Automatic Self-Configuration of the Logical Network using Distributed Software Agents

J.L. Marzo, P. Vilà, A.Bueno, L. Fàbrega, E. Calle
Institut d'Informàtica i Aplicacions, Universitat de Girona
Lluís Santaló Av.
17071 Girona, SPAIN
{marzo | perev | bueno | fabrega | eusebi}@eia.udg.es

***Abstract* – In this paper, we present a system for dynamic network resource configuration in environments with bandwidth reservation. The proposed system is completely distributed and automates the mechanisms for adapting the logical network to the offered load. The system is able to dynamically manage a logical network such as a Virtual Path network in ATM or a Label Switch Path network in MPLS or GMPLS. The system design and implementation is based on a Multi-Agent System (MAS), which make the decisions of when and how to change a logical path. Despite the lack of a centralised global network view, results show that MAS manages the network resources effectively, reducing the connection blocking probability and, therefore, achieving better utilisation of network resources. We also include details of its architecture and implementation.**

I. INTRODUCTION

Although the technology of high speed networks offers users increasing transmission capacity, the fact is that the increase in the amount of data to be transmitted is higher than the network capacity. This is due to the high increase in the number of new users and the appearance of new services (multimedia and interactive resource-consuming services). Therefore, it is clearly necessary to use network resources efficiently.

Large telecommunication companies have been using powerful network management tools, which provide network statistics and analysis tools to help human network managers in decision-making. These centralised decision-making could suffer a scalability problem when excessive network monitoring traffic arises.

Typically the main resource to be managed at a logical network level is bandwidth or capacity; which requires the network technology to have some kind of reservation mechanism, i.e. the ability to establish Logical Paths (LP) over the physical network (e.g. MPLS [1], GMPLS [2], ATM [3], etc.). Then, the user connections are established through these LPs. This is the concept of logical network (Figure 1). The LPs are reservations of bandwidth between different nodes in order to facilitate the establishment of user connections. The main characteristic of this logical network is its flexibility; this means that it is possible to change the logical topology, the capacities assigned to each LP etc. as required and hence the logical network can be adapted to changes in traffic profiles [4][5].

The Logical Network can be calculated in an optimisation process given the physical topology, the predicted traffic demands and the performance constraints, e.g. [6]. However, once the LPs are set up, the initial design may not be optimal due to either discrepancies between the real and predicted traffic patterns or changes in them. Thus an adaptive procedure is required to adjust the logical network bandwidth allocations and topology dynamically. Nowadays this adaptation is usually performed periodically (e.g. every hour, morning / afternoon / night topologies, daily, weekly, etc) on a semi-permanent basis. This is usually done by the management centre because this adaptation typically consists in recalculate the whole logical network topology. However, this brings the problem that the adaptation does not possibly happen when there are congestion problems in the network.

The proposed system [7] focuses on this idea of logical network, i.e. a dynamically configurable network. The proposed distributed architecture based on software agents groups and integrates several network resource management functions related to the logical network concept. These functions include the load balancing mechanisms. The dynamism and automation of these mechanisms represent a complex task and several proposals rely on Distributed Artificial Intelligence mechanisms, i.e. Multi-Agent Systems (MAS) [8] [9].

In Section II we briefly present the load balancing mechanisms. Section III introduces the MAS architecture and main characteristics. Section IV describes the implemented algorithms, after that, in section V, we present the results obtained for a set of load balancing experiments. Finally, we give our conclusions.

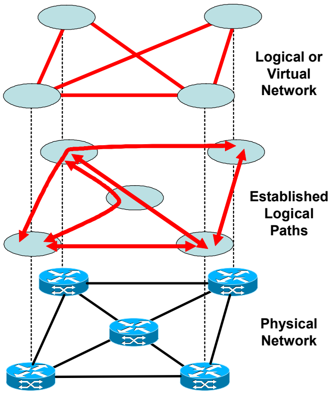


Fig. 1. Logical Network Concept

II. LOAD BALANCING

Load balancing attempts to manage the bandwidth assigned to the LPs, i.e. changing their bandwidth in order to better adapt the Logical network to the offered traffic. The ultimate objective is to maximise network performance. When, due to unforeseen changes in the offered load and/or when the Logical Network design is not optimal, some LPs can become under-utilised and others congested, i.e. all the LP bandwidth is already assigned to several connections and new connections requests are being rejected. Some of these rejected connections could be accepted if the traffic loads were better balanced. Therefore, one of the main objectives of load balancing is to minimise the Connection Blocking Probability (CBP), i.e. the probability that an offered connection is rejected.

