Modelling a Virtual Source to Virtual Destination Dynamic Bandwidth Reallocation Scheme With DSPN

Nathalie Omnès
Mitsubishi Electric ITE-TCL
Rennes – France
omnes@tcl.ite.mee.com

Annie Gravey
ENST Bretagne
Brest – France
Annie.Gravey@enst-bretagne.fr

Raymond Marie
IRISA Rennes
Rennes – France
Raymond.Marie@irisa.fr

Keywords : Connection-oriented, Hop-by-hop, Bandwidth, Negotiation, Deterministic and Stochastic Petri Nets.

Abstract : We propose a protocol which aims at offering a cheap transmission for non-real time loss sensitive applications and describe its validation and evaluation. This protocol can be implemented equally in ATM or in IP. It is based on a virtual source to virtual destination bandwidth reallocation scheme, which relies on two types of devices: generic nodes, or core routers, and the user end device. In order to manage bandwidth reservations, buffers are implemented in these devices, and threshold are marked in them. Furthermore, feedback can be sent so as to reduce the upwards rate of transmission or retransmission.

To validate and evaluate this protocol, a *DSPN* model is developped, and compared to a continuous time Markov chain model.

1. INTRODUCTION

1.1. The Challenge of Quality of Service

The ITU-T defines Quality of Service (**QoS**) as being ``the collective effect of service performance, which determines the degree of satisfaction of a user of a service''. Obviously then, the nature of QoS depends on the application.

Among applications, one can distinguish real-time ones and non real-time ones. In both cases, a sequence of packets coming from the same source and going to the same destination is induced, which is called a **flow**. On the one hand, real-time applications are associated to intolerant flows, called **rigid**, for which the network has to offer a high level of QoS. It implies to implement a lot of different functionalities, including in particular admission control, resource reservation, policing and scheduling. On the other hand, non real-time applications are associated to tolerant flows, also called **elastic** flows, which can adapt to the resources instantly available in the network. They permit to optimise the global resources, by regulating the activity of elastic sources depending on the current level of reservation operated by rigid flows. In this article, we will focus on elastic flows. See [1] and [2] for rigid flows.

Unless over-provisioning is applied in the network, specific mechanisms have to be specified to deal with elastic flows. With the current interest in DiffServ and MPLS (see [3],[4]), IP is agreeing with this point of view, so that both IP and ATM are getting closer and closer. We are going to study protocols which could exist equally in IP or ATM, assuming the application is non real-time but loss sensitive.

Given that there exists at least two classes of flows and that it is worth developing sophisticated mechanisms to deal with elastic flows, several types of solutions can be applied (see [5]-[14]). In all cases some feedback has to reach the source so that it adapts its emissions to the instantly available resources. This creates at least a control loop between the source and another device in which a resource control is applied. This control loop can indeed end-to-end or hop-by-hop, in which case the dialogue is limited in scope to one or few links, but several dialogues are necessary at the same time (see Figure 1).

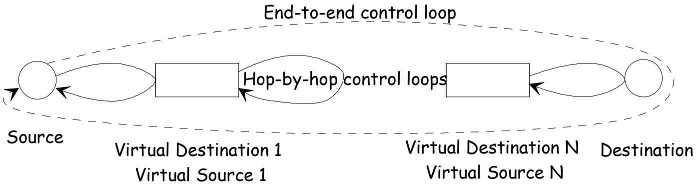


Figure 1 End-to-end and hop-by-hop control loops

In the case of an end-to-end control loop, the time needed for a feedback to reach the source can be long and its knowledge of the network topology obsolete (see [15]). In case of a hop-by-hop control loop the source will interact with an intermediate switch or router which will be its **virtual destination** (see [5], [6], [8], [11] and [16]). Then this router will interact either with the destination or with another router so that it will be its **virtual source**. Even if this control loop is limited in range, there is still a delay before any change is perceived by the source or by an intermediate virtual source. This induces an uncertainty. Hence, to guarantee a low loss rate, buffers are used to regulate the activity of the source as well as each virtual source, if any.

Even though end-to-end control loops are interesting, the current trend is rather to consider hop-by-hop mechanisms. It is for example the point of view that is adopted in DiffServ (see [17]-[18]). Therefore, we will only consider hop-by-hop control loops, that is virtual source to virtual destination schemes.

