informática de Sistemas y Computadores, Universidad Politécnica de Valencia,  
Camino de Vera s/n 46022 Valencia, Spain

Received 5 September 2007; Revised 13 November 2007; Accepted 12 December 2007

Recommended by Yong Pei

Multimedia traffic makes network provisioning a key issue. Optimal provisioning of network resources is crucial for reducing the service cost of multimedia transmission. Multimedia traffic requires not only provisioning bandwidth and buffer resources in the network but also guaranteeing a given maximum end-to-end delay. In this paper we present methods and tools for the optimal dimensioning of networks based on multimedia aggregates. The proposed method minimises the network resources reservations of traffic aggregates providing a bounded delay. The paper also introduces several methods to generate multimedia traffic aggregation using real video traces. The method is evaluated using a network topology based on the European GÉANT network. The results of these simulations allow us to discover the relationship between a required delay and the necessary bandwidth reservation (or the achievable utilisation limit). An interesting conclusion of these scenarios is that, following several recommendations, the network utilisation can reach values of around 80% or higher.

Copyright © 2007 E. Hernández-Orallo and J. Vila-Carbó. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

Multimedia network traffic has grown exponentially in recent years. Providing quality-of-service (QoS) requirements is a key issue in today's network communication. Guaranteeing performance on this communication usually requires some network resource allocation like bandwidth and buffers. Accurately evaluating these resources has become a challenging problem.

Network performance optimisation is an iterative and continual process. This process may consist of real-time optimisation subprocesses and non-real-time network planning subprocesses. The main difference between these subprocesses is the time-scale and the granularity of actions. Real-time optimisation must optimise network resources in small- and medium-size time-scales ranging from microseconds to hours (e.g., traffic shaping, queue management, congestion avoidance methods, etc.). On the other hand, the goal of the network planning subprocess is to improve the network architecture (this includes redimensioning the network resources and changing the network topology). The solution presented in this paper falls into the network planning subprocess category, although it can also be used in the real-time optimisation subprocess. The IETF defines traffic engineering [1] as "that aspect of Internet network engineering that

deals with the performance evaluation and performance optimisation of operational IP networks." In short, the traffic engineering problem is an attempt to optimise resources and to minimise costs. Performance evaluation can be achieved in many different ways, for example, by analytical and empirical methods, simulation, and so forth. IETF lists several instruments that may be applicable to the solution context of traffic engineering. In accordance with these goals, we present both an analytical model for optimal provisioning of a channel and tools for applying this model to real networks.

Several technologies have been devised to provide QoS networking: integrated and differentiated services networks, QoS routing, and multiprotocol label switching (MPLS) [2]. The integration of all these technologies may be the solution for providing real QoS in next generation networks. The goal of the differentiated services (DiffServ) architecture is to provide differentiated classes of service for Internet traffic and to support various types of applications, in addition to specific business requirements [3]. DiffServ are based on the aggregation of individual traffic flows into classes. The IETF has defined a premium class of service for DiffServ (known as PHB: per hop behaviour in the IETF terminology) where the expedited forwarding (EF) provides a differentiated service for high-quality transmission [4]. MPLS [5] can be used as the underlying technology to support traffic engineering.

It can offer dynamic routing and resource management in DiffServ networks. The research works on MPLS focus on optimal routing to reduce network resource utilisation (see the introduction and related work in [6]).

Several works have studied traffic aggregation and network efficiency. The aim of [7, 8] is to evaluate or develop admission control strategies using traffic aggregation. In [9], the authors studied the performance of aggregation. The performance of traffic aggregation in differentiated services networks is studied in [10, 11]. In fact, the goal of [10] is to evaluate the provisioning of an EF class of service using an experimental DiffServ wide area network. The traffic aggregate used in these experiments is a set of constant bit rate (CBR) sources. In [11], the authors studied the impact of traffic aggregation conformity in the context of DiffServ EF PHB. The traffic used in the simulations of this work is made up of synthetic flows where packets are generated according to the Poisson process. These previous works have one disadvantage: the traffic aggregates used in these works are very synthetic and consequently the results are not too much realistic.

In this paper, we focus on multimedia traffic engineering. Multimedia traffic requires not only provisioning bandwidth and buffer resources in the network but also guaranteeing a given maximum end-to-end delay. The proposed method is based on the use of traffic traces for optimising network resources guaranteeing a required delay. Real traffic and composite multimedia traffic aggregates are used to evaluate this method in different scenarios. The novelty of our approach is to use multimedia traffic to create traffic aggregates.

In summary, this paper attempts to answer questions like the following: given a traffic workload, what are the minimal network resources needed to guarantee a desired delay? How can a network be dimensioned to admit a given number of video channels? IETF defines this work as provisioning [1] (the “process of assigning or configuring network resources to meet certain requests”).

## 2. NETWORK RESERVATION ANALYSIS

The goal of this section is to obtain an algorithm to calculate the optimal network resources parameters (bandwidth  $R$  and buffer  $B$ ) for a given delay and a known traffic aggregate.

### 2.1. Traffic flow model

When a new traffic is going to be transmitted in a network, it must specify its requirements and traffic characteristics. This is a kind of contract between the customer and the network. In this paper, we use the classical token bucket specification. This specification is defined by two parameters: sustained bit rate  $r$  and token bucket depth  $b$ .

Given a traffic trace, it is easy to obtain the minimal  $r$  value for a given  $b$  and vice versa. A token bucket  $(b, r)$  corresponds to a constraint function  $\alpha(t) = b + rt$ . Expressing  $b$  as a function of  $r$  conveys an analysis that starts by estimating the bucket fullness of the token bucket model. Assume that the aggregate traffic flow has  $n$  frames and that the number of encoded bits produced by frame  $i$  is  $E_i$ . If the flow source is

constrained by a token bucket, then it cannot transmit more than  $b + rt$  bits at time  $t$ . Considering  $f$  frames per second, and ignoring the finite capacity of the bucket, the bucket fullness (in bits) at the end of the frame period  $i$  is [12, 13]

$$b_i = \max \left( 0, b_{i-1} + E_i - \frac{r}{f} \right). \quad (1)$$

In order to calculate the minimal  $b$  value so that the bucket does not overflow,  $b$  is chosen such that  $b_i \leq b$  for all  $i$ . The value of  $b$  can thus be written as a function of  $r$ :

$$b(r) = \max_{0 < i \leq n} b_i. \quad (2)$$

The function  $b(r)$  is a decreasing function in the domain  $[1, p]$  ( $p$  is the traffic peak rate) as shown in [13].

### 2.2. Node delay equation

The delay bound of any packet departing from a node to an output link can be expressed as [14]

$$D = \frac{B}{R} + E_p, \quad r \leq R, \quad (3)$$

where  $R$  is the configured service rate at the output interface and the total traffic destined to the output interface is bounded by a token bucket of rate  $r \leq R$  and depth  $B$ .  $E_p$  is an error term for the treatment of individual EF packets. Equation (3) is the queuing delay, that is, the time a packet has to wait in order to be transmitted on the link. To this delay, we must add the link delay that is usually a fixed constant that depends on the physical link characteristics. The value  $E_p$  depends on the scheduler used in the node. It is worth to note that the delay expression (3) can also be used for obtaining the node delay for a single channel in a node. In this case, the value  $E_p$  depends on the scheduler and the channel priority. For example, IETF uses this delay bound in the definitions of the differentiated service EF class of service [4].