There are two actions usually performed by the load balancing systems in order to increase the bandwidth of a congested LP: re-allocation of bandwidth and re-routing LPs [10]. For instance, consider the situation shown in Figure 2, where there are three LPs beginning in node 1 and suppose that LP 1-2 is congested. Therefore LP 1-2 needs an increase in bandwidth.

Bandwidth re-allocation means that LP 1-2 bandwidth can be increased without the need for re-routing. The required bandwidth for increasing LP 1-2 can come from two possible sources: the available bandwidth of the physical links, not assigned to any other purpose; and the unutilised bandwidth already assigned to other LPs with a low load (pre-emption), in this case, from the LP 1-3.

Suppose that bandwidth re-allocation cannot be applied because one or more physical links in the LP's same physical path have no available bandwidth and bandwidth from other LPs sharing the same physical path cannot be utilised either. In such a case, it is proposed to re-route the congested LP through other physical paths in order to have enough available bandwidth to increase it.

Re-allocation is preferable to re-routing because it is less traumatic for the already established connections going through the affected LP.

III. MAS ARCHITECTURE

The complexity of the load balancing mechanisms and the desired objective of achieving a suitable scalability along with the trends of automation and distribution in the network management field, lead us to propose the use of a Multi-Agent System in a completely distributed architecture. Therefore the proposed architecture will have not a global view of the network status and this inevitably provoke that an optimal logical network configuration cannot be guaranteed. However, as the system starts with an already established logical network, which can be optimally designed, it works making the necessary minimal adjustments in order to adapt that topology when an LP becomes congested.

The utilised Software Agents are fixed and distributed over the network elements being managed. This is because this proposal is presented for the management of backbone and core networks, which typically have much bandwidth and they are reliable enough. Therefore it would not be able to take advantage of the especial characteristics of the mobile agents [11].

The proposed architecture is presented in Figure 3. There are two types of agents: The Network Monitoring (M) agents and the Network Performance (P) agents. There is one P agent and several M agents per node.

The M agents are simple reactive agents (lightweight processes) with a stimulus/response type of behaviour, whose main mission is to monitor and control a single LP. The main task of an M agent is to decide when the monitored LP is congested and warn the P agent. There are many ways to make such a decision, which can be from a simple threshold value to a complex deliberation considering many parameters. The proposed monitoring techniques are proposed in [12]. They are called Triggering Functions and are based on statistics and thresholds.

The P agents are more deliberative and each one is responsible of the node where it reside, the outgoing physical links and the supervision and collaboration with the M agents in the same node. The P agent maintains a partial view of the logical network established and is responsible of the co-operation with other P agents. Using their partial network view and co-operating with their neighbours they are able to solve problems beyond the capacity of any one single agent by pooling their abilities.

When an M agent notifies the P agent that an LP is congested, then the P agent must decide what is the appropriate action to upturn the situation. Usually, the agent will check different possibilities and then decide what is the best action to take. In our proposal, P agents have maximised their autonomy and firstly they try to solve the problem using they partial view of the logical network.

Each P agent maintains two types of information from the logical network: a) the information the P agent is sure about, i.e. the information that is directly monitored by itself and its M agents, and b) the information the P agent is not sure about, i.e. information it has received from other P agents in the past and that could be out of date. This information is updated asynchronously, only when there is a problem and P agents communicate with their neighbours, then the partial network view of an affected agent is sent attached with the coordination messages. This idea is displayed in Figure 4. We have chosen not to refresh this information regularly so as to avoid having too many coordination messages, which could affect the scalability. This partial network view must only

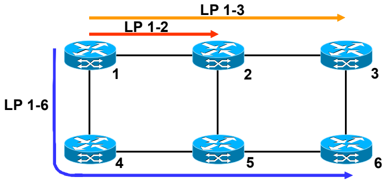


Fig. 2. Initial Situation: LP 1-2 Congested.

contain information strictly necessary for P agents, and must be coded in small size messages. Note that also the P agents' communications are restricted to the physical network neighbours and they have similarities with the signalling mechanisms.

In an decongested scenario there are no co-ordination messages. Only when congestion is detected the affected P agent may send a message (including its partial network view) to one of its neighbours requesting some action. The receiver P agent firstly merges the partial network view with its own, and then it uses it to make a decision or take action.

