Dynamic Bandwidth Management as part of an Integrated Network Management System based on Distributed Agents

P. Vilà, J.L.M arzo, E. Calle
Institut d'Informàtica i Aplicacions,U niversitat de Girona
Lluís Santaló Av.
17071 Girona,S PAIN
{perev | marzo | eusebi}@eia.udg.es

***Abstract* – In this paper, we present a system for dynamic network resource configuration in environments with bandwidth reservation and path restoration mechanisms. Our focus is on the dynamic bandwidth management results, although the main goal of the system is the integration of the different mechanisms that manage the reserved paths (bandwidth, restoration, and spare capacity planning). The objective is to avoid conflicts between these mechanisms. Thes ystem is able to dynamically manage a logical network such as a Virtual Path network in ATM or a Label Switch Path network in MPLS. This system has been designed to be modular in the sense that in can be activated or deactivated, and it can be applied only in a sub-network. The system design and implementation is based on aM ulti-Agent System (MAS). We also included etails of its architecturea nd implementation.**

I.I NTRODUCTION

Nowadays, telecommunications networks have become an indispensable technology. They are evolving quickly and continuously, however, there is a noticeable lack of powerful and dynamic management tools for resource configuration. This is mainly due to the rapid increase in transmission speeds, and to the increasing number of usersa nd services. Moreover, conventional network management systems perform the management tasks ina . centralised way. This centralised management results in a scalability problem because the network management centre is responsible for collecting and processing all the monitoring data from all the network elements being managed.

The overall objective of networkp roviders is financial profit, hence, in the face of increasing competition, maintaining high network performance means that they have to offer more competitive services. On the other hand, due to the very high network capacities and speeds, fast restoration is required, because large information losses means dissatisfied clients and less profit.

Therefore, there is a need for a dynamic resource management system that can maximise network resource utilisation in addition to fast restoration mechanisms. Most of these management systems were introduced for ATM networks because they have the appropriate reservation mechanisms. In ATM the hierarchy ofV irtual Path (VP)a nd Virtual Channel (VC) allows designers to set up a dynamically configurable logical network [1]. In MPLS this is carried out by means of Label Switch Paths (LSP). In this paper we use the term Logical Path (LP)t o refer to any kind of logical path (e.g.V P, LSP,e tc).

Several dynamic bandwidth management systems have been proposed in the literature, e.g. [2]. These systems are usually based on a centralised optimisation algorithm, which is executed periodically (e.g. every hour) and recalculates the entire logical network using traffic statistics and predictions. This obviously requires some changes in the LP set.

On the other hand,f ast restoration mechanisms have led to the use of backup paths (local, global, etc). When a fault affects a working path the traffic is then switched to the backup path. This also modifies the logical network. It is also important to perform ag ood spare capacity allocation, and there are schemes where the backup paths can share their bandwidth [3].

These proposals directly or indirectly modify the set of LPs and its characteristics, including the backup LPs and the management of the spare capacity in the links. There is a need for co-ordination between these mechanisms andt here are few proposals that consider thei nterference between them. Moreover, these mechanisms come into effect at different time scales (restoration needs to be fast, while bandwidth management can be slower). Such co-ordination is a complex task and several proposals rely on Distributed Artificial Intelligence mechanisms, i.e. Multi-Agent Systems (MAS) [4][5].

Proposals based on MAS usually replace the network control mechanisms and the MAS is responsible for the network operation [6][7]. This results in poor robustness, because if the MAS fails then the whole network fails. Therefore, network providers are not confident about using these systems. Another problem in these systems is their scalability. When then etwork resources and network users grow, the system becomes unmanageable, usually due to too many co-ordination messages.

The main goals ofo ur system are the integration and co-ordination of all the mechanisms that act over the logical network, but with a robust and scalable MAS. The System does not substitute the conventional Network control mechanisms but complements them (the management system can be activated or deactivated). If the MAS fails,t he network

This work was partially supported by the Spanish Research Council (CICYT) under contract TEL-99-0976.

0-7803-7632-3/02/$17.00 ©2002 IEEE

still works in a static way. Scalability is assured by using a distributed view of the network in order to minimise the management communications.

Moreover, the objectives of Traffic Engineering (TE) [8] in an MPLS environment are similar to the objectives ofo ur own system, and there are also proposals to perform TE using MAS [9]. Accordingly, our system could be considered as part of such TE mechanisms.

