
Pere Vilà, José L. Marzo, Antonio Bueno, Eusebi Calle, Lluis Fàbrega
Institut d’Informàtica i Aplicacions - Universitat de Girona
Campus de Montilivi, 17071 Girona (Spain)
phone: +34 972 418 890 fax: +34 972 418 098
e-mail: {pere | marzo | bueno | eusebi | fabrega}@eia.udg.es

Keywords: Multi-Agent System, Network Management, Scalability, Resource Management, Distributed Systems.

Abstract
This paper introduces a distributed architecture based on Software Agents, which is devoted to dynamic network resource management. Among others, one of the main objectives of that architecture is to achieve a suitable scalability. This paper presents a scalability study of the proposed Multi-Agent System (MAS) using a mathematical model which has been developed from our experience in the system design and implementation, and also analysing the many simulation results obtained. The MAS results to scale well enough for a wide range of network sizes.

1. INTRODUCTION AND BACKGROUND
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, usually based on standards, which provide network statistics and analysis tools to help human network managers in decision-making, manual operation, and programmed periodic readjustments. These centralised decision-making could suffer a scalability problem when excessive network monitoring traffic arises.

Typically the main resource to be managed at this level is the 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. Then, the user connections are established through these LPs. This is the concept of Virtual or Logical Network (Figure 1). The LPs are pre-reservations of bandwidth between different nodes in order to facilitate the establishment of user connections. Main characteristic of this Virtual 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 [1][2].

![Logical Network Concept](image)

The Logical Network can be calculated in an optimisation process given the physical topology, the predicted traffic demands and the performance constraints. 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 Virtual 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 Virtual Network topology. However, this brings the problem that the adaptation does not possibly happen when there are congestion problems in the network.

Multi-Protocol Label Switching (MPLS) [3][4], Generalised MPLS (GMPLS) [5][6], and Asynchronous Transfer Mode (ATM) [7][8] are examples of high speed core network technologies capable of establishing LPs.

The proposed system [9] 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 virtual network concept. These functions include load balancing, fault protection and spare capacity management.

1.1. Network resource management

Therefore the logical network can be dynamically changed and this implies that the network resources can be adapted to the traffic demand to obtain the maximum performance (and profit) from the available physical resources. There are several mechanisms that can modify this logical network. When these mechanisms can act dynamically at any time the need of co-ordination arises. It is necessary to introduce briefly the mechanisms that are integrated in the proposed architecture.

Load balancing

Load balancing attempts to manage the bandwidths assigned to the LPs, i.e. change their bandwidth in order to better adapt the logical network to the offered traffic. When due to unforeseen changes in the offered load some LPs can become underutilised and others congested, i.e. the LP bandwidth is already assigned to several connections and new requests are being rejected, which could be accepted if the traffic loads were better balanced. This is equivalent to the minimisation of the Connection Blocking Probability (CBP).

There are two actions usually performed for the bandwidth management systems in order to increase the bandwidth of a congested LP: re-allocation of bandwidth (using available bandwidth in the link, or using underused bandwidth from other LPs – pre-emption) and re-routing of LPs (the congested LP through an alternative less-congested path) [10]. It is preferable re-allocation than re-routing because it is less traumatic for the already established connections going through the congested LP.

Failure protection

The ultimate goal of the protection mechanisms is that customers do not notice failures. High speed networks require the utilisation of fast restoration mechanisms that minimise the traffic losses. Fast restoration is achieved by using pre-planned mechanisms, i.e. the use of backup LPs [11] [12] [13]. A working LP can hence be protected using a backup LP (Figure 2). As soon as a fault affecting a working LP is detected, the connections going through it are rapidly switched over the backup LP with a minimum packet loss. There are several backup techniques, and each one has its advantages and drawbacks [14]. The management of backup LPs is usually done by similar means as the working LPs.

Spare capacity management

Network providers want high revenues. Since bandwidth is an expensive resource, the objective is to minimize the bandwidth reserved for restoration procedures [15] [16] [17]. Therefore, besides the possible decision to protect only the high priority LPs, there is another decision to make: whether to protect the LPs against a single simultaneous failure or to protect them against multiple simultaneous failures. This allows sharing the bandwidth reserved for the backups. The spare capacity reserved for the backup LPs must also be controlled and managed.

![Figure 2: Working and Backup LPs.](image)

Resource management complexity

These three management functions are closely related each other. In the case that bandwidth re-allocation (increase or decrease) or rerouting is performed on an LP, then there could be interferences with the backup LP and the spare capacities reserved in case of failure. Other interferences can occur during the course of failures when the backup LPs are in use. All these possible interferences should be taken into account.

