Autonomous Agents for Self-Managed MPLS DiffServ-TE domain

Rana Rahim-Amoud[1], Leila Merghem-Boulahia[1], and Dominique Gaiti[1]

[1] Institut Charles DELAUNAY/LM2S, CNRS FRE n° 2848, UTT. 12, rue Marie Curie, BP 2060, 10 010 TROYES CEDEX, FRANCE
{rana.amoud, leila.boulahia, dominique.gaiti}@utt.fr

Abstract. The combination between DiffServ (Differentiated Services) and Multi-Protocol Label Switching (MPLS) presents a very attractive strategy to backbone network service providers. It provides scalable QoS and traffic engineering capabilities. However, the management of such a network is not a simple function and could not be done manually. In fact, it would be much more economic and effective to automatically manage networks. In this paper, we discuss the essential characteristics needed to build an autonomic network. We also propose a novel architecture based on Multi-Agent Systems (MAS) in order to automatically manage an MPLS-DiffServ TE domain. Simulation results are provided to illustrate the efficiency of our proposition.

Keywords: Traffic Engineering, Autonomic MPLS networks management, Multi-Agent Systems.

1 Introduction

In recent years, there has been active research in the field of Multi-Protocol Label Switching (MPLS) and an increasing number of networks are supporting MPLS [3]. One of the most significant applications of MPLS is the traffic engineering (TE) [4]. MPLS-TE enables resource reservation, fault-tolerance and optimization of transmission resources [19]. However, MPLS does not define a new QoS architecture [11] and cannot provide service differentiation by itself. DiffServ (Differentiated Services) [5] defines an architecture for implementing scalable service differentiation in the Internet by defining multiple classes of services. The combination between MPLS and DiffServ allows a differentiation of services and a traffic engineering based on a fast packet switching technology. This network is called MPLS DiffServ-TE.

As networks grow rapidly and traffic conditions change frequently, the management of such a network presents many difficulties and could not be done manually. Therefore, automated management is required to minimize this complexity and to engineer traffic efficiently [9].

In this paper, we discuss the essential characteristics needed to build an autonomic network. We also propose a novel architecture based on Multi-Agent Systems (MAS) in order to automatically manage an MPLS-DiffServ TE domain. A brief description of MPLS and MPLS-DiffServ is presented in the next section. We then study the important aspects needed by a network in order to be autonomic. In section 4, we present the MAS as a solution for communication networks. Then, we propose a novel architecture based on a multi-agent system able to automatically manage MPLS DiffServ-TE domains. In section 6, we provide simulation results and evaluate the performance of our proposition. Conclusion and future work are given in section 7.

2 MPLS

MPLS [23] is a technology that uses labels to forward packets by specifying the Forwarding Equivalence Class (FEC). All packets in such a class receive the same treatment in the domain. MPLS domain contains two types of equipments: LER (Label Edge Router) and LSR (Label Switch Router). I-LSR (Ingress LSR) is the LER which puts the label to an incoming packet and E-LSR (Egress LSR) is the one which removes the label from the outgoing packet to return it to its initial nature. An LSR is a high speed router device in the core of the MPLS network. The path between two LERs is unidirectional and is called LSP (Label Switched Path).

2.1 MPLS-TE

Traffic engineering is used to achieve performance objectives such as optimization of network resources and placement of traffic on particular links [19]. In other terms, MPLS traffic engineering routes traffic flows across a network based on the resources the traffic flow requires and the resources available in the network [21].

Current Interior Gateway Protocols (IGPs) always use the shortest path to forward traffic in order to conserve network resources. However, using shortest path is not always the best choice and it may cause the following problems [25]:

1. When different shortest paths from different sources converge at some links causing congestion on those links.
2. The shortest path between a source and a destination is over-used while a longer path between these two routers is under-used.

TE is needed to avoid these problems by optimizing resource utilization and network performance [25].