In Section II we briefly present the tasks the system should carry out and the system objectives. In Section III we describe the MAS architecture and how it works. After that, in section IV, we present the results obtained for a set of bandwidth management experiments. Finally, we give our conclusions and describe the work we plan to do in the future.

II SYSTEM SPECIFICATION

The three main functions, encompassed by our approach, are dynamic bandwidth management, fault restoration, and spare capacity planning.

Bandwidth Management

The typical demands made on a network mean that some parts of it can become under-utilised, and other parts congested. When this occurs, some connections are rejected which could otherwise be accepted if thet raffícl oad were better balanced.

One of the main objectives of bandwidth management is to minimise Call Blocking Probability (CBP), i.e. the probability that a call offered isr ejected due to insufficient capacity being available for the allocation of the new call. Two actions are usually performed for the bandwidth management system: bandwidth re-allocation and path re-routing.

There are four typical cases, which are shown in Fig. 1. (a) If there ise nough spare bandwidth in the link, then the congested LP is expanded using this bandwidth. (b) If there is not enough spare bandwidth and other LPs going through the same link are under-utilised, it is possible to transfer resources from one LP to the other. If (a)a nd (b)f ail, then a re-routing is needed: (c) If the congested LP finds another path with enough resources then it can be re-routed. Otherwise, (d) other LPs may be re-routed through other links in order tof ree enough capacity to expand the congested LP.

Fault Restoration

As networks have to be fault-tolerant, restoration after a failure needs to be fast. The ultimate goal is that customers do not perceive failures. To achieve this fast restoration, pre-planned schemes based on backup paths are used. However, there are several types of backup schemes (see Fig. 2), each one better than the others in particular situations. For this reason, and in order to minimise the required resources for the backup paths, many proposalsm ake use of several of these schemesa t the same time in an hybrid approach [10]. This adds yet more complexity to the management system.

Spare Capacity Planning

Network providers want high revenues. Since bandwidth is an expensive resource, the objective is tom inimise the bandwidth reserved for restoration procedures [11]. In other words, a good spare-capacity planning is essential. The main goal of hybrid restoration mechanisms is to save up spare capacity. It is necessary to establish the desired network protection level, i.e. protect the network against one simultaneous link or node failure. In such a scenario, there is the technique of sharing bandwidth between different backup paths (see Fig. 3).

III ARCHITECTURE AND CHARACTERISTICS

A Multi-Agent System is a good way of improving network management for two main reasons: it is an inherently distributed solution and it introduces artificial intelligence based techniques in ordert o automate some day-to-day tasks of the human network managers.

The main goal of our architecture is to achieve maximum integration with the conventional network mechanisms and protocols, helping these mechanisms to improve the management. Other important objectives are robustness and

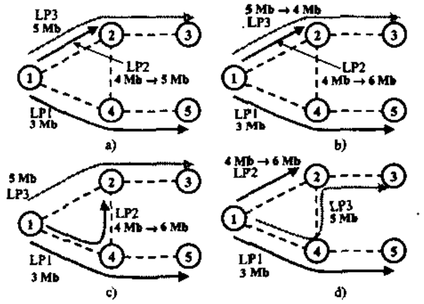

Fig. 1: Bandwidth Management. Initial situation: LP1 , LP2, and LP3 of 3, 4, an 5 Mb respectively. LP2 is congested and needs a bandwidth increase. a) using spare resources. b) using unused resourcesa ssigned to other paths. c) re-routing the congested LPi n order to find the needed resources. d) re-routing of another path in ordert o release the resources needed in the link to expand the congested one.

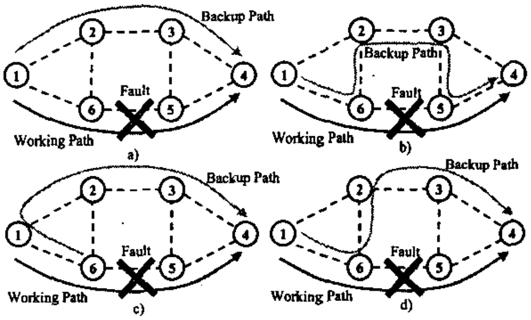

Fig. 2: Different backup mechanisms. a) Global. b) Local. c) Reverse. d) Hybrid-schemes

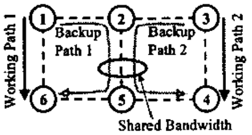

Fig. 3: Spare Capacity optimisation by sharing the bandwidth between backup paths.

scalability. The system itself must be robust in the sense that the network should continue working properly in case of failure of the Multi-Agent System, although it would work in a static way. When the network grows, the Multi-Agent System must not degrade its operation or overwhelm the network with excessive management load.