Solving out for  $R$ , we can obtain the reservation function:

$$R = \frac{B}{D - E_p}, \quad r \leq R. \quad (4)$$

### 2.3. A method for obtaining the optimal network parameters

The goal of this section is to minimise the bandwidth reservation  $R$  for a given traffic trace in order to guarantee a maximum deterministic delay  $D$  with no packet loss. The basis for the proposed method is the network delay formula presented in the previous section. Replacing  $B$  with  $b(r)$  in (4), we have

$$R = \frac{b(r)}{D - E_p}, \quad r \leq R. \quad (5)$$

The following theorem will show that the minimal reservation occurs when  $R = r$  (when the reserved bandwidth is equal to the drain rate), so the equation can be solved by iterations.

**Theorem 2.1.** *The optimal reserve is when  $R(r) = r$ .*

*Proof.* This proof is straightforward: given that  $R(r)$  is decreasing in the domain  $[1, p]$  and that condition  $R(r) \geq r$  must be satisfied, it can be stated that the minimal value is when  $R(r) = r$ . Therefore, this is the convergence criterion.  $\square$

It is important to determine the range of values that make  $R(r)$  converge, that is, to satisfy condition  $R(r) = r$ .

**Corollary 1.**  *$R(r)$  converges to  $r$  if  $1/(D - E_p) \leq p$ .*

*Proof.* Given that the range of  $r$  is  $[1, p]$ , if  $R(r)$  converges to  $r$ , it should take a value in the range  $[p, 1]$ . Since  $R(r)$  is decreasing and it only depends on  $b(r)$ , which is also decreasing, the minimal value of  $R(r)$  is when  $b(r)$  is  $1(r = p)$ , that is,  $1/(D - E_p)$ . This implies that the greatest value in the interval  $[1, p]$  (i.e.,  $p$ ) is at least  $1/(D - E_p)$ .  $\square$

Once it has been demonstrated that the optimal solution occurs when  $R(r) = r$ , an iterative method could be used to find the parameters that optimise reservations. This method is costly; in each iteration the video must be scanned to obtain  $R$  for a bucket depth  $B$ . An implementation of this algorithm can be found in the RTNOU project. The real-time network optimisation utilities (RTNOU) project is a set of network utilities and algorithms to work with traffic traces, test network schedulers, obtain optimal parameters, and so forth. It is not designed as a traffic engineering tool, but some utilities can be used for this purpose. The RTNOU project has a C++ library and a test program. The source code of the library, the program, manual, and some traffic traces can be freely downloaded from the following web: <http://www.disca.upv.es/enheror/RTNOU.html>. The algorithm to obtain the optimal parameters (Section 2) and the different methods to generate traffic aggregates (Section 3) are implemented in the RTNOU library.

### 3. TRAFFIC AGGREGATE WORKLOAD

This section is devoted to explaining the aggregate workload used in the following sections. Two classes of workload are used. The first one is a real traffic trace and its goal is to make provisions for a network using real traffic traces. The second class is a composite workload that is an aggregation of real traffic video traces. The generation of this synthetic traffic trace has proven to be very useful in multimedia traffic engineering.

#### 3.1. Real traffic traces

There are several real traffic traces available such as the Qbone traces, NLANR from the University of Auckland, MAWI from the WIDE backbone. We selected the MAWI traffic traces [15] due to their high resolution and their relatively high mean bandwidth. Although it is not specifically multimedia traffic, it represents a real traffic trace and is perfect for our optimisation process. Specifically, we take a 24-

hour trace from May 14, 1999, from a US-Japan link. This is a typical Internet traffic as shown in Figure 1.

These traces were in tcpdump raw format (near 9 GB of traces), so we distilled them to obtain a simple file that contains the number of bytes transmitted for a given period in each line. Using a period of 40 milliseconds (25 samples per second), the resulting traffic trace has 2 159 980 frames, a mean rate of 6 617 758 bps, and a peak rate of 20 645 200 bps. This is bursty traffic with a burst ratio of 3.12 (the burst ratio is defined as the peak rate divided by the mean rate).

#### 3.2. Composite traffic aggregation

We have focused in this paper on multimedia traffic engineering, and therefore we need multimedia traffic. This section describes how the synthetic traffic aggregate is generated using real video traffic traces. Two sets of variable bit rate (VBR) traffic were used for this aggregation. The first group is the well-known MPEG-1 traces studied by Rose [15] from the University of Wurzburg. The second group is from the Technical University of Berlin [16] and contains MPEG-4, H.261, H.263, and H.263+ traces.

The generation of network synthetic traffic is a classic research topic. There are several analytical models: renewal traffic models, Markov chains, self-similar or fractal models, and so forth. (See [17, 18] for a survey.) Our goal was to model the demand of traffic video in a network, so we generated a composite traffic aggregate using real video traffic traces simulating the Poisson process.

A simple traffic flow consists of the consecutive arrival of discrete entities (packets, frames) with defined intervals. We implemented several methods to generate the traffic aggregation (all these methods can be found in the RTNOU project).

- (i) *Static aggregate.* The aggregate is generated using a given (static) set of traffic traces. This method was similar to the one used in [8].
- (ii) *Dynamic aggregate.* The aggregate is generated dynamically by simulating an arrival process with a given mean bandwidth.
- (iii) *Shaped aggregate.* This is similar to the dynamic aggregate except that the bandwidth is variable; that is, the traffic aggregate is generated following a given traffic shape.

The *static aggregate* is the simplest way to generate a new traffic aggregate. The resulting aggregate is the sum of a given set of traffic video traces. For example, using a set of known workloads, a new file is generated as the sum of the number of bytes transmitted in each period. This produces a very artificial traffic trace because all the traffic is generated at the start time. An improvement of this is to generate the start and duration of the traffic as an exponential distribution. For example, Figure 2(a) shows static aggregate traffic. This traffic was generated using 6 classes of workloads with different codification algorithms: MPEG-1, MPEG-4, H.263 (see Table 1). The start of each load is exponentially distributed with a mean of 100 seconds and duration limited to 1200 seconds (20 minutes). With this configuration, the traffic