Therefore the agents must make their decisions with partial, incomplete and/or out-of-date information. The main goal of each P agent is to achieve the maximum utilisation of the resources it manages. This objective is equivalent to minimising the CBP for all the LPs that start in a particular node. P agents can also co-operate with each other.

IV. IMPLEMENTED ALGORITHMS

In order to try to achieve the P agent's objective of load balancing, three different bandwidth reallocation algorithms are proposed. We call them Free-Bandwidth-Only (FBO), First-Node-Only (FNO), and Any-LP (ALP). These algorithms can be combined with a re-routing algorithm. A simplified flow diagram is presented in Figure 5. Note that this figure only gives an idea of the algorithms (what is done in every P agent) and that in the final states it is performed several functions (e.g. send response messages, etc).

Free-Bandwidth-Only (FBO)

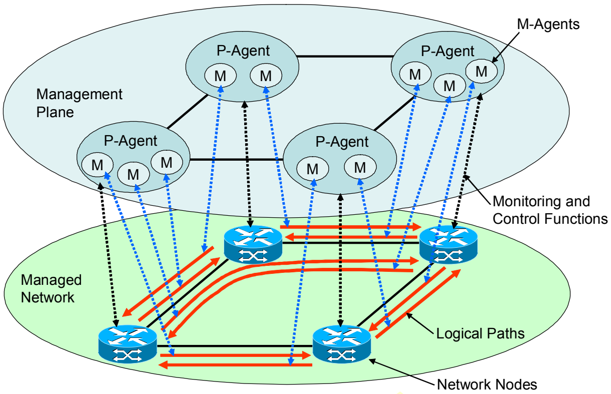
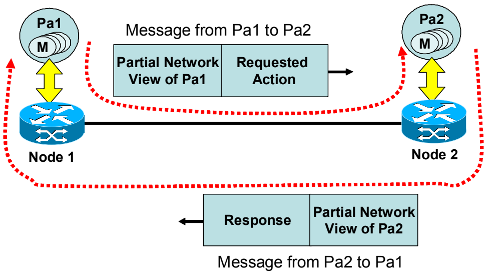


Fig. 3. Performance and Monitoring Agents Distribution.



Partial Network View of Pa1	Requested Action

Response	Partial Network View of Pa2

Fig. 4. Partial Network Views attached to the P agents' Messages

In this case the congested LP can only be increased by means of the free bandwidth in the link. If the congested LP spans over several links there must be enough free bandwidth in all the links. Therefore the only task of the P agents is to check whether there is free bandwidth in the physical links the LP goes through. Therefore, the first P agent (at the origin of the LP) sends a message downstream asking for an LP to be increased and each P agent implied in the process makes a check. If there is enough free bandwidth it is pre-reserved and the message continues downstream to the next P agent. When the last P agent is reached with success then a positive response message is returned upstream, otherwise a negative response is sent. When the positive message arrives to the origin P agent, then the LP capacity is effectively increased.

First-Node-Only (FNO)

This algorithm is a minor modification of the previous one, the FBO. In this case, in the node origin of the congested LP the free bandwidth in the link is checked, and in the case that there is enough free bandwidth to enlarge the congested LP, the FNO's behaviour is exactly the same as the FBO algorithm. In the case that there is not enough free bandwidth in the first link, it is possible to look for other LPs and check if one of them is being underused in order to reduce their bandwidth, and assign it to the congested LP (pre-emption). In this algorithm, the LP candidates must begin in that same node and go through the same first physical link as the congested LP.

Any-LP (ALP)

In this case, in all the nodes the congested LP traverses, it is possible to assign free bandwidth from the links and also unused bandwidth from any other LPs coinciding in one or more LP spans with the congested LP, independent of their initial node. In a given node of the congested LP route, the algorithm first checks the possibility of using free bandwidth from the link; if this is not possible it checks the possibility of using underused bandwidth assigned to LPs beginning at that particular node; if this second option is also not possible, the algorithm checks other LPs not beginning in that node. The procedure consists in asking the P agent in the initial node of the selected LP, for a bandwidth reduction.

Re-routing

It is important to note that whichever algorithm is used it is possible that the attempt to increase a congested LP can finish without success. In that case it is possible to try to reroute the congested LP through another physical path with enough free bandwidth to enlarge it.