2.2 MPLS - DiffServ

In MPLS domain, the classification of incoming packets is done just at the entry of the domain by the edge router (I-LSR), by assigning a particular packet to a particular FEC. Within the domain, there is no reclassification and packets are just switched by LSRs according to labels. In DiffServ domain, the traffic classification is also done by edge routers by setting the DSCP (Differentiated Service Code Point) field. In the core network, there is also no reclassification, routers use the DSCP value in the IP header to select a PHB (Per-Hop Behavior) for the packet and provide the appropriate QoS treatment [12].

The functioning of both MPLS and DiffServ consists of 3 main steps: (1) traffic classification, (2) labeling of packets after classifying them, (3) traffic forwarding for MPLS and routing or switching for DiffServ. In addition, both MPLS and DiffServ are based on aggregation.

The MPLS support of DiffServ is still an open research issue [3]. Currently, there are two solutions [15], the first one is applied to networks that support less than eight PHBs and it uses the 3 Exp (experimental) bits of the MPLS label to determine the PHB. In this case, LSPs are called E-LSPs. The second solution is applied to networks that support more than eight PHBs. In this solution, the PHB is determined from both the label and the Exp bits and LSPs are called L-LSPs. Each solution has its advantages and its disadvantages and the use of one of them depends on the particular application scenario [19].

In our proposition, we are going to consider the second solution by using different LSPs for different classes of traffic. The effect is that the physical network is divided into multiple virtual networks, one per class. These virtual networks may have different topologies and resources [25]. In this case, three virtual MPLS networks are defined for EF, AF and BE classes. The bandwidth set by administrators on each physical link is partitioned among these MPLS virtual networks. This will provide better resource utilization and each DiffServ level can be treated alone.

We have seen that the TE resolves some serious problems and DiffServ is needed to provide a differentiation of services into the MPLS network. However, an automated management of MPLS DiffServ-TE network is needed to reduce the complexity of the management tasks. The model network should be able to be self-configured, self-managed, self-protected and self-organized. In order to build such a network, some essential characteristics are needed. These characteristics are discussed in the next section.

3 Essential Characteristics to Build an Autonomic Network

In order to build an autonomic network, it is necessary to empower it with some essential characteristics. These characteristics are:

3.1 Decentralization

The current network management scenario is always based on a client-server mode. This centralized management provides a better vision of the global network. However, this represents a heavy and non fault tolerant solution. Therefore, we think there is a real need to decentralize the network control. Control decentralization is obtained by allowing network components to be able to decide on actions to undertake. Furthermore, the component can, if necessary, ask for help from a human administrator or another autonomous component for the realization of some tasks.

3.2 Reactivity

The networks environment is very dynamic and is always in evolution. The router must thus be able to choose the most convenient mechanisms according to the current conditions.

3.3 Proactivity

By being reactive, the router has the possibility of being in phase with the events taking place in its environment. However, we should not rely only on the reactivity to control a router. In fact, a router should envisage the actions to be undertaken.

3.4 Sociability (Cooperation)

The guarantee of end-to-end QoS is a classical problem in networking which is still not totally resolved. Indeed, a given packet may cross several networks, belonging to different operators and thus managed by different strategies. Even if we consider that all these networks are endowed with QoS mechanisms, we cannot guarantee end-to-end QoS. In order to do that, the different networks should cooperate between them and reach agreements to satisfy the requirements of each of them.

3.5 Adaptability (Learning)

In order to realize its goals (accepting more traffic from a given customer, etc.), the router has to carry out some plans to be executed. However, the router must be able to self-evaluate these plans in order to improve its operation. In fact, by observing the results of the plans application, the router can evaluate the relevance of these plans and adapt them, if necessary, to meet more effectively its objectives.

In the next section, we demonstrate that all these characteristics needed to build an autonomic network are offered by the multi-agent solution.

4 Multi-Agent Solution

The agents represent a good tool to make networks autonomic. Indeed, a multi-agent system consists of a set of agents which [16] (1) are able to communicate together, (2) possess their own resources, (3) perceive their environment (until a limited degree), (4) have a partial representation of their environment and (5) have a behavior which aims to realize their purposes. Their main characteristics are developed in the following.