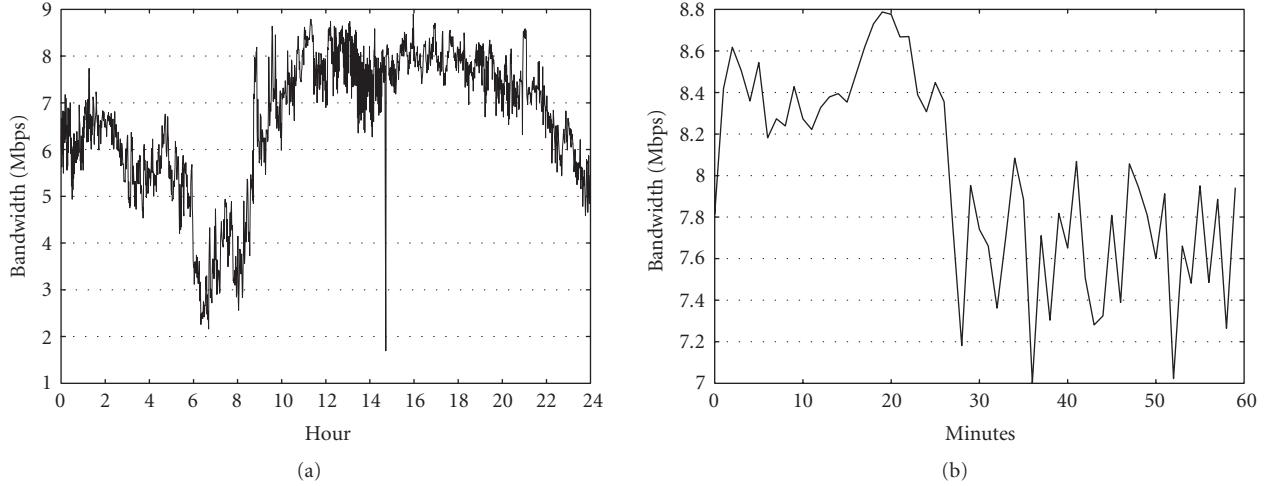


FIGURE 1: MAWI traffic average over 1 minute interval: (a) 24-hour trace, (b) from 11:00 am to 12:00 am trace.

generated has 71 356 frames (47 minutes) with a peak rate of 33 355 200 bps and a mean rate of 9 239 448 bps. As expected, the aggregate burst ratio versus the individual burst ratios was reduced notably to 3.99.

There are several problems with this method of traffic aggregation: it is not easy to generate traffic with a desired mean bandwidth and duration, and the traffic does not follow a real pattern as can be seen in Figure 2(a).

The *dynamic aggregate* tries to resolve these problems. First, we can select the mean bandwidth of the generated aggregate (known as offered bandwidth  $O_B$ ), and second, the traffic is generated simulating the Poisson process. From the offered bandwidth, we can estimate the average number of flows  $O_F$  in the aggregate as the quotient of the offered bandwidth  $O_B$  and the mean bandwidth of the traffic mix. The traffic is generated as follows: flow arrivals are generated according to the Poisson process with average arrival rate  $\lambda$ . In order to simulate this Poisson process, we must calculate the next arrival time. If we distribute exponentially the durations of the traffic with mean  $\beta$ , then the average number of flows in the aggregate is  $\beta \times \lambda$  so  $\lambda = O_F/\beta$ . For example, the mean rate in the traffic set of Table 1 is 480 918. If we want an offered load of 10 Mb, then  $O_F$  must be 20.79. If we use a value  $\beta$  of 100 seconds, this implies an arrival rate  $\lambda = 0.21$  (i.e., the mean arrival time between flows is  $1/\lambda$ , about 5 seconds). In order to simulate the Poisson process, the next arrival time is obtained using a pseudorandom exponential function with mean  $1/\lambda$ . Each flow has traffic characteristics which are chosen randomly from the characteristics of a traffic mix consisting of different video traces. This traffic generation is very similar to the one introduced in [19] and used in [20] for comparing service disciplines.

Figure 2(b) shows a dynamic traffic aggregation for an offered bandwidth of 20 Mbps with duration of one hour. The aggregation is made up of the traffic traces described in Table 2. The mean duration of the flow was set to 500 seconds. The traffic generated had 90 000 frames with a peak

rate of 34 681 800 bps and mean rate of 19 983 113 bps (the burst ratio 1.73 is very low). For this offered bandwidth, the mean number of simultaneous flows was 45. The great advantage of this traffic generation scheme is that using the same traffic workload defined in Table 2 we can generate different aggregates with the desired mean bandwidth.

The main problem with the dynamic aggregate is that the generated traffic has a fixed mean bandwidth. This can be useful in some experiments, but in others, it might be necessary for the offered bandwidth to vary. In order to modify the offered bandwidth dynamically, we introduced the *shaped aggregate*. This traffic aggregate follows a given traffic shape and can reflect different network load patterns (e.g., an hourly, daily, or weekly traffic profile). This can be useful if we only have partial samples (e.g., 1 sample per minute) about the traffic inside a network.

Formally, we have a set of offered bandwidths  $S_O = O_1, O_2, O_3, \dots, O_N$  with a period  $T$ . This implies that for time interval  $[0, T]$  the offered bandwidth must be  $O_1$ , and for time interval  $[T, 2T]$  the offered bandwidth must be  $O_2$ , and so on. The method for generating the shaped aggregate is based on the dynamic aggregate. Accordingly, in the first period we generate a dynamic aggregate with an offered bandwidth  $O_1$ ; in the second period we use  $O_2$ , and so on.

Figure 3(b) shows a shaped traffic aggregate using the Qbone traffic profile (this trace is described in [9] and has a sample period of 5 minutes (see Figure 3(a)). The workload used was the same as the dynamic aggregate (Table 2). The generated aggregate had a peak rate of 107 299 400 bps and a mean rate of 55 240 785 bps (burst ratio = 1.94). As shown, this composite traffic followed a very realistic pattern, similar to the MAWI trace.

#### 4. NETWORK PROVISIONING SCENARIOS

This section details the provisioning of network resources in several scenarios. The goal of the scenarios presented in the

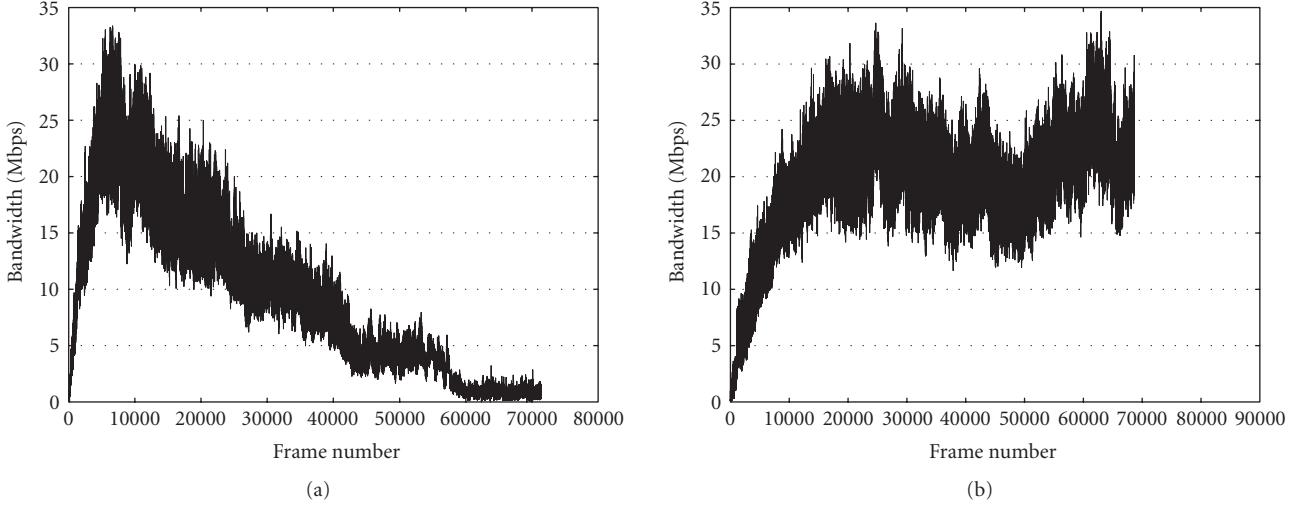


FIGURE 2: (a) Composite static traffic aggregate, (b) composite dynamic traffic aggregate.