Then the P agent initiates the rerouting mechanism by pre-reserving in the first link the required bandwidth for the rerouted LP, and sending a Reroute message to the next P agent of the selected route. The P agent that receives a Reroute message acts as follows. First it updates its partial network view as usual, and then it recalculates the rest of the route using its up to date partial network view. This means that the route is recalculated at each node until the destination node is reached. On this calculation it is possible to utilise any of the existing constraint based routing algorithms. On the way back the rerouted LP is established.

The final path of the rerouted LP can partially coincide with the original congested LP route. The LP spans not used any more are released.

It could also happen that the reroute process reaches a node belonging to the new route where there is no possibility of continuing on to the destination node, i.e. the routing algorithm returns void. In such a case, the re-routing algorithm may consider alternative routes.

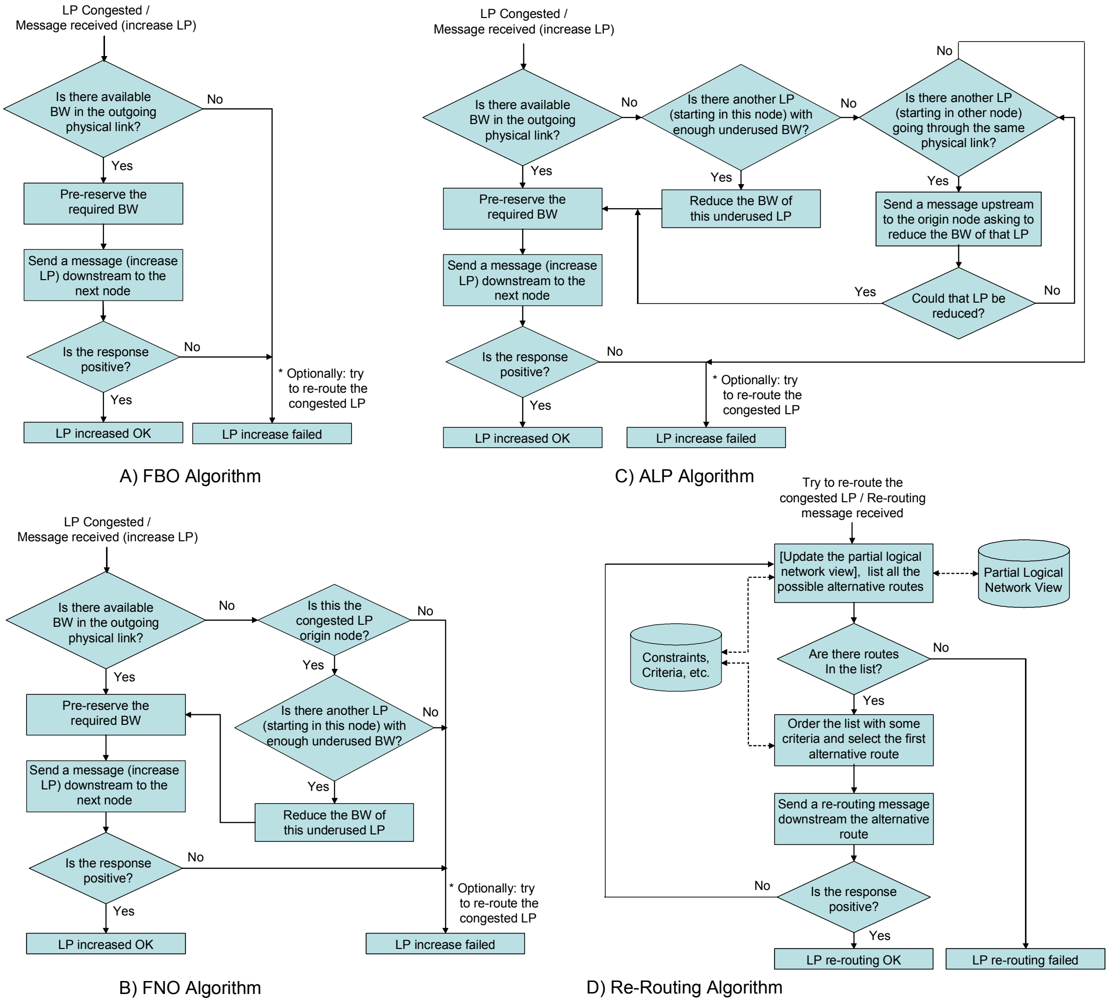
V. EXPERIMENTS AND RESULTS

This section presents two different experiments. The first one compares the three proposed bandwidth reallocation algorithms, and the second one compares the use of the ALP algorithm with and without the re-routing option. The experiments have been carried out using a distributed simulator [13].