For these reasons, our system is integrated in a management plane, and it performs a fast but not a real time control. At this level, our system deals exclusively with the logical paths and the management is transparent to the mechanisms that deal with connection or flow control (e.g. Admission Control and routing), which arep erformed with conventional algorithms independently of the logical network management system. In any case, the system can also co-operate with these independent systems.

Our Multi-Agent architecture(Fig. 4) has two different sets ofa gents. First, there is a reactive type of agents whose main task is monitoring and they are called M-Agents (Monitoring-Agents). Second, there is a set of more deliberative agents, which are called P-Agents (Performance-Agents), responsible for deciding the best way to achieve a maximum network performance. This results in a hybrid agent architecture: M-Agents are subordinated to P-Agents, and typically, any actions taken by the M-Agents are under the supervision of the P-Agents. When M-Agents detect a problem they cannot deal with, then P-Agents take control.

M-Agents

There is one M-Agent per unidirectional logical path. Their main responsibilities are monitoring the LP status and detecting any problems (congestion) as well as receiving the alarm notifications when a fault occurs. When congestion is detected, the M-Agent uses its programmed mechanisms to solve it. If the problem cannot solved, the P-Agent is notified. With respect to faults, if the LP is protected by means of a backup path, the M-Agent is responsible for both paths and implements the switchover mechanism.

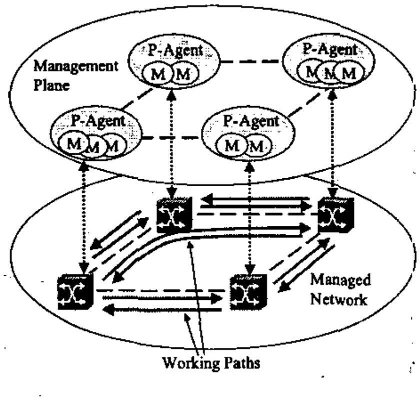

Fig 4: Multi-Agent System Architecture

The M-agents are simple rule-based agents and do not have any world representation model. The number of these agentsc hanges over time according to the establishment or release of LPs.

P-Agents

There is one P-Agent per node and each one is responsible for all the LPs that begin in its node. The idea is that they try to get the maximum performance out of the outgoing physical links, managing and planning logicalp aths, both working and backup. For this reason, each P-Agent maintains a knowledge base with the whole physical network topology and characteristics analysis as well asp artial information on the logical topology (the part it is interested in). When a problem is detected and the P-Agent is required, a need for communication with its neighbours arises. There are two types of communication: co-operation (asking the neighbour for some resources for a already established LP) and negotiation (asking some nodes for the best path to re-route an LP). We are currently evaluating different negotiation mechanisms such as theC ontract Net Protocol and several types of Auctions.

Another task of these agents is the creation and deletion of M-Agents according to the establishment or release of LPs in the network.

To achieve good scalability the number of P-Agents is static while M-Agents are lightweight processes. With respect to inter-agent communications, we apply the constraint that only P-Agentsa re able to communicate outside a node and they can only establish communicationw ith their physical and logical network neighbours. If some of the required information is not available in the neighbourhood, these neighbours can ask successively their own neighbours and so on.

If the logical network continuously changes, a performance degradation isp roduced due to the increase of management traffic. Therefore, the system should control the number of bandwidth re-allocations.

We implemented the MAS usingJ ava as a distributed system. Each P-Agent is an independent process and the M-Agents are threads inside each P-Agent. The communication between P-Agents makes use of the Java RMI functionality. The distributed MAS manages a simulated network [12], which is also a distributed system and each agent is tied to one node by a TCP/IP socket.

IV EXPERIMENTS

The experiments presented here were carried out in order to evaluate different mechanisms for the identification of congested logical paths. These experiments are focused on bandwidth management, for reasons of simplicity the restoration capabilities (backup paths and spare capacity) are disabled. An interesting problem is how to find out if an LP is congested. Each M-Agent periodically performs a monitoring function over a single logical path (a time interval of 10 seconds for these experiments) and it decides whether the LP is congested or not. If the LP is considered to be congested, then the M-Agent triggers the mechanism for increasing the bandwidth of the LP by taking spare resources from the link. If this is not possible, the system tries to allocate unused resources that are already assigned to other LPs (using a preemptive policy). In the proposed scenario theM AS do not makes use of the LP re-routing functions.