1.2. Outline

The details of the dynamic bandwidth allocation protocol are first introduced in Section 2. This description involves two different devices : intermediate switches or routers (2.1) and the user end system (2.2). Modelling issues are tackled in 2.3.

The modelling of this protocol is then developped in Section 3. We first describe the software in Section 3.1. Modelling is described in 3.2, and performance measures in Section 3.3. The comparison of the different types of models is addressed in Section 3.4. The first results concerning the protocol performance are listed in Section 3.5.

We finally conclude in Section 4.

2. DESCRIPTION OF THE DYNAMIC BANDWIDTH ALLOCATION SCHEME

The source generates data at any rate ranging from 0 to its peak rate. Furthermore, the path to be followed by packets is known before the source begins its emission and a minimum bandwidth Δ_1 is reserved all along this path.

2.1. General Description of the Nodes

Here we describe generic nodes, which include a virtual source and a virtual destination. In particular, they are located on the path from source to destination. Ideally, all intermediate switches or routers play these roles but, in particular to allow convergence, this is not compulsory.

We assume there are L different levels of reservation on each link. Let us denote them by

$$\Delta_1, \Delta_2, \ldots, \Delta_L.$$

We assume that $L \geq 2$, and that $\Delta_i < \Delta_j$, $\forall 1 \leq i < j \leq L$.

As we are in a connection-oriented environment, the bandwidth of the incoming traffic cannot go up unless the node has explicitly authorised it to do so. Still, a buffer has to be implemented to cope with variations on input and output bandwidth. Actually, as reservations are operated hop-by-hop, a virtual source can let its incoming bandwidth to go up to Δ_{i+1} even when its output bandwidth is $\Delta_i < \Delta_{i+1}$.

Usual methods to manage reservations on output of the virtual source n aims to predict or estimate future needs (see [10]-[12], [14] and [15]). In [9] and [13], several thresholds are marked in its buffer :

$$s_1 < s_2 < \ldots < s_{L+1} \leq C,$$

where C is the size of the buffer (see Figure 2).

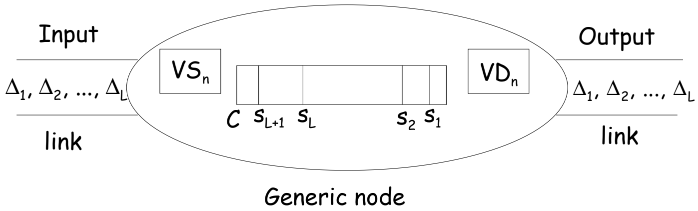


Figure 2 Generic node

Virtual Source Behavior

When the current level of reservation on output of node n is lower than Δ_i, $2 \leq i \leq L$, and the level of its buffer exceeds s_i, the virtual source n generates a signalling message asking the virtual destination $n+1$ for an output bandwidth of Δ_i. The time elapsed between the emission of this request and the reception of the answer is RTT_{n+1}, the round trip time between the virtual source n and the virtual destination $n+1$.

Conversely, when node n output bandwidth is Δ_i, its virtual source releases this bandwidth to go down to Δ_{i-1} when the level of its buffer drops below to s_{i-1}. The effect of this decrease is immediate. A signalling message is needed for the virtual destination $n+1$ to update its reservations.

Hence, bandwidth Δ_i is requested when the level of node n buffer exceeds s_i, and released when it drops below s_{i-1}. This creates an hysteresis cycle, which aims at providing stable bandwidth changes.

When the virtual source n receives a positive response to its request for a bandwidth increase that was addressed to the virtual destination $n+1$, it immediately increases its output rate. On the contrary, if this response is negative, it formulates a new bandwidth request.

Note that most studies do not take into account the possibility to refuse the requests (see [12] and [15]).

Virtual Destination Behavior

The network can be congested, and hence demands for bandwidth increases be refused. When this happens, virtual source $n-1$ has to be locked to avoid losses. In practice, this is achieved when the level of node n buffer exceeds s_{L+1}. A signalling message is then sent to the virtual source $n-1$, obliging it to immediately reduce its output bandwidth to Δ_1. Note that, due to RTT_n, the effect of this order is delayed.