TABLE 1: Workload for static traffic aggregate.

Traffic	Loads	Origin	Codec	Frames	Mean (bps)	Peak (bps)	Ratio
Lambs	10	Rose traces	MPEG-1	40 000	182 788	3 355 600	18.36
Soccer	10	Rose traces	MPEG-1	40 000	678 213	4 679 400	6.9
Asterix	10	Rose traces	MPEG-1	40 000	558 718	3 684 440	6.59
Robin Hood	10	Berlin set	H263(VBR)	71 146	765 231	3 812 000	4.98
StarWarsIV	10	Berlin set	MPEG-4(HQ)	90 000	275 282	1 874 000	6.81
Troopers	10	Berlin set	H261(VBR)	90 000	435 277	2 045 400	4.7

following subsections is to answer questions like the following.

- (i) Given a traffic trace, what is the optimal provisioning for a desired delay?
  - (ii) When traffic traces are not available, how can a network be provisioned to support a desired number of video channels?
  - (iii) When a traffic pattern of the workload is available, is it possible to optimise the usage of the network resources?

The network used in the following scenarios is based on the topology of the GÉANT network (as known in April 2004). GÉANT is a pan-European multigigabit data communications network, reserved specifically for research and education use. The network is detailed in Figure 4. The link delays values are approximated based on the propagation delay:  $\sim 5$  ms/1000 km for fibre. For example, we want to set up a channel between the Spain node (ES) and the Sweden node (SE). This channel will be used to transport the traffic aggregate and will use a portion of the bandwidth/buffer resources of the links along the path. We focus our study on the link between Spain (ES) and France (FR). We assume that the node uses a strict nonpreemptive priority queue (PQ) scheduler. For PQ, the latency term  $E_p$  is  $MTU/C$ , where MTU is the maximum packet size and  $C$  is the speed of the output link.

[21]. Then we have  $E_p = 1500 \text{ bytes}/10^*10^9 = 1.2*10^{-7} \text{ seconds}$ .

The end-to-end channel delay can be calculated if the traffic is limited inside the network and the network topology is known [22]. In our case, the traffic is constrained (as bounded by the token bucket) and we know the path of the channel, so we can obtain the end-to-end delay bound as the sum of the delays of the nodes and links along the path [23]. For example, if we set up a channel between the ES and SE nodes (path ES-FR-UK-SE) configuring all nodes with a node delay of 0.01 second, then we have 3 departing nodes and 3 links. That implies an end-to-end delay of  $D = 3 * (D_{node} + D_{link}) = 3 * (0.01 + 0.005) = 0.045$  seconds. This simple approach is very practical because if we know the path of the channel, we can limit the packet delay in a network.

For reasons of brevity, in this paper we present only the results using the network in Figure 4. The experiments with other topologies and network parameters are very similar to the ones presented.

#### **4.1. Provisioning using real traffic traces**

This is a typical traffic engineering scenario. The goal is to optimise the network resources used for a traffic aggregate for a given node delay. These are the steps to follow.

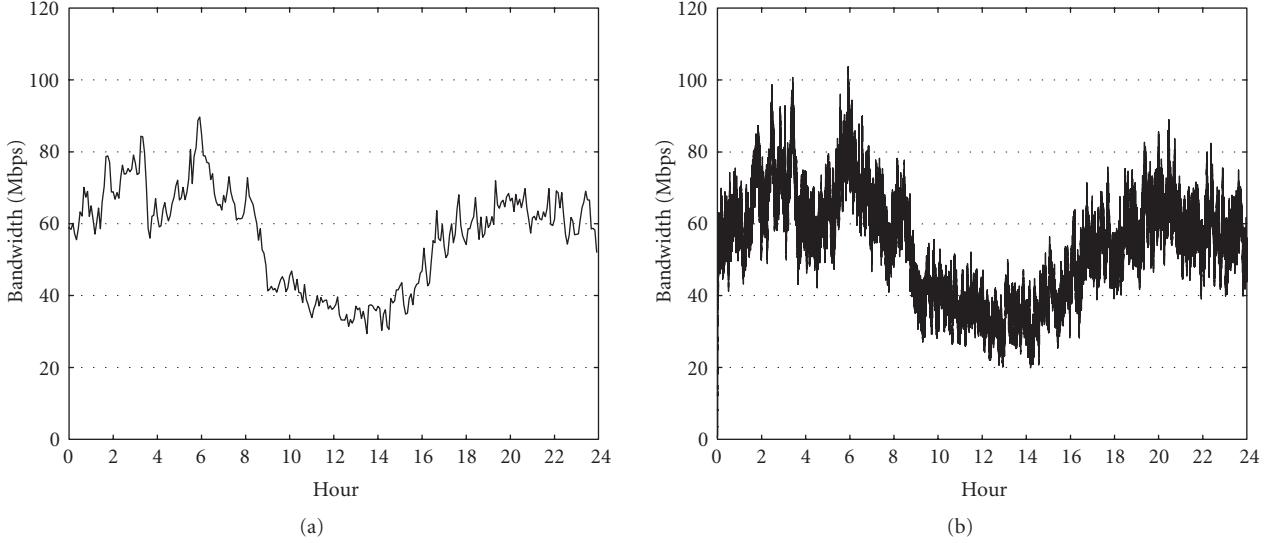


FIGURE 3: (a) Qbone traffic (5-minute trace), (b) Qbone-shaped traffic aggregate.

TABLE 2: Traffic aggregate composition.

Traffic	Origin	Codec	Frames	Mean (bps)	Peak (bps)	Ratio
Asterix	Rose traces	MPEG-1	40 000	558 718	3 684 400	6.59
Dino	Rose traces	MPEG-1	40 000	326 953	2 990 800	9.15
Lambs	Rose traces	MPEG-1	40 000	182 788	3 355 600	18.36
Mtv	Rose traces	MPEG-1	40 000	615 105	5 730 000	9.32
News	Rose traces	MPEG-1	40 000	150 767	1 155 000	7.66
Term	Rose traces	MPEG-1	40 000	272 618	1 989 000	7.3
TheFirm	Berlin set	H261(64 kbps)	90 000	63 890	397 800	6.22
Troopers	Berlin set	H261(VBR)	90 000	435 277	2 045 400	8.19
Robin Hood	Berlin set	H263(VBR)	71 146	765 231	3 812 000	4.98
Aladdin	Berlin set	H263(256 kbps)	31 226	737 843	3 860 600	5.23
Lecturer	Berlin set	H263(VBR)	45 459	123 517	1 106 592	9.33
Soccer	Berlin set	MPEG-4 (High qual.)	90 000	1 106 582	3 585 400	3.24
Formula 1	Berlin set	MPEG-4 (Med. qual.)	44 998	291 183	1 406 400	4.83
StarWars IV	Berlin set	MPEG-4 (High qual.)	90 000	275 282	1 874 000	6.81
Mr.Bean	Berlin set	MPEG-4 (High qual.)	89 113	581 983	3 050 200	5.24
N3 Talk	Berlin set	MPEG-4 (High qual.)	90 000	552 622	3 412 400	6.17
Office Cam	Berlin set	MPEG-4 (High qual.)	67 498	396 860	1 982 200	4.99

- (i) Measure the traffic inside a link node for a given period. The measured traffic must be the traffic that is going to be transmitted.
- (ii) Obtain the optimal resource reservation for the channel using the method described in Section 2.
- (iii) Modify the network configuration to assign the new reservation for the channel.