4.1 Characteristics of the Agents

The multi-agent systems can constitute a good tool to provide the autonomic scheme by guaranteeing the different characteristics which seem necessary to reach an autonomic behavior. In the following, we demonstrate that all these characteristics are indeed offered by the multi-agent solution:

Decentralization. Multi-agent approach is decentralized by definition. No agent possesses a global vision of the system and the decisions are taken in a totally decentralized way;

Reactivity. One of the basic attributes of an agent is to be situated (situadness, [8]). That is, an agent is a part of an environment. Its decisions are based on what it perceives of this environment and on its current state. This basic characteristic is very important in a context of highly dynamic networks where appropriate decisions must be taken;

Proactivity. An agent can be able to set goals and to realize them by implementing plans, setting up a strategy, starting cooperation with other agents, etc.

Sociability. One of the interesting features of the multi-agent approach is its ability to distribute the intelligence between the different agents composing the system. This implies that an agent can handle some tasks individually but cannot make everything by itself. Many works concerning the concepts of negotiation and cooperation are realized and the research in this field remains very active [24]. The economic theories constituted a good source of inspiration (Contract Net Protocol, auctions, etc.) [13];

Adaptability. In order to provide more flexibility, researchers are interested in the learning in multi-agent systems. A part of the researches is focused on genetic algorithms [7], while the others use the reinforcement learning [14], etc.

After seeing the characteristics provided by MAS which are well suitable to the management of distributed systems, we propose in the next section a solution based on MAS for MPLS networks.

5 Our Proposition

Since the MPLS functioning is based on the use of LSP in order to forward packets, and the MPLS support of DiffServ is also based on the LSP, it seems that the LSP management is the most important need. It includes LSP dimensioning, LSP setup procedure, LSP tear-down procedure, LSP routing, and LSP adaptation for incoming resource requests. In order to effectively control and manage the path of LSPs, one or more attributes can be assigned to each LSP. Such attributes can be Bandwidth, Path attribute, Setup Priority, Holding Priority, Affinity, Adaptability, Resilience, etc. [25].

As agents have to take the convenient decisions into the MPLS domain, so the introduction of these agents will take place into the MPLS decision points. The first step of our research consisted of finding the decision points of the MPLS network which are especially identified on the entry of the domain on the LER routers [22]. An agent will be, as a result, introduced into each LER router in the MPLS domain. In order to control and manage effectively the network and to benefit from the decentralization feature of MAS, we decided to introduce also an agent into each intermediate LSR. Hence, an agent is introduced into each router in the MPLS domain. Actually, each agent is responsible for the router on which it is introduced and for the corresponding interfaces. All these agents form a multi-agent system. These agents interact and communicate together and interact also with the routers and switches in the domain.

The architecture of our agent is shown in Fig.1. The agent includes two entities: the collector entity (CE) and the management entity (ME) which includes, in its turn, two sub entities: the LSP route management entity and the LSP resource management entity. In addition, the architecture contains a Data Base (DB) which is shared between the CE and the ME.

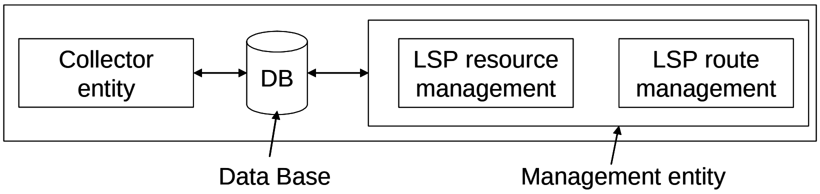

Fig. 1. Our agent architecture

5.1 Collector Entity (CE)

The CEs collect the available bandwidth on physical links and on each LSP. Each CE collects only the information concerning the corresponding router and its interfaces. The CE also collects the network topology information such as the new created LSPs, if an opened LSP is still in use or not, etc. Furthermore, the interaction between agents is done by their CEs by exchanging some of the collected information when necessary. This highlights the social feature of the multi-agent system.