The virtual source $n-1$ is unlocked either when the output bandwidth of the virtual source n increases, or when the buffer level in node n drops below s_1. In order to avoid instability, unlocking virtual source $n-1$ only means allowing it to send requests for a bandwidth greater than Δ_1.

Depending on the current level of bandwidth reservation in node n, a request for more bandwidth coming from the virtual source $n-1$ is either accepted or refused. Assuming all connections transmit data at a constant bit rate, let C_{Ln} be the capacity of the link between the virtual source *n-1* and the virtual destination *n*, C_{curr} be the current level of reservation on this link, and ρ_n be the acceptable load on this link. Then the virtual destination *n* allows the virtual source *n-1* to increase its emission rate from Δ_k to Δ_i if the current level of its buffer is below s_{L+1} and

$$C_{curr} - \Delta_k + \Delta_i < \rho_n C_{Ln}, \ \forall 1 \le k < i \le L, \ \forall n=1,\dots,N.$$

2.2. Detailed Description of the User End System

The behaviour of the user end system is similar to the one of a generic node. The main difference is that the rate of the source can increase even if it has not been explicitly authorised to do so. A buffer is again implemented to cope with these variations. This device is depicted in Figure 3.

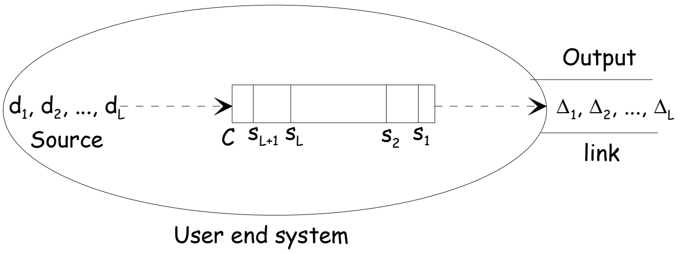


Figure 3 User end system

The source can generate packets at *L* different levels,

$$d_1, d_2, \dots d_L,$$

where $d_1 \ge 0$, and d_L is the peak rate of the application. We assume that $d_i<d_j, \ \forall 1\le i < j \le L$, $d_1<\Delta_1$, $d_i \le \Delta_i, \ \forall 2 \le i \le L$, and that $\Delta_i < d_{i+1}, \ \forall i=1,\dots,L-1$.

We mark thresholds $s_1, s_2, \dots, s_{L+1}$ in the user end system buffer, which are utilised just as in any generic node.

2.3. Modelling

The system to model can be seen as a sequence of queues. There are three well-known approaches to tackle such a system. The first approach is to develop a theoretical model and numerically analyse its behaviour. Depending on the complexity of the system this approach is not always possible. It also implies to rely on several assumptions, and checking whether these are realistic is not easy. Still, it leads to closed-form formulas, which may enable a good understanding of the behaviour of the system. The second approach it to develop a discrete event simulation model. This leads to statistical results, which has the disadvantage of being less accurate than results of theoretical models. Yet it can be overwhelmed by a more faithful representation of the system. Another drawback is that simulation may induce very long computation times. The third and last approach is to develop a prototype, which is out of the scope of our work.

Considering the complexity of the system, even a discrete event simulation approach would require a lot of simplifications. As our aim is to catch the general behaviour, and not to precisely model each entity, we decided to develop a theoretical model. A simulation model would be interesting to develop for further study.

Within theoretical approaches to modelling, there exists once again several possibilities (see [2], [5], [11], [12] and [19]). Markovian models are widely used, because they are relatively easy to solve, and may represent worst cases. These models can either be discrete or continuous in time. Continuous time Markov models represent all flows as Poisson processes, which characteristics are far from the streams we want to model. Actually, we assumed all streams are constant bit rate, which corresponds to periodic flows. To model these flows, a data unit is generated each time a deterministic variable expires. In a Poisson model, this deterministic variable is replaced by an exponential random variable, radically changing the statistical properties of the flow. Still, because of the simplicity of Markov theory, we are going to evaluate the system with such a model.

A discrete time Markov model could indeed enable the representation of periodic flows. Unfortunately, the number of states of the Markov chain dramatically increases with the ratio d_L/d_1. Furthermore, the transition matrix of the discrete time Markov chain has to be recomputed each time a rate d_i or Δ_i changes. We therefore decided not to adopt this approach.