One important decision that must be taken is the time-scale of the provisioning. The measured traffic can be a long-term trace (daily or weekly traces) or a short-term trace (hourly traces). This depends on the network capability to

support dynamical variation in the reservation of the channel resources (e.g., an hour).

To test this scenario, we simulated the GÉANT network using the MAWI traffic traces as described in Section 3.1. Two examples are presented here: the first one is a long-term dimensioning using the 24-hour trace as presented in Figure 1(a), and the second one is a short-term dimensioning using the same traffic trace from 11:00 am to 12:00 am (Figure 1(b)).

*Long-term provisioning.* For example, say we want to ensure a node delay of 0.02 second for the aggregate. Using the whole MAWI trace and the algorithm presented in Section 3,

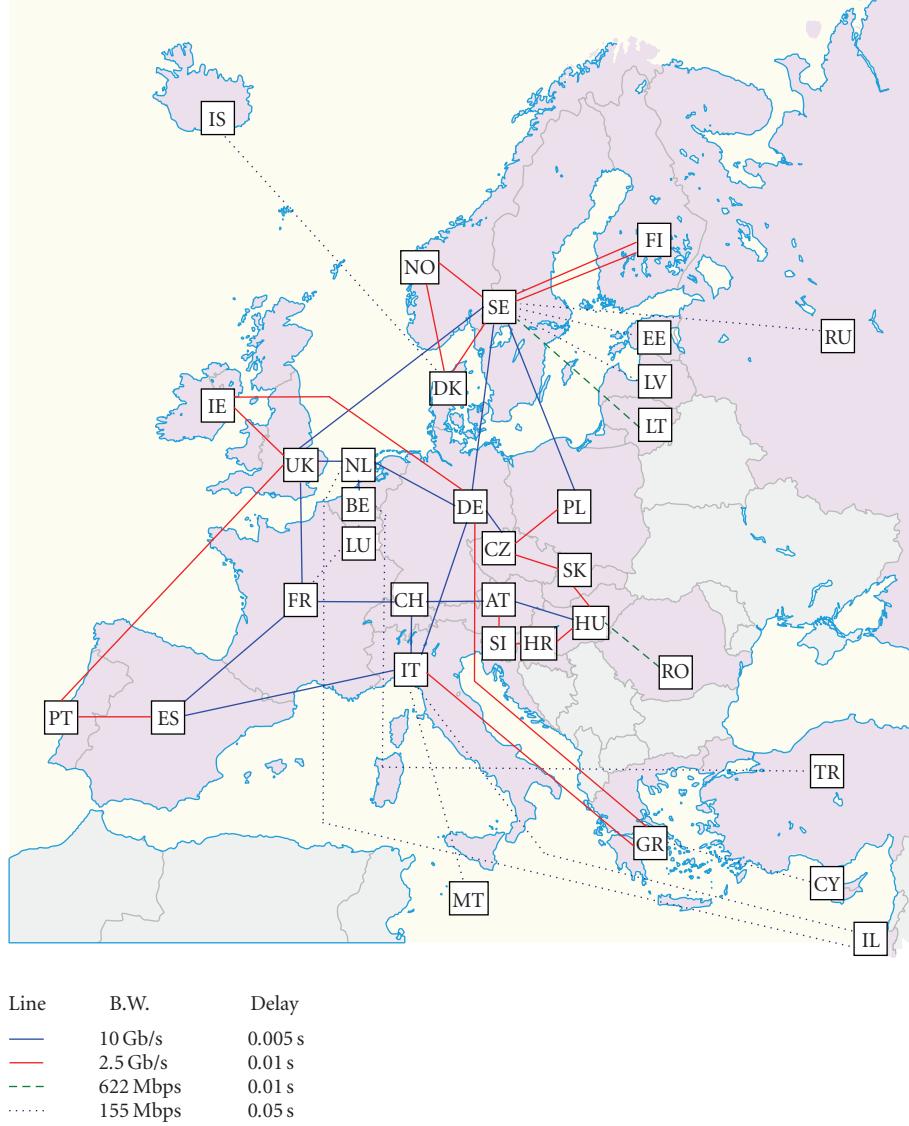


FIGURE 4: GÉANT network topology.

the optimal channel parameters are  $R = 10\,867\,901$  bps and  $B = 108\,671$  bits (these values represent the minimal bandwidth and buffer reservation in a node). Therefore, the channel must be dimensioned with these parameters in the 3 links that form the aggregate path.

Figure 5(a) shows the  $R$  and  $B$  parameters as a function of the required delay ranging from 0.002 second to 0.5 second. As shown, the bandwidth reservation  $R(D)$  is a decreasing function. This plot is very helpful because we can determine whether a given delay requires the reservation of too many resources. The increase in  $B$  versus time is quite linear. On the other hand,  $R$  decreases very quickly from 20 Mbps to 8 Mbps in the range from 0.002 second to 0.1 second. However, if we increase the delay, the value of  $B$  increases to 1 Mb and this implies that every node must store at least 1 Mb which some networks may not be able to do. Therefore, in order not to waste network resources, we must adjust the network delay correctly.

*Short-term provisioning.* Using a one-hour trace from 11:00 am to 12:00 am, the optimal channel parameters are  $R = 11\,809\,142$  bps and  $B = 59\,026$  bits for 0.01-second delay. As expected, the resources reservation is lower than using the 24-hour trace (the shorter the time trace, the lower the burst ratio). Figure 5(b) shows the  $R$  and  $B$  parameters as a function of the required delay ranging from 0.002 second to 0.5 second. The reservations for the short-term results are lower than the reservation for the long-term results, mostly when there are short delays.

#### 4.2. Provisioning for a dynamic multimedia aggregate

This scenario supposes that we do not have a trace of the traffic to be transmitted in the channel (this is usually the case when a network is first deployed). Therefore, it is very useful to predict the multimedia load that a channel will admit or/and how many resources must be reserved for the channel

