Using a multi-agent system for network management[1]

Pere Vilà†, Josep L. Marzo†, Lluís Fàbrega†

† Institut d'Informàtica i Aplicacions
Universitat de Girona
Avinguda Lluís Santaló s/n
17071 Girona
{perev, marzo, fabrega}@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. The objective is to avoid conflicts between the mechanisms that make changes over the same resources (i.e. bandwidth, restoration, and spare capacity planning) enabling its integration. The system is able to dynamically manage a logical network such as a Virtual Path network in ATM (Asynchronous Transfer Mode) or a Label Switch Path network in MPLS (Multi-Protocol Label Switching). 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 a Multi-Agent System (MAS). This is an hybrid MAS where reactive agents take charge of monitoring and time-constrained decisions, and a more deliberative layer take care of situations that the reactive one cannot solve. We also include details of its architecture and implementation.

Keywords: distributed AI, multi-agent system, network management, resource management.

1 Introduction

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 users and services. Moreover, conventional network management systems perform the management tasks in a 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 network providers 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 of Virtual Path (VP) and 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) to refer to any kind of logical path (e.g. VP, LSP, etc).

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. The logical network is modified using these results.

On the other hand, fast 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 a good 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

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

LPs and the management of the spare capacity in the links. There is a need for co-ordination between these mechanisms and there are few proposals that consider the interference 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 the network resources and network users grow, the system becomes unmanageable, usually due to too many co-ordination messages.

The main goals of our 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, the network 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 of our 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 2 we briefly present the tasks the system should carry out and the system objectives. In Section 3 we describe the MAS architecture and how it works. After that, in section 4, 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.

2 System specification

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

2.1 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 the traffic load 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 is rejected 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.

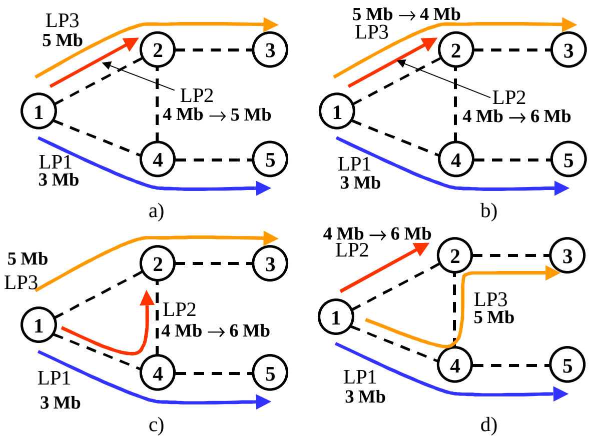

Fig. 1: Bandwidth Management. Initial situation: LP1 , LP2, and LP3 of 3, 4, an 5 Mb respectively. LP2 is congested and needs a bandwidth increase. a) using spare resources. b) using unused resources assigned to other paths. c) re-routing the congested LP in order to find the needed resources. d) re-routing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

There are four typical cases, which are shown in Fig. 1. (a) If there is enough 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) and (b) fail, 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 to free enough capacity to expand the congested LP.

2.2 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 proposals make use of several of these schemes at the same time in an hybrid approach [10]. This adds yet more complexity to the management system.

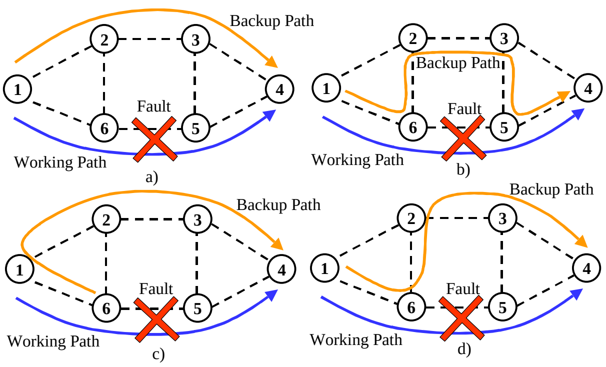

Fig. 2: Different backup mechanisms. a) Global. b) Local. c) Reverse. d) Hybrid-schemes

2.3 Spare Capacity Planning

Network providers want high revenues. Since bandwidth is an expensive resource, the objective is to minimise 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).

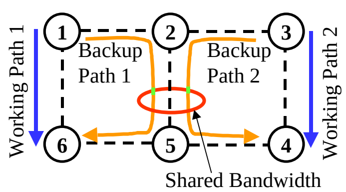

Fig. 3: Spare Capacity optimisation by sharing the bandwidth between backup paths.

3 System Characteristics and Architecture

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 order to 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 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 are performed 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 of agents. 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.

3.1 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.