Another possibility would be to use fluid models (see [12]). In this case, only incoming and outgoing processes are represented in each node, not taking into account the buffer level. As the protocol we defined is deeply depending on buffers, modelling it with fluid flows would dramatically change its characteristics. This is why we eliminated these models.

Finally, we decided to focus on regenerative Markov processes (see [20]), because they enable the representation of both exponentially distributed random variables, and deterministic variables. In particular, it leaves us the possibility to model Poisson flows as well as periodic flows. Furthermore, we will use Deterministic and Stochastic Petri Nets (*DSPNs)* to create our models, because it offers an easy way to validate and evaluate the system. Unfortunately, to allow a numerical computation, the number of periodic flows is limited to one. Still the representation is more accurate than for Markovian models, as we highlight in 3.4.

3. DETERMINISTIC AND STOCHASTIC PETRI NETS MODELLING

3.1. Software Implementation

DSPNexpress1.5 (see [21]) is used for validating the protocol and evaluating its performances. This tool can only deal with a single deterministic transition at once, which limits us to a single periodic flow.

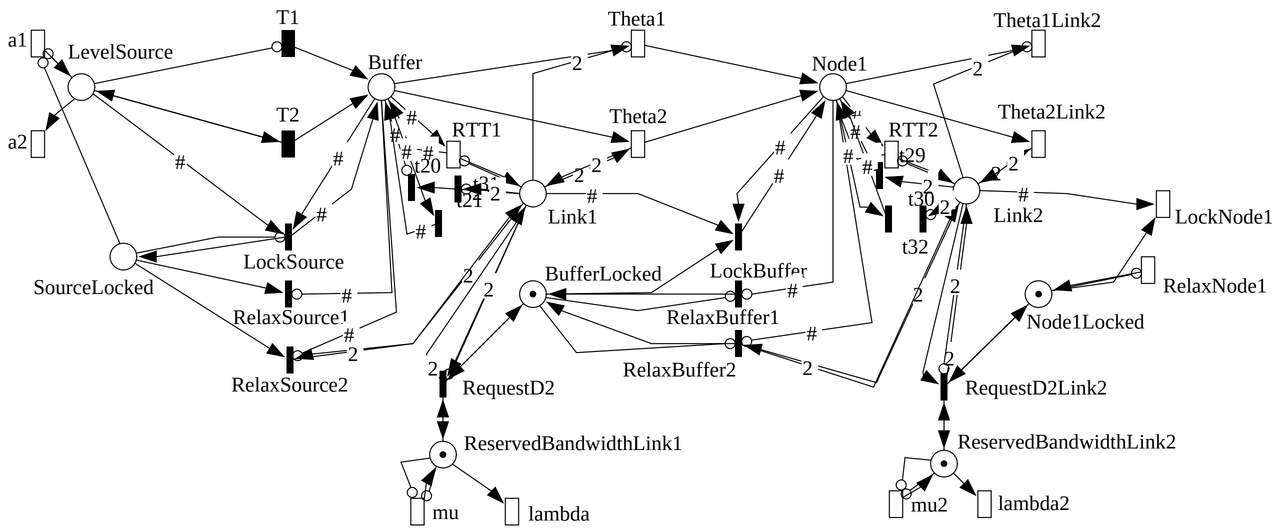


Figure 4 *DSPN* model including 1 generic node

3.2. System Modelling

We now assume that $L=2$ and data units are of constant size. This is the case if ATM is used, and otherwise it simplifies the system. Figure 4 shows the *DSPN* model of the system including the user end system and the first generic node.

Data emission is represented by places *LevelSource* and *SourceLocked*, as well as 7 transitions (*a1, a2, T1, T2, LockSource, RelaxSource1* and *RelaxSource2*). Depending on the source needs and its imposed lockings, either transition *T1* or transition *T2* is enabled. *T1* corresponds to the emission of a data unit every $1/d_1$ time unit, which models a periodic flow of rate d_1, whereas *T2* models a periodic flow of rate d_2.