The CE uses SNMP (Simple Network Management Protocol) [10] to collect information from the MIB (Management Information Base) [18] and stores them into the DB. We decided to collect the available bandwidth because we estimate that this is the most important parameter to be treated. It gives us a view of the current network state.

One of the DB tables, called “LSP table”, contains a list of the already created LSPs traversing the corresponding router, their ingress and egress routers, their current available bandwidths and the virtual topologies to which they belong.

5.2 Management Entity (ME)

The ME is an important part of our agent. The ME is responsible for the route and resource management. Precisely, it is responsible for determining when and where an LSP should be created. Indeed, the ME which has access to the DB, uses the stored information to take the appropriate decision. The next step performed by the ME is to automatically implement this decision. The ME contains two sub entities: the LSP route management entity and the LSP resource management entity (Fig. 1).

LSP Route Management Entity. The role of this entity is to route the new LSP on the physical network. More specifically, in case of creating a new LSP, the role of this entity is to decide, for a specific network state, how to select the most suitable route for the LSP avoiding inserting many LSPs on a given link. The idea is to use the routing information generated by a standard IP routing protocol and the local information of the agent in order to reach this purpose. This route will be then used by a signaling protocol such as CR-LDP or RSVP-TE in order to create the LSP.

LSP Resource Management Entity. The role of this entity is to manage the LSP resources. In other terms, this entity has to find the best way to forward the incoming traffic by affecting it to an already created LSP, by re-dimensioning an already created LSP and increasing its allocated resources or by creating a new LSP.

One of the mentioned features of mutli-agent systems is the reactivity. In fact, in order to take the suitable decisions, the multi-agent system has to follow a strategy. In our case, we have proposed a strategy and called it the “LSP creation strategy”. This strategy is described in the next section.

5.3 The LSP Creation Strategy

The general goal of this strategy is to create LSP according to the network conditions. Currently, given the physical topology, the operator has to design a layout or virtual topology by finding an optimal set of paths and a flow distribution over it to accommodate a given demand, as well as to adapt the layout to varying traffic conditions [6].

To design the MPLS layout, there are off-line and on-line proposed approaches. Off-line approaches are based on the estimation of the traffic demand over time. An example of off-line approaches can be the creation of a fully connected MPLS network. This approach consists of creating one or several LSPs between each pair of nodes. This provides a large number of LSPs introducing, as a result, high management cost and complexity. According to Kodialam [17], off-line approaches are not appropriate to MPLS networks due to the high unpredictability of the Internet traffic. Since the off-line approaches present many disadvantages, this solution has to be avoided.

On-line methods calculate paths on demande. Three different on-line approaches can be distinguished: (1) Request-driven, (2) Topology-driven and (3) Traffic-driven.

The request-driven approach is used when MPLS transmits multicast traffic [20]. We do not study this approach because we are not interested in this paper by the multicast case. In the topology-driven approach, a standard IP routing protocol runs and calculates the network's topology. In addition, a Label Distribution Protocol (LDP) constructs a mesh of labeled paths between ingress and egress LERs according to the routing entry generated by the routing protocol [2]. The constructed path is released only if the corresponding routing entry is deleted. In this approach, LSPs already exist before traffic is transmitted. Thus, a constructed path may not be used because the creation of the LSP was based only on the routing information.

In the Traffic-driven approach, the LSP is created according to the traffic information. When a new request arrives, the corresponding path is established and it is maintained until the session becomes inactive. In this approach, only the required LSPs are setup. This approach conserves labels, bandwidth, signaling and management.

It should be noted that the available bandwidth on a physical link is equal to its maximum bandwidth minus the total bandwidth reserved by LSPs crossing it. It does not depend on the actual amount of available bandwidth on that link [25]. The available bandwidth on a physical link is given by the following equation (1):

$$B_a = B_{rt} - \sum_{i=1}^{n} B_i \ . \qquad (\mathbf{1})$$

where B_a is the available bandwidth on the physical link, B_{rt} is its maximum reserved bandwidth, B_i is the bandwidth reserved for the LSP_i and n is the number of LSPs crossing this link.