In both experiments the same scenario is used (Figure 6). All the physical links have the same capacity (166 Mbps). There are 34 established LPs between edge nodes (there are 7 edge nodes and 8 core nodes), and they all originally have 20 Mbps of assigned capacity. In both cases the simulated time period is of one hour, and in order to test the load balancing algorithms the initial offered load of several LPs changes at about 1/3 of the simulation and changes again at about 2/3 of the simulation. Therefore the bandwidth reallocation algorithms and the re-routing algorithms must adapt the logical network to the new situation. Note that in the first experiment there is a higher offered load than in the second experiment.

Results for the first experiment are displayed in Figure 7. Clearly the ALP algorithm, which considers more options when trying to increase an LP, performs better than the FBO and FNO algorithms, as expected. However the cost of using the ALP algorithm is that the P agents send more co-ordination messages than in the other two cases.

Results for the second experiment are displayed in Figure 8 and Table I. In the case of rerouting there were between 4


LP Congested / Message received (increase LP)
Is there available BW in the outgoing physical link?
No
Yes
Pre-reserve the required BW
Send a message (increase LP) downstream to the next node
Is the response positive?
No
Yes
* Optionally: try to re-route the congested LP
LP increased OK
LP increase failed
LP Congested / Message received (increase LP)
Is there available BW in the outgoing physical link?
No
Is there another LP (starting in this node) with enough underused BW?
No
Is there another LP (starting in other node) going through the same physical link?
No
Yes
Pre-reserve the required BW
Yes
Reduce the BW of this underused LP
Yes
Send a message upstream to the origin node asking to reduce the BW of that LP
Send a message (increase LP) downstream to the next node
Yes
Could that LP be reduced?
No
Is the response positive?
No
* Optionally: try to re-route the congested LP
Yes
LP increased OK
LP increase failed
LP Congested / Message received (increase LP)
Is there available BW in the outgoing physical link?
No
Is this the congested LP origin node?
No
Yes
Yes
Pre-reserve the required BW
Is there another LP (starting in this node) with enough underused BW?
No
Send a message (increase LP) downstream to the next node
Yes
Reduce the BW of this underused LP
Is the response positive?
No
Yes
* Optionally: try to re-route the congested LP
LP increased OK
LP increase failed
Try to re-route the congested LP / Re-routing message received
[Update the partial logical network view], list all the possible alternative routes
Partial Logical Network View
Constraints, Criteria, etc.
Are there routes In the list?
No
Yes
Order the list with some criteria and select the first alternative route
Send a re-routing message downstream the alternative route
No
Is the response positive?
Yes
LP re-routing OK
LP re-routing failed

A) FBO Algorithm

C) ALP Algorithm

B) FNO Algorithm

D) Re-Routing Algorithm

Fig. 5. Bandwidth Reallocation and Re-Routing Algorithms

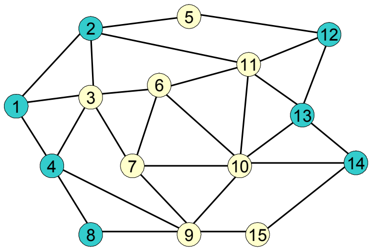


Fig. 6. Simulated network

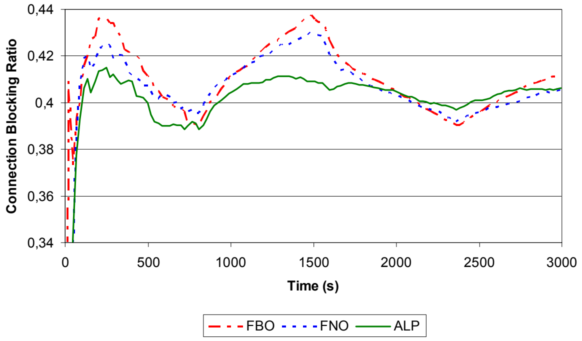


Fig. 7. Comparison of bandwidth reallocation algorithms

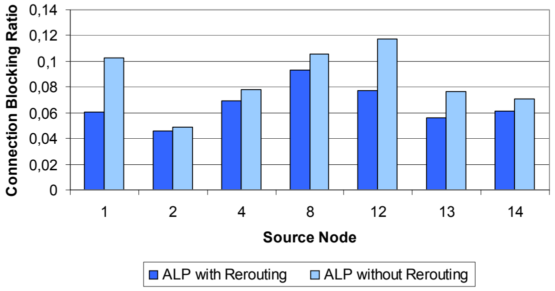


Fig. 8. ALP with and without re-routing

TABLE I
NUMBER OF MESSAGES BETWEEN P AGENTS

Case	Total number of messages	Max number of messages sent by a node	Min number of messages sent by a node
With re-routing	1858	274	23
Without re-routing	3814	758	3