The triggering mechanism is called 'Trigger Function' and we have evaluated three different functions:

a) Rejected-5: If the 5l ast consecutive connection or flow requests for a given LP are rejected, the LP is congested.
b) CBP-70: The Call Blocking Probability (CBP) is calculated using the last 30 connection or flow requests. If the CBP is greater or equal to 70%, the LP is congested.
c) Load-90: If the load of the LP is greater or equal to 90% of its capacity,t hen it is congested.

The network simulation for the experiments is depicted in Fig. 5. This network has 4 nodes and 4 bi-directionalp hysical links. Each physical link has 100 Mbps of capacity. There are 10 LPs numbered from 1 to 10 in the figure. Initially each LP has an assigned capacity of 15 Mbps. All LPs have the same offered traffic load (specified in table 1). Using negative exponential distributions for the interarrival time and duration, the mean load for each LP is 100 Mbps, hence all links tend to be congested.

Another point to investigate is the amount of bandwidth by which to increase an LP every time the Trigger function detects congestion. We tested five different, fixed capacities : 0.5, 1, 2, 3 and 4 Mbps. The simulation time was 1 hour in each case and the general behaviour, shown in Fig. 6, was as follows: In the case of a single LP per link, this LP increased its bandwidth up to the maximum level of the link (LP 3 in Fig. 6). In the case of two LPs per link they increased their bandwidth until they reached half of the link capacity and then they competed for bandwidth (LPs 1 and 2 in Fig 6).

We evaluated two parameters for the global network made up of the 10 LPs: the total Call Blocking Probability (CBP) as defined below in (1) and the total throughput defined in (2),

$$CBP = \sum_{i=1}^{nLP} \frac{R_i}{O_i} \qquad (1)$$

$$Throughput = \frac{1}{T}\sum_{i=1}^{nLP}\sum_{j=1}^{nA_i}\left(CA_{ij}\cdot t_{ij}\right) \qquad (2)$$

where:

nLP number of LPs in the network
R_i number of rejected connections for LP_i
O_i number of offered connections for LP_i
nA_i number of accepted connections for LP_i
CA_{ij} assigned capacity for the accepted connection j of the LPi
t_{ij} duration for the connection j of the LP_i
T total simulation time

The results show significant differences between the three trigger functions but less difference in terms of the size of bandwidth variation. The global CBP results are depicted in Fig. 7, while the throughputr esults are shown in Fig.8. It can be seen that the Load-90 Trigger mechanism produces the best results, as we expected (it does notw ait to detect blocked connections), but its main disadvantages are that this Trigger function produces many more attempts of bandwidth changes than the other two. Moreover, there is the possibility that LPs may never use the remaining 10% of theira ssigned capacity. A Load-100 function (100% of LP capacity used) could avoid this second problem, but it would not work well when, for

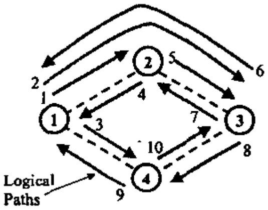

Fig. 5: The simulated network with the established LPs.

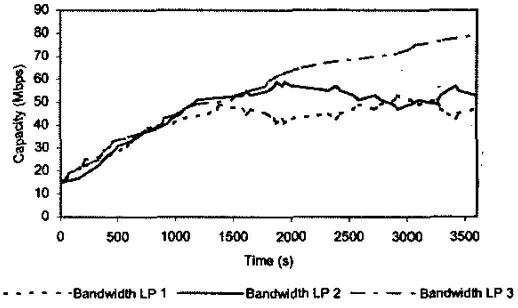

Fig. 6: Capacities assigned to LPs begining in node 1 (LPs 1, 2, and 3), using the Trigger function Reject-5 and every change of bandwidth is of 2Mbps.

TABLE I
GENERATED TRAFFIC FOR EACH LP

Traffic Class	Assigned Bandwidth	Mean Interarrival Time (s)	Mean Duration (s)
1)	64Kbps	2	60
2)	2Mbps	10	120
3)	2Mbps	20	300
4)	4Mbps	30	120
5)	10Mbps	100	300

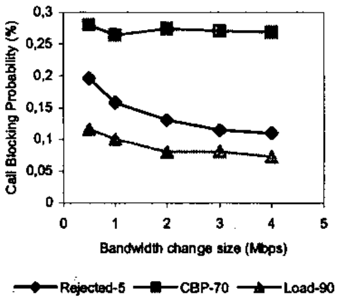

Fig. 7: Call Blocking Probability of the network for the three different Trigger functions and for the 5 different sizes of bandwidth change.