The user end system buffer is represented by places *Buffer*, *BufferLocked*, and several transitions (*LockBuffer, RelaxBuffer1, RelaxBuffer2*). This buffer is either emptied by transition *Theta1*, corresponding to an output bandwidth of Δ_1, or by transition *Theta2*, which corresponds to Δ_2. Only one of these transitions can be enabled at once, depending on the state of the output link. The output link of the user end system is represented by place *Link1*. This place is related to the user end system buffer by several transitions, depending on the virtual source and the virtual destination decisions. This includes *RTT1* which models the emission of requests for bandwidth increases. The delay *RTT* is assoiated to this transition. Transitions *RequestD2*, and place *Reserved-BandwidthLink1* model *CAC* decisions. Transition *lambda* models the arrival of flows competing for bandwidth on this link, while transition *mu* models their departure.

The first generic node, its virtual source, virtual destination, and output bandwidth are modelled the same way as the user end system, without the source functionalities.

3.3. Performance Indicators

We associate to this *DSPN* model several performance measures, including the mean rate of emission of the user, the probability for the source to be locked, as well as the mean time elapsed in the locked state and between two periods of locking. We further model the distribution of the user end system buffer level, and the mean sojourn time of packets in this device. As for the source, we also characterise lockings. Finally, concerning the output link of the user end system, we model the mean reserved rate on this link, the duration of periods of allocation of Δ_1 and Δ_2, as well as the additional resources needed for signalling.

In the first generic node, we represent the same performance indicators as in the user end system, except the ones related to the source.

3.4. Analysis of the impact of deterministic transitions

Let us first study the limitation of the model without deterministic transitions. This model is derived from the *DSPN* model depicted in Figure 4 by replacing transition *T1* and *T2* by exponential transitions of same mean duration. It can therefore be solved using a continuous time Markov chain. We will call it the Markovian model, by opposition to the regenerative Markov case. We can solve this model with one generic node, with buffers of size equal to 72 data units. Adding the second generic node, the number of data unit in each buffer has to be reduce to 8. The limitation lies around 10^6 different markings.

For the *DSPN* model with deterministic transitions, the limitation lies around 2 10^4 states, meaning we cannot include more than one generic node, with buffers limited in size to 8. This does not permit to properly study the dynamic behavior of the system. Still, this model can be compared to previous one, where *T1* and *T2* are exponential.

Variations on 10 parameters are imposed. The results given by the *DSPN* models with and without deterministic transitions are always the same from a qualitative point of view. Experimentations also show that *DSPN* models with deterministic transitions give better performances than Markovian models.

Let us now describe the results of an experience, where a variation on the round trip times is imposed. They show that the mean rate of emission is higher with deterministic transitions than without them. Still, reservations on the first link are lower, and so is the probability to lose data. This shows that *DSPN* models with deterministic transitions are more optimistic than Markovian models. Figure 5 shows the mean number of data units per time unit that the source generates.

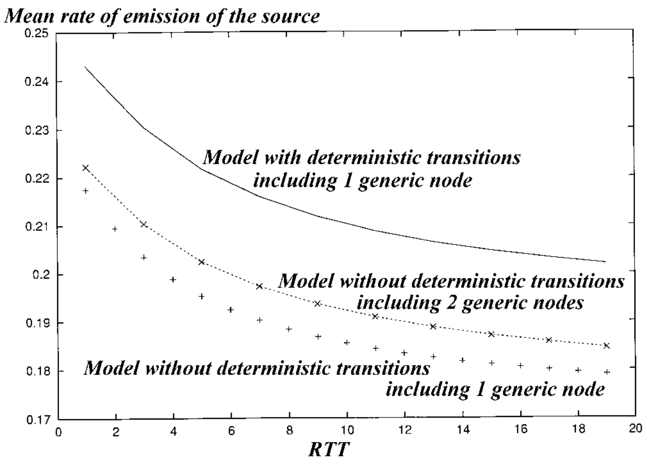
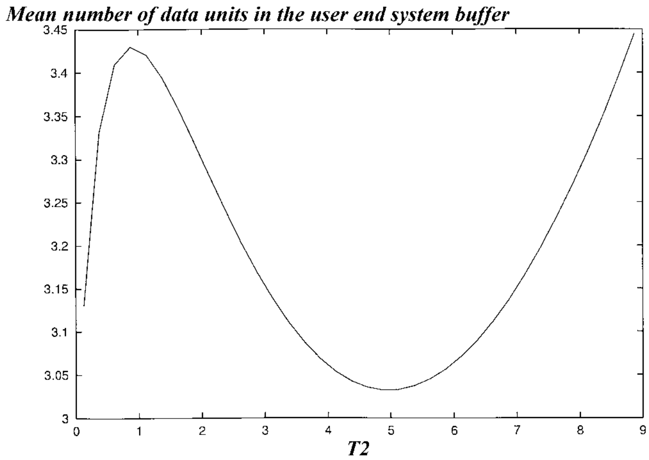