That means that the establishment of a non used LSP will have bad consequences on the total behavior of the MPLS network. A part of the bandwidth will be reserved without being used. Moreover, another LSP may be prevented from taking a path fault of the lack of the bandwidth. In this context, the traffic-driven technology is more advantageous than the topology-driven one.

The solution, which seems the most logical and the most advantageous to design an MPLS network, is to determine an initial MPLS network topology and to adapt it to the traffic load. Consequently, a topology change will take place when a new LSP is created or released after receiving a real request. Our goal is to decide when to create a new LSP and when to pass a new traffic in an already created LSP. To do that we define the most important factors which can have an influence on the possible decision, these factors are: (1) The requests, (2) The network state and (3) The cost.

A request can be a new bandwidth request, a disabled bandwidth request or a request for tearing-down an existing LSP.

The network state includes the current three virtual topologies such as the created LSP, the existence or not of an LSP between a pair of routers. The network state includes also the LSP attributes (i.e. the available bandwidth, the priority, etc.) and finally, the physical link attributes (i.e. the available bandwidth, the delay, etc.).

The cost includes three different components [3], (1) the signaling cost which is considered only when creating a new LSP or re-dimensioning an LSP. In the other cases, signaling is not needed. (2) The switching cost which depends on the switched bandwidth and the cost defined by the operator. (3) The bandwidth cost which depends on the carried bandwidth and the number of traversed nodes.

Let us discuss the arrival of each type of request. If a request for tearing-down an already created LSP arrives to the entry of the domain. The MAS takes the decision to tear down this LSP and applies it by releasing the corresponding labels and liberating the reserved bandwidth. This information is exchanged between the corresponding agents. As a result, the available bandwidth on the physical link is increased by the value of the liberated bandwidth and all corresponding LSP tables are updated.

Consider now that a request for bandwidth is deactivated. In this case, the MAS takes the corresponding decision and applies it by liberating the reserved bandwidth and increasing the available one of the corresponding LSP. Thus, the available bandwidth on the physical link remains the same and the LSP tables are updated.

Consider that a new bandwidth request arrives between a pair of routers demanding a certain level of QoS. In this case, the first step consists of verifying the existence of an LSP between these two routers in the corresponding virtual topology (EF, AF or BE). This verification is done by consulting the “LSP table”. If an LSP exists, the next step is to compare the available bandwidth of that LSP with the requested one. If the available bandwidth is higher than the requested one, the requested bandwidth is allocated on that LSP and its available bandwidth is reduced accordingly.

If the available bandwidth is lower than the requested one, the multi-agent system verifies the possibility of re-dimensioning the LSP. To do that, the requested bandwidth is compared to the available bandwidth on the physical link (B_a). If the requested bandwidth is lower than or equal to B_a, the multi-agent system decides to increase the capacity of the LSP. In other words, the bandwidth reserved for the LSP in question will be increased by a value equal to the requested bandwidth to be able to forward the new traffic. Consequently, the available bandwidth of the physical link is decreased by the value of the requested bandwidth. If the requested bandwidth is higher than B_a, the multi-agent system eliminates this possibility and verifies the possibility of creating a new LSP on another physical link indicated by the constraint shortest path routing protocol. In this case, the down stream on demand technique is used in order to distribute labels. If no physical link is found, the multi-agent system decides to reject the request.

If there is no LSP between the pair of routers, the multi-agent system reaction will be identical to the one where the requested bandwidth is higher than B_a.

To summarize, our proposition is based on the multi-agent system. Currently, it considers three of the multi-agent systems’ features: the decentralization, the reactivity and the sociability. At now, our solution does not consider the proactivity and the adaptability features. These two features will be treated in our future work.