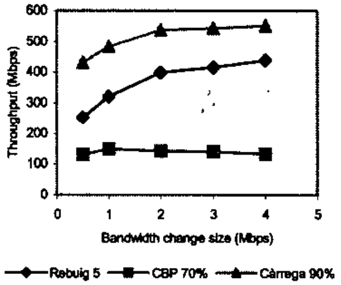

Fig. 8: Network Throughput for the three different Trigger functions and for the 5 different sizes of bandwidth change.

instance, with 1 Mbps of free bandwidth on an LP, thisL P could be rejecting a great number of 2 Mbps or 4 Mbps connections or flows.

On the other hand, the Trigger function CBP-70 performs badly because it produces very few bandwidth changes. This seems to be due to the difficulty in reaching a probability of 70% rejection because of the number of small size connections. This probability value(70%) can be adjusted, butw e think that this Trigger function is too dependent on the type of traffic offered.

The Trigger function we chose for the M-Agents is the Rejected-5 because it performs quite well, it does notp rovoke excessive bandwidth change attempts, and is sufficiently independent of the type of offered traffic.

As for the size of the bandwidth changes, it is clear that small changes (0.5 Mbps)a re worse than biggero nes (Fig. 7 and 8). Moreover, small sizes provoke more bandwidth changes attempts. It is necessary to select a size big enough to avoid having excessive changes, but it cannot be larger than the largest connection or flow size in order to avoid wasting bandwidth.

V.C ONCLUSIONS AND FUTURE WORK

The MAS architecture we present here carries out a dynamic management of the resources (bandwidth), implements the fast restoration mechanism and plans the spare capacity. This is done by means of a logical network. The system complements the conventional mechanisms and it can be enabled and disabled as required. The objective is to integrate and automate the resource management functions in order to maximise the network performance. The work presented here focuses on bandwidth management and how to detect that an LP is congested, because it is a critical point of the system.

In order to determine which one fits our system best, we have analysed three Trigger functions, which wereu sed to determine whether a logical path is congested or not and which activate the bandwidth re-allocation,.

On the other hand, we are studying different heuristics for the M-Agents toc hoose a dynamic size of the bandwidth changes. They should take into account the spare bandwidth on the link, thenu mber of LPs on the same link, and the behaviour of the traffic (has it been increasing very fast or slowly?) in the recent past.

REFERENCES

[1] V.J. Friesen, J.J. Harms, J.W. Wong, "Resource management with virtual paths in ATM networks", IEEE Network vol.10 no.5,1 996.

[2] T-H. Cheng, Y-K. Sze, C-W. Tan, "A heuristic algorithm for allocating virtual path bandwidth ina nA TM network", Computer Communications vol.22 no.9,1 999.

[3] B.G. Józsa, D. Orincsay, "Shared backup path optimization in telecommunication networks", 3rd International Workshop on Design of Reliable Communication Networks, DRCN, Budapest (Hungary), 2001.

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

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

[6] F.Somers, R. Evans, D. Kerr, D. O'Sullivan, "Scalable low-latency network management using intelligent agents", XVI World Telecom Congress ISS, Toronto (Canada),1 997.

[7] Z. Luo, J.Bigham, L.G. Cuthbert, A.L.G. Hayzelden, "Traffic control and resource management using a multi-agent system", 5th Int. Conf. on Broadband Communications,H ong Kong (China), Nov 1999.

[8] X. Xiao, A. Hannan, B. Bailey, L.M. Ni, "Traffic Engineering with MPLS in the Internet", IEEE Network Magazine,M arch 2000.

[9] G. Osinaike, R. Bourne, C. Phillips, "Agent-based dynamic configuration of differentiated traffic using MPLS with CR-LDP signalling", 17th UK Teletraffic Symposium UKTS 2001, Dublin (Ireland).

[10] E. Calle, T. Jové, P. Vilà, J.L. Marzo, "A Dynamic Multilevel MPLS Protection Domain", 3rd InternationalW orkshop on Design of Reliable Communication Networks, DRCN,B udapest (Hungary), 2001.

[11] Y. Xiong, L.G. Mason, "Restoration strategies and spare capacity requirementsi n self-healing ATM networks", IEEE/ACM Trans. on Networking vol.7 no.1,1 999

[12] J.L. Marzo, P. Vilà, L. Fàbrega, D. Massaguer, "An ATM Distributed Simulator for Network Management Research", 34th Annual Simulation Symposium , ASS, Seattle (USA), 2001, IEEE Computer Society ISBN 0-7695-1092-2.