Figure 5 Mean rate of emission of the source

The Markovian model including one generic node gives results that are 10% lower than the same model with deterministic transition. Comparing this last *DSPN* model with deterministic transitions and the Markovian model including the second generic node, we observe that results are only 8% lower with the Markovian model. During other experiments, we also observe that the source always generates less data with a Markovian than with the equivalent *DSPN* model including deterministic transitions. The difference ranges from 9% to 20%. Furthermore, the Markovian model including the secong generic node gives results lying in between the two previous ones, with a difference compared to the model with deterministic transitions ranging from 3% to 9%. Finally, the Markovian model including the third generic node gives results that are even closer to the model with deterministic transitions including a single generic node.

3.5. Evaluation of the Protocol Performance

Consider another experimentation in which we make d_2 (the peak rate of the source) and Δ_2 (the maximum bandwidth that can be allocated to the flow on each link) vary. More precisely, we impose variations on $T_2=1/d_2$ and on $\Theta_2=1/\Delta_2$, under the condition $T_2=\Theta_2$. When T_2 increases, the peak rate of the source and the maximum link bandwidth decrease. Therefore the system is converging towards a system with a single rate of emission of the source, as well as a single level of reservation on each link. This is a particular case, in which the system is stable. We therefore expect the mean number of data units in the user end system buffer to decrease while T_2 increases. Figure 6 shows the



result of the experimentation.

Figure 6 Mean level of the user end system buffer

We see, when T_2 is larger than 5, that the level of the user end system buffer increases. This is due to the decrease in ouput bandwidth. In particular, it shows that decreasing the source sporadicity, which means decreasing the ratio d_L/d_1, does not by itself increase the performance of the protocol.

We further observe that, when T_2 is smaller than 1, the level of the user end system buffer also increases. This is due to the round trip time between the user end system and the first generic node. Actually, this parameter is set to 1. Furthermore, we use buffers which size equals 4. Therefore, when T_2 is smaller than 1, more than 1 packet is generated before reception of the response to a request for a bandwidth increase. This means that the level of the user end system buffer has exceeded its threshold s_{L+1} in between, which induces a locking of the source. This highlights that a misconducted estimation of *RTT*, which may be possible in practice, can lead to unexpected behaviors. This should be confirmed by other experimentations, with larger buffers.

4. CONCLUSION

In this article, we modelled a complex virtual source to virtual destination bandwidth reallocation with *DSPNexpress1.5*, a software that solves *DSPNs*. This tool is interesting in particular because it is flexible, and permits to represent a lot of characteristics and performance parameters. Unfortunately, it is limited with respect to complexity, so that we have to consider either only a few number of core routers, or else buffers of small size.

Experimentations show that models including deterministic transitions, solved with Markov regenerative processes, are more optimistic than equivalent models without deterministic transitions, which correspond to continuous time Markov chains. Furthermore, the results obtained with both types of models are always in accordance.

Some preliminary results concerning the protocol performance highlight two different unexpected behaviors. In particular, catching the influence of the peak rate of the source and of the round trip time between the user end system and the first generic node appear to be sensitive issues. The influence of these parameters should be carefully studied.

Future work will include exploiting the models to reach a more precise analysis of the system. Some simulation models should also be developped, to compare the results of the three types of evaluations.