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

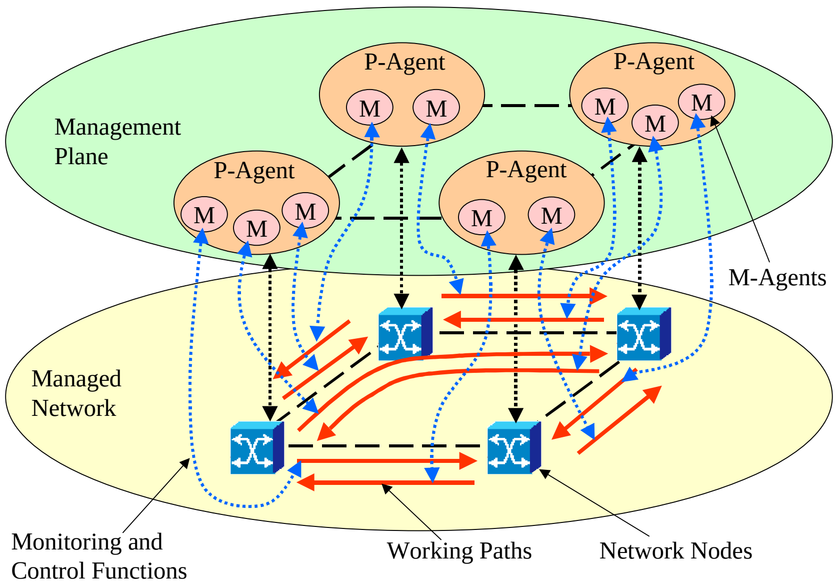

Fig.4.: Multi-Agent System Architecture

3.2 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 logical paths, 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 as partial 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 the Contract 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-Agents are able to communicate outside a node and they can only establish communication with 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 is produced due to the increase of management traffic. Therefore, the system should control the number of bandwidth re-allocations.

4 Implementation and experiments

We implemented the MAS using Java 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 where each agent is tied to one node by a TCP/IP socket. Therefore both the network simulator and the MAS are distributed systems and are able to be executed using several computers in a LAN. The network simulator was implemented as independent as possible from the MAS. The objective of such a design is to simplify a possible adaptation of the MAS to a real network.

The experiments presented here are focused on bandwidth management, because this illustrates better the agents actions and for reasons of the restoration capabilities (backup paths and spare capacity) are disabled. Each M-Agent periodically performs a monitoring function over a single logical path (a time interval of 10 seconds in 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 pre-emptive policy). In the proposed scenario the MAS do not utilise the LP re-routing functions.

In order to detect whether a LP is congested or not the M-agents use a triggering function. This function can be adjusted (or even changed) for the P-agents. The triggering function used in these experiments is the called “Rejected-5”. This means that if the 5 last consecutive connection or flow requests for a given LP are rejected, the LP is considered congested.

The network simulation for the experiments is depicted in Fig. 5. This network has 4 nodes and 4 bi-directional physical 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.

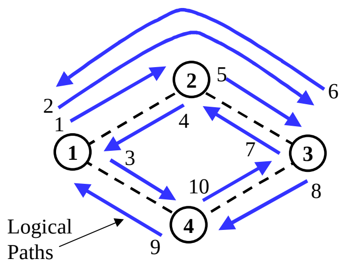

Fig. 5: The simulated network with the established LPs.

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

Table 1: Generated traffic for each LP

Another decision to take is the amount of bandwidth by which to increase an LP every time the trigger function detects congestion. In these experiments we used a fixed capacity of 2Mbps. 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).

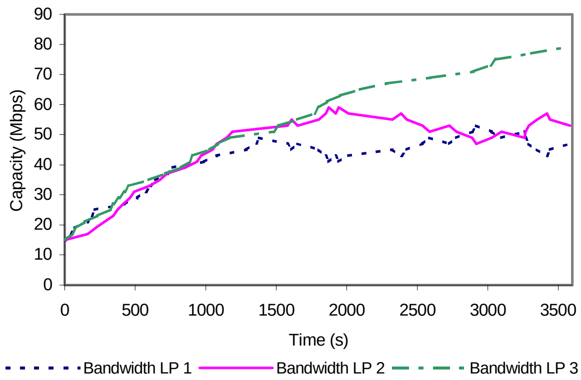

Fig. 6: Capacities assigned to LPs begining in node 1 (LPs 1, 2, and 3), using the Trigger function Reject-5 and every change of bandwidth is of 2Mbps.

On the other hand figure 7 shows some examples of interactions between agents. Most of the interactions are intra-node and only P-agents are able to communicate with other nodes.

5 Conclusions 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 using 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.

This system demonstrates once more the ability of MAS to perform network management functions. Main differences with other systems are that our system does not substitute the network control but complements it. Thus it brings an increase in robustness. The system is also designed to achieve a good scalability. The experiments performed until now demonstrates that the system can carry out the network resource management improving the performance of the network.