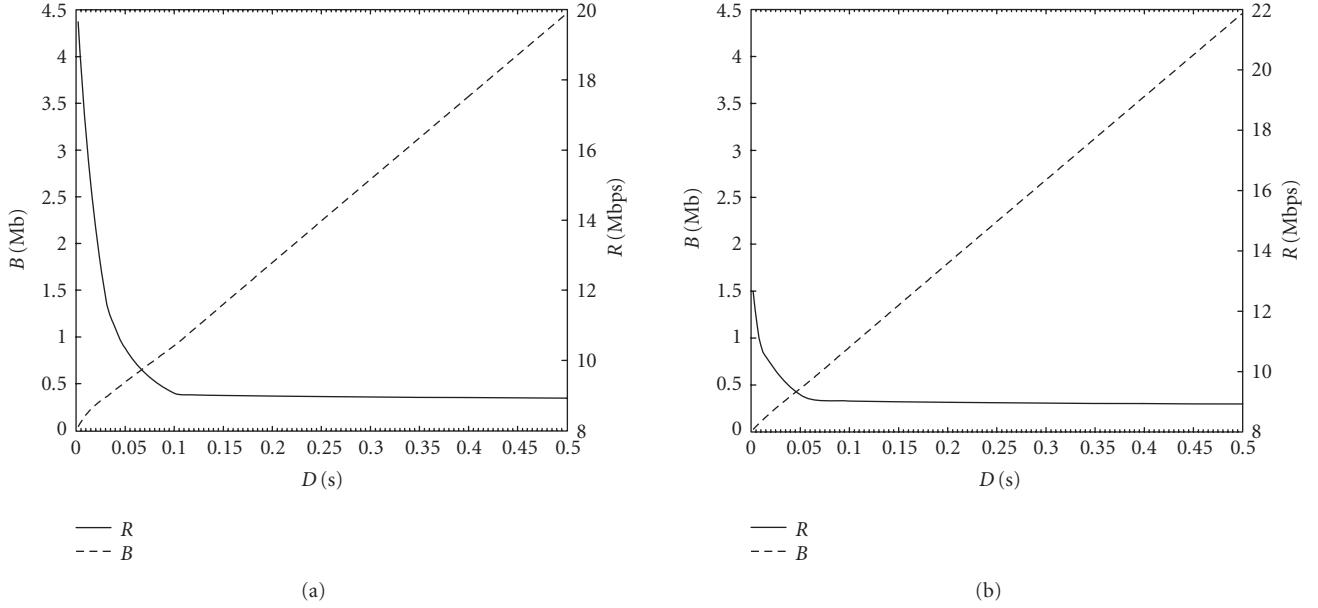


FIGURE 5: Dimensioning for (a) 24-hour MAWI traffic and (b) 1-hour MAWI traffic.

to support a desired multimedia load. These are the steps to follow.

- (i) Create a composite traffic for a desired offered bandwidth using the dynamic aggregate method as described in Section 3.2.
- (ii) Obtain the optimal resource reservation using the method described in Section 2.
- (iii) Configure the network channel with this reservation.

For example, using the dynamic traffic generated in Section 3 with an offered bandwidth of 20 Mbps, the optimal channel parameters are  $R = 29\,780\,679$  bps and  $B = 297\,787$  bits for 0.01-second delay. Therefore, by provisioning the network channel with these parameters, a mean of 45 video flows is admitted. Figure 6(a) shows the optimal channel parameters for a delay range of 0.002 to 0.5 second. The curves are similar to the real-traffic traces, although the relative difference between the reservation for low delays and high delays is smaller (about 7 Mbps).

Using this dynamic traffic, it is easy to obtain the channel parameters for different offered bandwidths. Figure 6(b) shows the value of  $R$  depending on the delay for different aggregates. Five aggregates are used with the following offered bandwidths 10 Mbps, 20 Mbps, 40 Mbps, 60 Mbps, 80 Mbps, and 100 Mbps. The figure shows that the reservation decreases very slowly for delays greater than 0.02 s. The main reason is that the aggregate traffic is bursty less than the MAWI traffic traces (the burst ratio of the dynamic aggregate traffic is below 2). Figure 7(a) shows the value of  $B$  depending on the delay, and the results show that it is practically linear.

#### 4.3. Provisioning for a traffic pattern

This scenario supposes that we have a traffic pattern or low resolution traffic traces inside the network. As in Section 4.1,

the goal is to obtain the optimal parameters in order to support this traffic. These are the steps to follow:

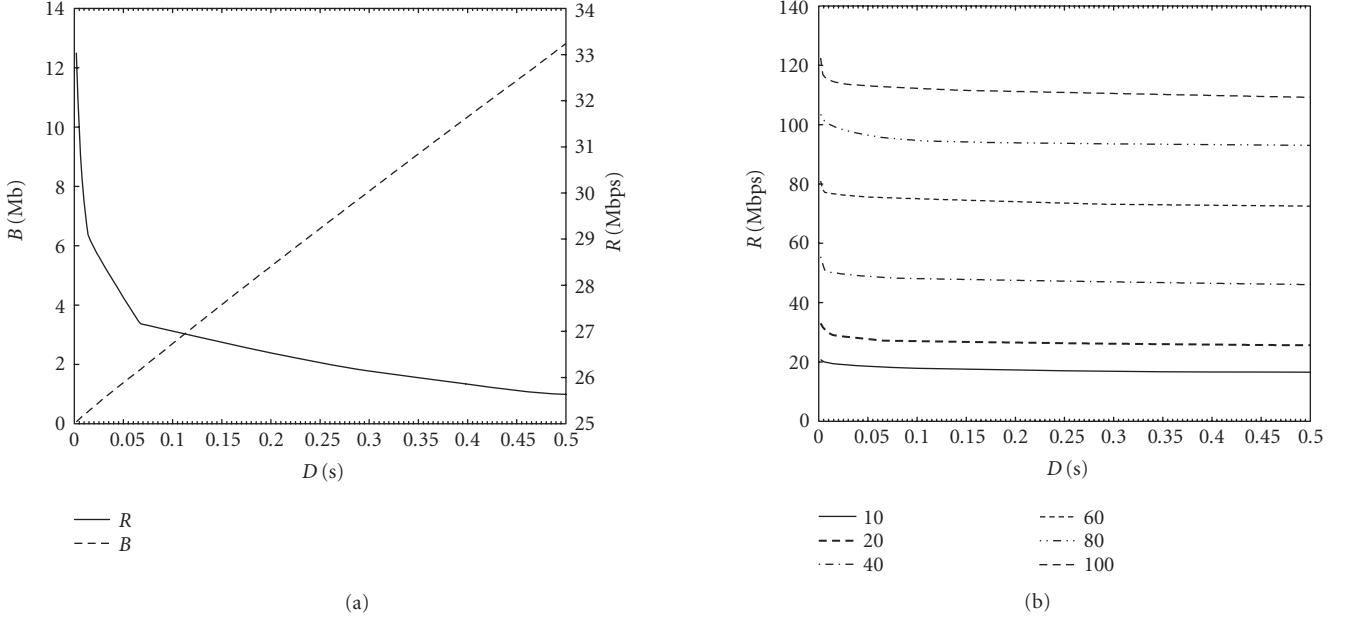
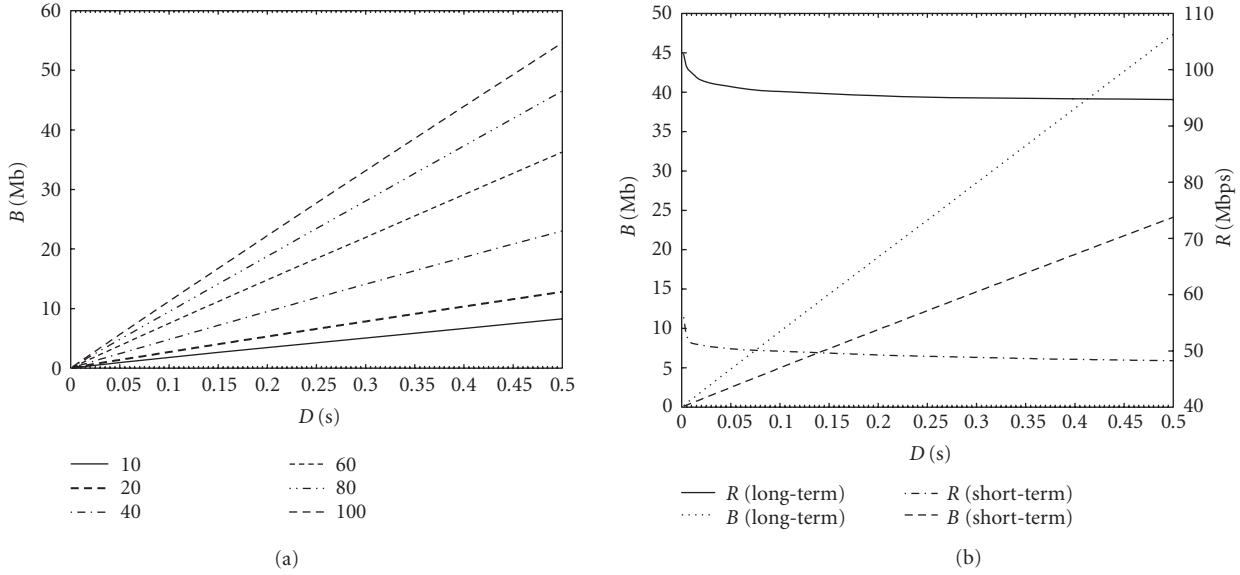
- (1) Create a composite traffic using a shaped aggregate as described in Section 3.2.
- (2) Obtain the optimal resource reservation using the method described in Section 2.
- (3) Modify network configuration to assign the new reservation.