5. REFERENCES

1. Annie Gravey and Pierre Boyer, *Cell Delay Variation Specification in ATM Networks*, Model. and Perf. of ATM Technomogy (C-15), p. 305-324, Elsevier Science B.V.
2. Wassim Matragi and al., *Jitter Calculus in ATM Networks : Multiple Nodes*, IEEE/ACM Transactions on Networking, vol. 5, No. 1, p.122-133, February 1997.
3. RFC 2475 : *An architecture for Differentiated Services*. 12/98.
4. *Special issue on MPLS*, IEEE Comm Magazine, 12/99.
5. Rajeev Agrawal and al., *Performance bounds for Flow Control Protocols*, IEEE/ACM transactions on networking, Vol. 7, No. 3, p.310-323, June 1999.
6. N. Anerousi, A. Lazar, *Virtual Path Control for ATM Networks with Call Level Quality of Service Guarantees*, IEEE/ACM trans. on net., vol. 6, No. 2, p.222-236, 04/98.
7. Pierre E. Boyer and Didier P. Tranchier, *A reservation principle with application to the ATM traffic control*, Computer networks and ISDN systems, 24 (1992), 321-334.
8. Magda Chatzaki and al., *Resource Allocation in Multiservice MPLS, IWQoS'99*, p. 197-206, London, May - June 1999.
9. Serge Halberstadt, Daniel Kofman, *A dynamic Bandwidth Allocation Mechanism for connectionless traffic over ATM networks*, ITC 15, Washington, 1997.
10. Baochun Li and al., *Optimal State Prediction for Feedback-Based QoS Adaptations*, IWQoS'99, p. 37-48, London, May 31 - June 4, 1999.
11. Michel Mandjes and Hans Van Den Berg*, Some new technique for resource allocation in single- and multi-link ATM systems*, ITC 15, p. 1257-1268, 1997 Els. Science B.V.
12. L. Reiss, L. Merakos, *Performance analysis of an adaptative bandwidth reservation scheme for ATM virtual path traffic*, Computer networks and ISDN systems, 28 (1996), 391-400.
13. J. L. Rougier and al., *Traffic Management for Connectionless Services over ATM*, Telecom. Systems, 12 (1999) p. 193-200.
14. Shigeo Shioda and al., *Self-sizing network: a new concept based on autonomous VP bandwidth adjustment*, ITC 15, p. 197-1006, 1997 Elsevier Science B.V.
15. F. Clérot and al., *Dynamical Resource Reservations Schemes in an ATM Network Using Neural Network-Based Traffic Prediction*, 5th works. on Perf. Model. on ATM Net, 07/97.
16. G. Bianchi and al., *Packet Management techniques for Measurement Based End-to-end Admission Control in IP Networks*, J of Comm. and Net, Vol.2, No.2, p.147-156, 06/0.
17. RFC 2597 : *Assured Forwarding PHB Group*, 6/99.
18. RFC 2598 : *Expedited Forwarding PHB Group,* 6/99.
19. R. Addie and al., *Broadband Traffic Modeling: Simple Sol-utions to Hard Problems*, IEEE Com. Mag., p.88-95, 08/98.
20. Hoon Choi and al., *Markov regenerative stochastic Petri nets*, Performance Evaluation 20, 337-357, 1994
21. C. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*, John Wiley and sons, 1998.

6. BIOGRAPHY

Nathalie Omnès received in 1996 a Master of Science in applied mathematics–probability, from the university of Rennes, France. She then joined the CNET, France Telecom research center, to begin her PhD in 1996. She has been working as a research engineer in Mitsubishi Electric since 1999, while finishing her PhD.

Annie Gravey is currently head of the computer engineering department at ENST-Bretagne, a French engineering school. Before that, she conducted studies related to the design of mechanisms for controlling broadband traffic at France Telecom R&D. She took part in the specification of operational resource allocation policies for ATM services, and in the introduction of constraint based routing in broadband networks. Her current research interests concern the control of multimedia services over B-ISDN and IP networks, the design of traffic engineering methods, and QoS specification and management. She received her PhD in Applied Mathematics and Signal Theory from Paris University in 1981.

Raymond A. Marie received the Doctorat d'Ingénieur and the Doctorat d'Etat es-Sciences Mathématiques from the University of Rennes, France, in 1973 and 1978, respectively. From 1977 to 1999, he was a Research Manager of an INRIA group in modelling. He spent the 1981- 1982 academic year as a Visiting Associate Professor at North Carolina State University, Raleigh, NC, USA. Since 1983, he is a Professor at the Computer Science Department of the University of Rennes. His active research interests include performance evaluation of computer systems, high speed networks and reliability computation of complex systems.