In an ideal world, the perfect load balancing would be done by calculating the optimal logical network (working LPs, backup LPs, and spare capacities) and updating it for every new established connection. This is not possible because of two reasons. First, it is not possible to calculate the optimal Logical Network fast enough since to do that a centralised algorithm is needed. Therefore, this centralised algorithm would need an up-to-date network status view. Second, there would be such a number of bandwidth re-adjustments, re-routings, etc. that a huge amount of management traffic would be generated, reducing instead of increasing the network performance. In order to avoid these problems there must be found a trade-off between the number of readjustments of the logical network and the distance to the optimal configuration.

For these reasons, among others, nowadays network providers typically utilise periodic adjustments (hourly / daily / weekly). These periodic adjustments, however, have the drawback that they cannot cope with short unpredictable changes in the offered traffic load. Therefore there is a need for an automated logical network management which can act when a problem arises. However, the proposal must not provoke performance degradation due to too many changes in the logical network.

1.2. Scalability

Scalability is a well-studied aspect in the fields of parallel computing and programming. It is also studied from
the perspective of distributed systems, e.g. client-server, but these studies usually refer to generic distributed systems or to particular services such as the Web. However, Network Management systems are different from generic distributed systems and they have special characteristics, for instance they should operate continuously, with high reliability and over several different time scales. Most scalability evaluation methods are based on the instruction count of every function in the system and/or on measures and tests performed in real system implementations. Thus, such evaluation includes both system design and system implementation. We believe that the overall system scalability depends mainly on the system design and architecture, while the implementation details only allow performance tuning and final adjustments.

Finding a scalability metric and comparing different systems is very difficult. In the literature, different scalability definitions are presented:

- [18] assert that ‘a system is scalable if it can be deployed effectively and economically over a range of different sizes, that are suitably defined’. They define a metric based on the relation between the rate of providing valuable services, the quality of those services and the cost of providing those services.

- Another definition is proposed in [19] where scalability is considered as a measure of how the performance changes as the system size is increased. The performance is viewed as the response time, as seen by a user under normal working conditions, coupled with the cost of the system – hardware requirements – per user. The authors relate the scalability study with the system design, analysing several characteristics without giving an overall scalability measure. It is also pointed out that network communication is often the key performance limitation.

- A different definition of scalability is given in [20] where it is defined as “the ability to augment the scope of the system (new functions, new classes, new instances). Network size, configuration and technologies keep changing. Management systems must be able to track and support this evolution in an incremental way without disrupting the service”.

- Other research, such as [21], define scalability for MAS as “the average measure of the degree of performance degradation of individual agents in the society as their environmental loading, caused by an expansion in the size of the society, increases”.

Thus it can be asserted that there is no clear definition of scalability and even less agreement on a method or measure to evaluate it. As a more generic definition that comprises the above definitions it is possible to say that a “network management system is scalable if its performance does not degrade as the system to be managed grows”.

The scalability problems can arise for different causes or a combination of more than one. These causes can be classified in terms of Processing, Data storage and Communication problems. Processing problems arise when there is insufficient capability to execute the required processes in a node. This includes the lack of memory as well as the lack of computation capacity. Data storage problems arise when a large amount of data has to be stored in a node. Communications problems arise when management messages flood the network. If these three values are maintained under a defined threshold while the network size and number of users increase, then the system is scalable.

In [18] and [22], a scalability metric for distributed systems is presented. This metric encompasses logical and physical system aspects (it is based on a ratio between the system throughput and a combination of response time and cost) and is used for the comparison of two system configurations. The ratio of the two measures indicates whether the system is scalable or not.

It is necessary to evaluate the system scalability in several different sizes, for instance configuration 1 and configuration 2 can be compared using the following ratio, called p-scalability:

\[ \Psi_p = \frac{F(\lambda, QoS_2, C_2)}{F(\lambda, QoS_1, C_1)} \]  

where \( \lambda \) is the throughput, \( C \) is the cost and \( QoS \) represents a set of parameters that measures the quality of service perceived by users. \( F \) is a function which evaluates the performance and the economy of the system. When all scaling variables can be expressed as a function of a parameter \( k \), the scaling of the system can be defined by this scaling factor \( k \). The following Figure 3 shows different scalability situations as a function of this scaling factor.

![Figure 3: Different Scalability Behaviours.](image)

This scalability metric cannot be used for direct comparison of different systems. However it is suitable for evaluating the scalability of each system and asserting how well it scales.