As an example of this scenario, we used the QBone trace as described in Section 3.2 (Figure 3(b)). Therefore, by using this traffic pattern we generated a traffic aggregate. Following the tests for the MAWI traffic, we plot the long-term and short-term parameters for this traffic aggregate in Figure 7(b). The results are very similar to the previous ones.

#### 4.4. Bandwidth utilisation

If the goal is to reduce the bandwidth reservation in the network, the bandwidth utilisation level provides a simple way of configuring the node delay. The bandwidth utilisation is defined as the quotient between the mean rate of the aggregate and the bandwidth reservation ( $R$  value). For example, the MAWI daily traffic has a mean rate of 6 617 758 bps. Then, for a delay of 0.01 second, the bandwidth reservation is 16 516 377 bps and the utilisation is 40%.

As stated in [9], one critical factor that affects the network utilisation is the traffic characteristics of the flows making up the aggregate. Figure 8(a) shows the bandwidth utilisation for the dynamic traces. The figure clearly shows that the utilisation is better when the offered bandwidth aggregate is greater. That is the effect of the statistical multiplexing among aggregate flows: the greater the number of aggregated flows, the better the utilisation. Another conclusion is that it

FIGURE 6: (a) Dimensioning for dynamic traffic (offered load 20 Mbps), (b)  $R$  for different aggregates.FIGURE 7: (a)  $B$  parameter for different aggregates, (b) dimensioning for Qbone-shaped traffic.

is not efficient to use delays greater than 0.2 second (the increment of the utilisation for a delay greater than 0.2 second is very low).

Figure 8(b) shows the bandwidth utilisation for long-term and short-term MAWI and Qbone traces. It clearly shows that it is more efficient to use short-term traffic traces (for the MAWI traffic the utilisation is very high). Therefore, if the network resources can be dynamically dimensioned, it would be better to obtain the values of  $R/B$  depending on the time. This can be seen in Figure 9 for the MAWI traces using a delay of 0.01 second. The channel requirements were calculated using a 1-hour period. This greatly improves the total

utilisation which was 68.2% versus the 40% utilisation of the long-term dimensioning.

#### 4.5. General conclusions

This section presents three scenarios for the application of the network provision method described in Section 2. The first one uses a real traffic trace and can be used to optimise the utilisation of a running network, the second uses a composite traffic and can be used at the initial provisioning of the network, and the last one is a combination of the previous

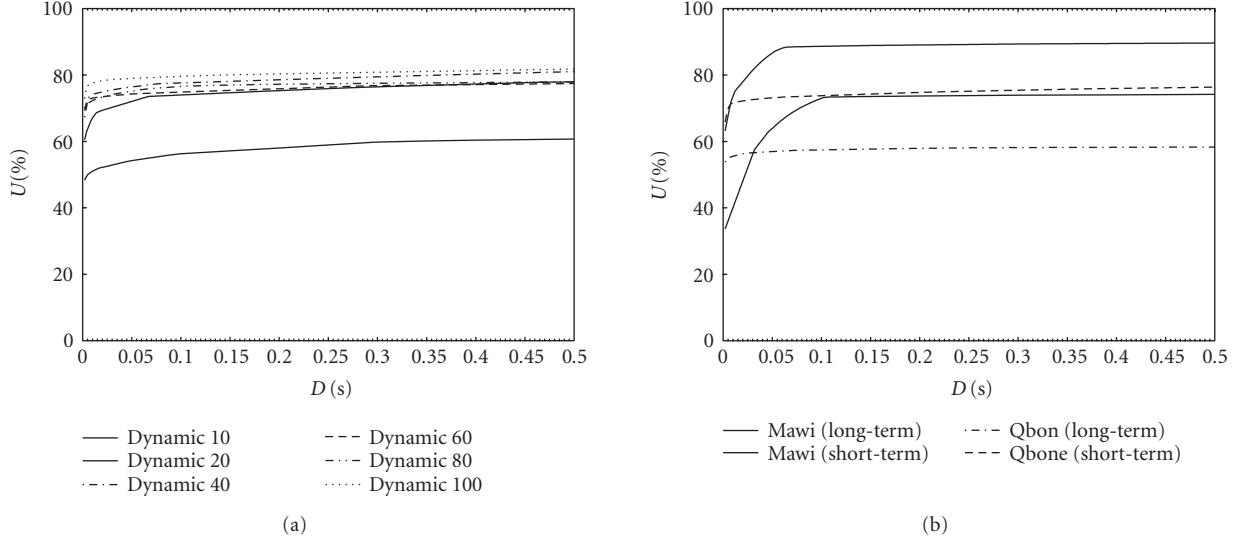


FIGURE 8: Bandwidth utilisation for (a) dynamic aggregates and (b) short- and long-term traces.

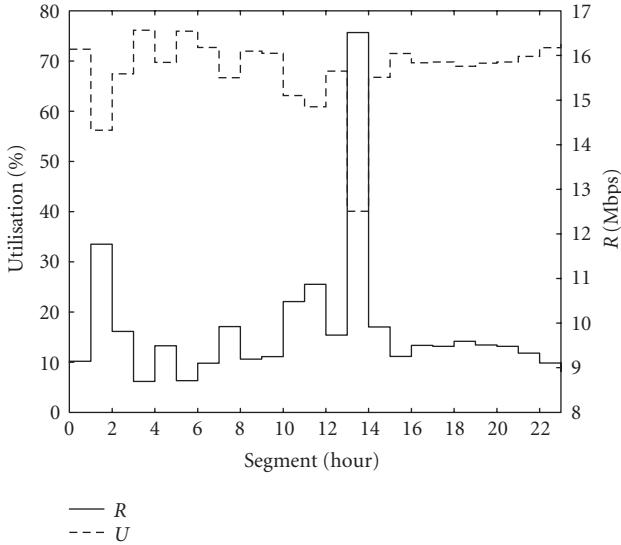


FIGURE 9: Dynamic short-term provisioning for MAWI traffic.

ones—it needs a coarse description of the network traffic and the provisioning is done using a shaped aggregate.