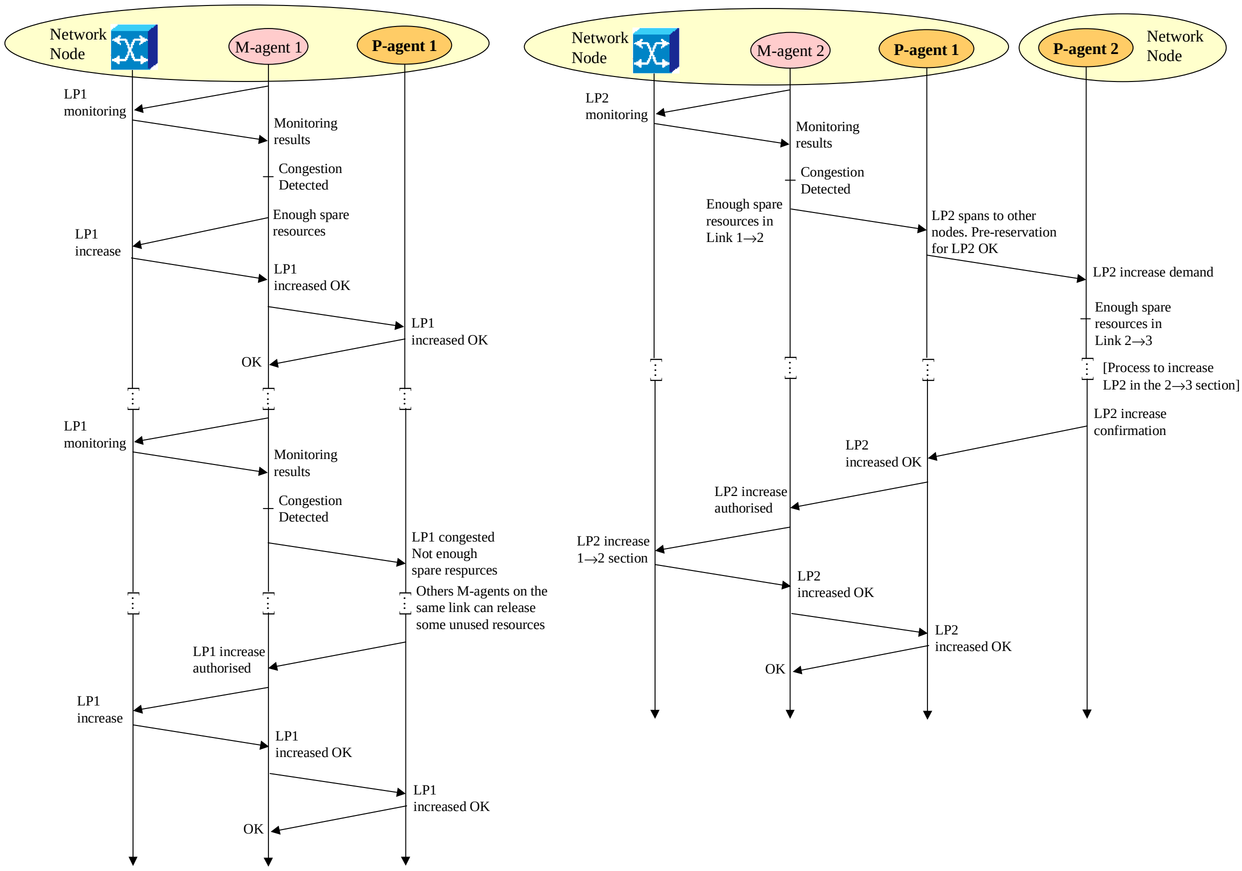

Fig. 7: Example of M-agents and P-agents interactions.

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

Acknowledgement

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

References

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

[2] Calle E., Jové T., Vilà P., Marzo J.L., “A Dynamic Multilevel MPLS Protection Domain”, 3rd International Workshop on Design of Reliable Communication Networks, DRCN, Budapest (Hungary), 2001. [10]

[3] Cheng T-H., Sze Y-K., Tan C-W., “A heuristic algorithm for allocating virtual path bandwidth in an ATM network”, Computer Communications vol.22 no.9, 1999. [2]

[4] Friesen V.J, Harms J.J., Wong J.W., “Resource management with virtual paths in ATM networks”, IEEE Network vol.10 no.5, 1996. [1]

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

[6] Józsa B.G., Orincsay D., “Shared backup path optimization in telecommunication networks”, 3rd International Workshop on Design of Reliable

Communication Networks, DRCN, Budapest (Hungary), 2001. [3]

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

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

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

[10] Somers F., Evans R., Kerr D., O’Sullivan D., “Scalable low-latency network management using intelligent agents”, XVI World Telecom Congress ISS, Toronto (Canada), 1997. [6]

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

[12] Xiong Y., Mason L.G., “Restoration strategies and spare capacity requirements in self-healing ATM networks”, IEEE/ACM Trans. on Networking vol.7 no.1, 1999[11]