and 7 LPs rerouted at the end of the simulation time. There are many LPs that do not change their offered load and the offered load changes occur at around 1/3 and 2/3 of the way through the simulation, this means that the cases of re-routed LPs are in the second half of the simulation. This fact, along with the offered load not being extremely high, means that there is a small but significant difference between the re-routing case and the case without re-routing. In order to present the results, the LPs have been grouped according to their origin node and the rejection ratio per node is presented (see Figure 8). There are fewer messages sent by P agents because when a LP is congested and there is no possibility of increasing it using the bandwidth reallocation algorithm, the P agents keep trying and sending messages in order to find an LP that can reduce its capacity.

VI. CONCLUSIONS

This paper introduced a Multi-Agent System capable of carrying out automatic, dynamic and distributed load balancing by making use of the logical network concept.

The results show the ability of the proposed MAS to carry out the load balancing and the network is gradually adapted when there is an unforeseen change in the offered load. Moreover, the proposed MAS achieves good scalability and robustness due to the distribution and autonomy of the P agents.

ACKNOWLEDGMENT

This work was partially supported by the Spanish Ministry of Science and Technology (MCyT) under contract TIC2003-05567

REFERENCES

[1] Armitage G., “MPLS: The Magic Behind the Myths”, IEEE Communications Magazine, January 2000.

[2] Eric Mannie (Ed), “Generalized Multi-Protocol Label Switching Architecture”, Work in Progress - Internet Draft, May 2003.

[3] Othmar Kyas, “ATM networks”, International Thomson Computer Press, 1995, ISBN 1-85032-128-0

[4] Ken-ichi Sato, Satoru Ohta, Ikuo Tokizawa, “Broad-Band ATM Network Architecture Based on Virtual Paths”, IEEE Transactions on Communications, vol 38 no 8, August 1990

[5] Z. Dziong, Y. Xiong, L.G. Mason, “Virtual Network Concept and its applications for resource management in ATM based networks”, International IFIP/IEEE Conference on Broadband Communications, Chapman & Hall, 1996.

[6] T-H. Cheng, Y-K. Sze, C-W. Tan, “A heuristic algorithm for allocating virtual path bandwidth in an ATM network”, Computer Communications vol.22 no.9, 1999.

[7] Pere Vilà, Josep L. Marzo, Antonio Bueno, “Automated Network Management Using a Hybrid Multi-Agent System”, In Artificial Intelligence and Applications, September 9-12, 2002. Málaga, Spain.

[8] S. Albayrak (Ed.), “Intelligent agents for telecommunication applications”, Lecture Notes on artificial intelligence vol.1699, Springer-Verlag 1999, ISBN 3-540-66539-0

[9] A.L.G. Hayzelden, R.A. Bourne (Eds.), “Agent technology for communications infrastructure”, John Wiley & Sons Ltd. 2001, ISBN 0-471-49815-7.

[10] Friesen V.J., Harms J.J., Wong J.W., “Resource Management with Virtual Paths in ATM networks”, IEEE Network, vol 10 no 5, September/October 1996.

[11] C. Bohoris, G. Pavlou, H. Cruickshank, “Using mobile agents for network performance management”, Network Operations and Management Symposium (NOMS). 10-14 April 2000.

[12] P. Vilà, J.L. Marzo, E. Calle, L. Carrillo, “Lightweight Monitoring of Label Switched Paths for Bandwidth Management”, Accepted in IEEE Symposium on Computers and Communications (ISCC'04), Alexandria (Egypt), June 29 - July 1, 2004.

[13] J.L. Marzo, P. Vilà, L. Fàbrega, D. Massaguer, “A Distributed Simulator for Network Resource Management Investigation”, Computer Communications Journal (Elsevier) 26 pp. 1782-1791, September 2003.