1.3. Paper organization

Section 2 details the proposed architecture for the Multi-Agent System (MAS) that is in charge of the above
described network resource management mechanisms. It also details the MAS characteristics and gives an idea of the implementation. Section 3 presents the mathematical model and the scalability study results. Finally, section 4 shows the conclusions and future work.

2. AGENT SYSTEM ARCHITECTURE

The complexity of the proposed functions, the desired objectives, and the trends in the network management 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 a problem arises, i.e. when an LP becomes congested and new connections are being rejected.

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 and it is not worthwhile the added complexity of the mobile agent systems [23].

The proposed architecture is presented in Figure 4. There are two types of agents: The Network Monitoring (M) agents and the Network Performance (P) agents. Of course, all the agents are situated at the nodes where the computing facilities are. 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 and react quickly when an event (connection rejections, faults, etc.) occurs. 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. It is possible to utilise any of the artificial intelligence techniques for decision making, for instance Case Based Reasoning (CBR), Fuzzy Logic, etc. It is even possible to use Learning techniques that show the M agents how to detect that the logical path is congested. The technique used is not the point of our work. In the implementation of our system special Functions called Triggering Functions have been designed in order to monitor and detect the LP congestion [24], based on statistics and thresholds. The M agents are also responsible of the switchover mechanism when a failure alarm is received. The M agents’ rules/actions have assigned a priority level, with the highest priority set to the failure-related ones.

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 (restricted to the physical neighbours). Using this 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.

Figure 4: Performance and Monitoring Agents Distribution.

Figure 5 shows a network node being managed by one P agent and three M agents. Note that there is an M agent for each logical path starting in the node and there are not any M agents for the logical paths ending or going through that node. Note also that if a logical path is protected by a backup path both are the responsibility of the same M agent, which also could perform the switchover. Figure 5 summarises the main tasks for each agent.

Figure 5: Detail of the Agents Placed in one Node.
2.1. Partial network view and autonomy

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 their partial view of the logical network.

Each P agent has a complete scheme of the physical network, and it is supposed that it does not change. 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 6. 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 contain information strictly necessary for P agents, and must be coded in small size messages.

![Figure 6: Partial Network Views attached to the P agents’ Messages](image)

Let us suppose that there are no problems in the logical network and therefore P agents are not sending coordination messages. In such a situation there are no refreshing messages between P agents. Suppose now that in this scenario an M agent detects a congested LP. The affected P agent may send a message to one of its neighbours requesting some action, the message has the P agent’s partial view of the logical network attached to it. The P agent that receives the message firstly merges the partial network view of the P agent sender with its own partial network view and then it uses its actualised partial network view 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 Connection Blocking Probability (CBP) for all the LPs that start in a particular node. In order to achieve their goal, P agents must cooperate with each other. Therefore, communication mechanisms and protocols must be used.

2.2. Communications

Regarding this communication between P agents there are several options. The use of proprietary protocols, the use of standard agent protocols (typically application level protocols with a rich capacity for transferring knowledge), or even it could be possible to make use of the Operations and Maintenance (OAM) -style of packets by the definition of new types of these special packets or cells.

One of the scalability problems in distributed systems is produced when every agent in the system needs to communicate with all the other agents quite often. In order to assure good scalability we propose the constraint that every P agent can only communicate with its physical neighbours. Of course there are advantages and drawbacks of using this restriction. The advantages are that agents have a limited number of “known” physical neighbours and this simplifies the communication management between the P agents, and helps achieve good scalability. The drawbacks are that communication mechanisms are less flexible and, in most cases, slower due to the hop-by-hop message processing by the P agents. On the other hand, this restriction allows the possibility of establishing special LPs between physical neighbouring nodes for management purposes only. Therefore the communication protocols we propose, situated in an application level, have similarities with the signalling ones. The communication restriction along with the simplicity of the decision making mechanisms of our proposed agents mean that the communication richness in this case is not too high. Although it is perfectly possible to use standardised agent protocols, we have chosen to implement simpler communication protocols.

2.3. Implementation and co-ordination

The specific implementation of the network resource management algorithms for the congestion detection, the bandwidth reallocation, the LP re-routing, the failure protection, and the spare capacity management are out of the scope of this article, due to the space limit. This subsection is introduced here in order to give an idea of how the proposed architecture has been implemented and tested. For the coordination of all the mechanisms and to avoid interferences between them, we applied the following rules and constraints:

- The working LP and the backup LP should be increased and decreased accordingly. The decreasing procedure is more or less straight, however in the increasing process the proposed MAS tries to increase both the working LP and the backup LP simultaneously. If one of them or both fail, then the increasing process is aborted.
For the rerouting process we simply decided that only the working LP can be rerouted through a disjoint path of the backup LP. When a failure occurs the bandwidth of the affected LPs and their corresponding backups cannot be increased nor decreased, and the possible congestion alarms are ignored by the P-agent.