Several conclusions can be drawn from the scenarios presented in this paper. First, a very reduced delay implies a reduced utilisation of network resources; if the delay is increased, the utilisation is better but the buffer (the  $B$  value) also increases. Second, the greater the number of loads in an aggregate, the better the network utilisation (due to the statistical multiplexing). Third, in order to use the network more efficiently, it is better to use short-term traces than long-term traces. However, this implies that the network must be capable of redimensioning the network resources periodically. This is a tradeoff: if the aggregate is not frequently redimensioned, the signalling overhead in the core network is minimal and the network utilisation is low. On the

other hand, if the aggregate is redimensioned too frequently, the network utilisation is high but the signalling overhead could be very high as well [9, 24].

One question arises when a traffic workload is used for network dimensioning: what happens if, in a given period, the traffic load is greater than the workload used to dimension the network? This is a classic issue in traffic engineering and there are several solutions. In a strict approach, if we want to assure that no packet will be lost and no packet will miss its deadline, we must use a worst case workload to dimension the network and limit the traffic inside the network. In a permissive environment, we can use average workloads and assume that some packets will miss their delay bound or that the network will drop them when the traffic load is high.

## 5. CONCLUSIONS

Bandwidth represents the dominant cost factor in any network-based application or service. Optimal provisioning of network resources is crucial for reducing the service cost of multimedia transmission. This is the goal of traffic engineering: the design, provisioning, performance evaluation, and tuning of operational networks. The fundamental problem with provisioning is to have methods and tools to decide the network resource reservation for a given quality of service requirement.

Multimedia traffic requires not only provisioning bandwidth and buffer resources in the network but also guaranteeing a given maximum end-to-end delay. The method introduced in this paper minimises the network resources reservations of traffic aggregates providing a bounded delay. The method can be used to optimise operational networks and to deploy new ones. The scenarios presented in this paper show that the utilisation can reach values that are closer to 80%. All these methods and algorithms have been

implemented and can be downloaded free of charge from the web.

Another interesting contribution of this work is the introduction of several methods to generate traffic aggregates using real video traces that can be used as benchmarks to evaluate network reservations.

As basic guidelines, to obtain high network utilisation, the following general recommendations could be carried out: (i) do not use very low end-to-end delays (whenever possible), (ii) use short-term traces if the network is capable of periodically redimensioning the network resources, and (iii) use a large aggregate (with a high number of flows) instead of a small aggregate.

## ACKNOWLEDGMENTS

This work was developed under grants of the *Generalitat Valenciana* (GV/2007/192) and the Spanish Government (CICYT), TIN2005-08665-C03-03.

## REFERENCES

- [1] D. Awdanche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," RFC 3272, Internet Engineering Task Force, May 2002.
- [2] X. Xiao and L. M. Ni, "Internet QoS: a big picture," *IEEE Network*, vol. 13, no. 2, pp. 8–18, 1999.
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, Internet Engineering Task Force, December 1998.
- [4] B. Davie, A. Charny, J. C. R. Bennett, et al., "An expedited forwarding PHB (Per-Hop behavior)," RFC 3246, Internet Engineering Task Force, March 2002.
- [5] D. Awdanche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," RFC 2702, Internet Engineering Task Force, September 1999.
- [6] P. Trimintzios, T. Baugé, G. Pavlou, P. Flegkas, and R. Egan, "Quality of service provisioning through traffic engineering with applicability to IP-based production networks," *Computer Communications*, vol. 26, no. 8, pp. 845–860, 2003.
- [7] J. Schmitt, M. Karsten, L. Wolf, and R. Steinmetz, "Aggregation of guaranteed service flows," in *Proceedings of the 7th IEEE International Workshop on Quality of Service (IWQoS '99)*, pp. 147–155, London, UK, June 1999.
- [8] J. Qiu and E. W. Knightly, "Measurement-based admission control with aggregate traffic envelopes," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 199–210, 2001.
- [9] H. Fu and E. W. Knightly, "Aggregation and scalable QoS: a performance study," in *Proceedings of the 9th IEEE International Workshop on Quality of Service (IWQoS '01)*, vol. 2092 of *Lecture Notes in Computer Science*, Karlsruhe, Germany, June 2001.
- [10] T. Ferrari, "End-to-end performance analysis with traffic aggregation," *Computer Networks*, vol. 34, no. 6, pp. 905–914, 2000.
- [11] R. A. Guérin and V. Pla, "Aggregation and conformance in differentiated service networks: a case study," *Computer Communication Review*, vol. 31, no. 1, pp. 21–32, 2001.
- [12] A. R. Reibman and A. W. Berger, "Traffic descriptors for VBR video teleconferencing over ATM networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 329–339, 1995.
- [13] E. Hernández-Orallo and J. Vila-Carbó, "A new approach to optimize bandwidth reservation for real-time video transmission with deterministic guarantees," *Real-Time Imaging*, vol. 9, no. 1, pp. 11–26, 2003.
- [14] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, vol. 2050 of *Lecture Notes in Computer Science*, Springer, New York, NY, USA, 2001.
- [15] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proceedings of the 20th Annual Conference on Local Computer Networks (LCN '95)*, pp. 397–406, Minneapolis, Minn, USA, October 1995.
- [16] F. H. P. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, 2001.
- [17] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol. 32, no. 3, pp. 70–81, 1994.
- [18] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," Tech. Rep. 120, Institute of Computer Science, University of Würzburg, Würzburg, Germany, July 1995.
- [19] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control of piecewise linear traffic envelopes at EDF schedulers," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 558–570, 1998.
- [20] E. Hernández-Orallo, J. Vila-Carbó, S. Sáez, and S. Terrasa, "Optimal traffic shaping with deterministic guarantees using GPS scheduling," in *Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems: Protocols and Systems for Interactive Distributed Multimedia (IDMS/PROMS '02)*, vol. 2515 of *Lecture Notes in Computer Science*, pp. 1–14, Springer, Coimbra, Portugal, November 2002.
- [21] A. Charny, J. C. R. Bennett, K. Benson, et al., "Supplemental information for the new definition of the EF PHB (expedited forwarding Per-Hop behavior)," RFC 3247, Internet Engineering Task Force, March 2002.
- [22] A. Charny and J.-Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *Proceedings of the 1st International Workshop on Quality of Future Internet Services (QoFIS '00)*, Berlin, Germany, September 2000.
- [23] E. Hernández-Orallo and J. Vila-Carbó, "Efficient QoS routing for differentiated services EF flows," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 91–96, Murcia, Cartagena, Spain, June 2005.
- [24] S. Sargent and R. Valadas, "Tradeoffs between signaling and resource utilization in DiffServ networks with hierarchical aggregation," in *Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems: Protocols and Systems for Interactive Distributed Multimedia (IDMS/PROMS '02)*, vol. 2515 of *Lecture Notes in Computer Science*, Coimbra, Portugal, November 2002.