In the version of the proposed architecture that has been implemented, we have used the Java language in an object-oriented design paradigm. A P agent has been designed as a stand-alone program. Each P agent is capable of executing the required number of M agents as Java threads. There is no real communication between a P agent and its M agents, because they are part of the same program and share the same variables. The only real communications are between P agents. In this case we used the Java Remote Method Invocation (RMI) in order to implement the required messages and protocols.

The proposed architecture has been tested using a simulated network [25], which was also implemented as a distributed system. The simulator processes communicate with each other using the socket interface following a client/server model. The P agents’ processes also act as client processes for the simulator node processes (which are the servers). The whole set of processes (simulator and management architecture) can be executed in a single computer or distributed in several computers.

3. SCALABILITY EVALUATION

One of the main objectives of the proposed MAS is to achieve good scalability. In this section we present a mathematical model of the interactions of the software agents in our system. The purpose is to evaluate the system’s scalability based on the ideas presented in the background. The proposed MAS exhibits a suitable scalability for a wide range of scales.

In this study, the network administrator is considered to be the user of the proposed dynamic network resource management system. From this point of view, the mathematical model is developed from the starting point given by Formula 1. Function $F$ is also assumed to be the one suggested in [18], presented in Formula 2.

$$F = \frac{\lambda}{C} QoS$$

where $\lambda$ is the throughput, $C$ is the cost and $QoS$ represents a set of parameters that measures the quality of service perceived by the users. Therefore, it is necessary to define these three functions ($\lambda$, $QoS$, and $C$), taking into consideration a scale factor $k$.

In our proposed MAS, the scale factor $k$ should be considered as the number of P and M agents in the network, which is identical to the number of network nodes and the number of LPs respectively. It is also necessary to evaluate the function $F$ with different network sizes and different numbers of LPs. For this purpose we have defined the networks proposed in Figure 7. Note that the scalability factor $k$ is proposed to be the length of the side of the squared network.

![Networks for the Scalability Evaluation](image)

where each network has the following properties:

- number of nodes $n = k^2$
- number of edge nodes $e = 4k - 4$
- number of unidirectional LPs $nLP = e(e-1)$ (this is because we suppose that the edge nodes constitute a full meshed logical network)
- mean length (number of hops) of the LPs: $lLP = (4k^2 - 8k - 4) / (e - 1)$ (the addition of all the lengths of the LPs from one edge node – the origin – to all the other edge nodes, divided by the number of edge nodes except the origin edge node)

The scalability of the MAS basically depends on the amount of processing to be done by each P agent. This mainly depends on the management traffic due to the action of the P agents performing bandwidth readjustments and LP rerouting, and the number of P agents involved. The possible effect due to network failures is considered not to be significant in this study and is omitted. We are also omitting the weight due to the number of M agents in the edge nodes, because they are lightweight processes (threads), which perform simple calculations and they are not active most of the time (this depends on the monitoring period).

Note that all the constant values used in the following definitions are derived from our experience in the implementation and simulation of the proposed MAS. In addition, it is worth saying that the behaviour of the resulting mathematical model mainly depends on the variable values.

The function $C$ represents the amount of management traffic processed by each P agent, and is defined as:
where \( M \) represents the size (bytes) of the representation of a single LP (this depends on the mean LP length plus a fixed part), and it is defined as \( M = 10 + 2 \cdot l_{LP} \). The partial network view is that a P agent has is defined here as the knowledge of the P agent about the LPs that traverse it. Therefore, the factor \( M \cdot n_{LP} / n \) in Formula 3 is the size of the representation of a partial network view, and it is equal to the size of a management message containing a partial network view.

The remaining factors of the numerator \( \alpha \cdot \sum_{LP} (CBP_{LP} \cdot OC_{LP}) \) represent the number of management messages, where \( \alpha \) represents the number of messages needed to solve one congestion situation, \( CBP_{LP} \) is the connection blocking probability for each LP, and \( OC_{LP} \) is the offered connections for each LP. This model considers that all the connection rejections provokes congestion detection and, therefore, a bandwidth reallocation.

The term \( \alpha \) depends on the algorithms used for solving a congestion situation. It is supposed that the MAS uses a bandwidth reallocation algorithm (which can decrease the underused bandwidth of an LP and use it to increase a congested LP) combined with the re-routing mechanism (a congested LP can be rerouted through a path where its assigned bandwidth can be increased). Analysing the simulation results, the possible situations are divided into three groups (probabilities \( p_1, p_2 \) and \( p_3 \)). In group 1, it is supposed that the congestion situation is solved using available bandwidth from the physical links and only 2 messages are used. In group 2, it is supposed that the congestion situation is solved by using underused bandwidth from another LP. In this case, it is also considered that all the LPs going through a node are checked. Therefore, the number of messages generated depends on the number of LPs. It is supposed that these LPs are uniformly distributed throughout the network and that every P agent must check \( n_{LP} / n \) LPs. In the third group, it is supposed that an LP re-routing is needed and that the messages used in the re-routing procedure are proportional to the mean LP length \( (l_{LP}) \). Therefore \( \alpha \) is defined as follows:

\[
\alpha = 2 \cdot p_1 + \frac{n_{LP}}{n} \cdot p_2 + l_{LP} \cdot p_3
\]  

(4)

The function \( \lambda \) represents the total amount of dispatched connections, and it is defined as:

\[
\lambda = \sum_{LP} (OC_{LP} \cdot (1 - CBP_{LP}))
\]  

(5)

The function \( QoS \) represents the time spent in solving a congestion situation compared with a target time \( t_0 \). It is similar to the function \( QoS \) defined in [18]. However, it is not realistic to expect the same response time in a small network as in a large network, because this is not an on-line service that requires the fastest response time possible in all situations, neither is it a real-time mechanism. Therefore, we considered that the target time is acceptable if the congestion situation is solved in a time limit depending on the LP length. We have defined the target time as a function of the mean LP length \( t_0 = 3 \cdot l_{LP} \). The function \( QoS \) is defined as:

\[
QoS = \frac{1}{1 + \frac{\alpha \cdot l_{LP}}{t_0}} = \frac{3}{3 + \alpha}
\]  

(6)

Therefore the function \( QoS \) just depends on the difficulty (number of messages used) of solving a congestion problem. In order to simplify the model it is supposed that the \( CBP_{LP} \) and \( OC_{LP} \) are the same for all the LPs (\( CBP \) and \( OC \)). In this case, the functions \( C, \lambda, \) and \( QoS \) can be expressed as:

\[
C = \frac{M \cdot n_{LP}^2 \cdot \alpha \cdot CBP \cdot OC}{n^2}
\]  

(7)

\[
\lambda = n_{LP} \cdot OC \cdot (1 - CBP)
\]  

(8)

\[
QoS = \frac{3}{3 + \alpha}
\]  

(9)

After the definition of the three functions, their combination using function \( F \) results, after simplification, in the following expression:

\[
F = \frac{\lambda}{C \cdot QoS} = \frac{3 \cdot n^2 \cdot (1 - CBP)}{M \cdot n_{LP} \cdot \alpha \cdot CBP \cdot (3 + \alpha)}
\]  

(10)

Function \( F \) depends on the scalability factor \( k: F(k) \). However, as stated in [18] the scalability of a system should be evaluated comparing different scales of the system (Formula 1). Therefore, function \( F \) is calculated for different scale factors \( k \) starting from \( k=10 \) and every one of them is compared with the previous one. The results are shown in Figure 8.

The results obtained from the proposed mathematical model show that the MAS is not scalable (i.e. when \( k \rightarrow \infty \) then \( \lambda \rightarrow 0 \)), although it shows a suitable scalability using a wide range of scale factor \( k \). This is because the model has a tendency to be very slow when approaching the zero value.
This means that the system operation is fine with large networks (e.g. when $k=100$, there are 10000 nodes and 156420 LPs in the network), however, it seems reasonable that several design improvements could help in the achievement of better scalability. For instance, to reduce the number of transmissions of the partial network view and increase the autonomy of the P agents when making their decisions. It could also be interesting to study if it is possible to define a mechanism that allows the transmission of only the updated information from the last message between two P agents, and not the whole partial network view.

\[ \Psi = f(k) \]

Figure 8: Scalability Evaluation.

4. CONCLUSIONS

This paper has introduced the architecture of a Multi-Agent System devoted to dynamic network resource management by means of a logical network. This system integrates three functions (load balancing, fault restoration and spare capacity management) and carries out its tasks in a complete distributed way. Therefore, the MAS agents are distributed over the network and there is a need for coordinating their actions. Moreover, none of the MAS agents have a global view of the logical network being managed; therefore there is also a need for collaboration between them. An important issue in these kind of distributed systems is the scalability, and in order to evaluate the MAS scalability a mathematical model of the agents’ interactions and co-ordination messages has been proposed. Results form this mathematical model show that the proposed MAS has a suitable scalability over a wide range of network sizes.

5. REFERENCES