6 Performance Evaluation

In this section, we demonstrate the performance of our proposition by comparing it to the MPLS-TE solution. We assume that in MPLS-TE solution, the constraint shortest path routing algorithm is used for LSP establishment. In addition, both networks used the traffic-driven approach. As our proposition takes into account the bandwidth parameter and it does not consider the preemption priority, we assume that the LSP preemption is not activated for both networks. The objective of the simulation study is to show that the use of our MAS within the MPLS network provides a smaller number of LSPs introducing, as a result, low control traffic, low signaling cost, low management cost and low management complexity. We evaluate the performance of our proposed solution through extensive simulations on the network showed in Fig. 2 using a Java simulator developed in our laboratory. This network includes 7 edge routers and 3 core routers. We assume that the requests arrive one at a time at the network and only one LSP is allowed to be established per LSP request. The source and destination nodes for the requests are randomly chosen from the set of edge routers. The bandwidth request is uniformly distributed between 100 and 200 units. The number of requests varies from 100 to 1000. We set each physical link capacity to 5000 units of bandwidth which is very restrictive in order to oblige a request blocking and thus evaluate the performance of our strategy.

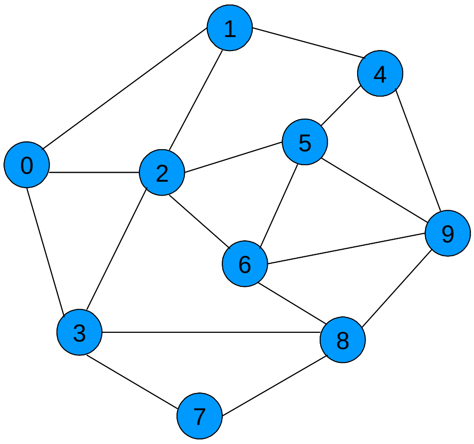

Fig. 2. Network topology

In Fig. 3a, we show the number of LSPs in the network, this number includes the LSPs for the three virtual topologies (EF, AF and BE). We see that when we use the multi-agent system the number of LSPs is lower than that when we do not use the multi-agent system. Thus, by applying our strategy, the number of LSPs is in average reduced by a factor of 2.38 compared to a pure MPLS-TE solution. This improvement in the number of LSPs is due to the re-dimensioning decision taken by our multi-agent system.

The number of LSPs is not the only performance factor that our proposition enhances. Another performance factor is the number of signaling operations. Fig. 3b. shows the number of signaling operations produced in the network. In fact, our strategy reduces the number of signaling operations by a factor of 1.63. This is attributed to the fact that, in many times, the re-dimensioned LSPs have enough space to forward the incoming traffics and there is no need to create a new LSP. In addition, the simulations show that with the increase of requests number, our proposition provides better results.

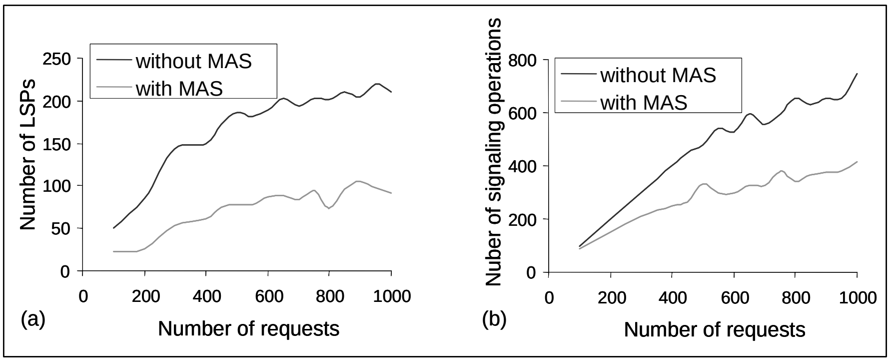

Fig. 3. a) Number of LSPs in the network; b) Number of signaling operations

In order to verify the performance of our proposition, we calculate also the blocking rate. If there is a lack of available capacity to choose the path then the request is rejected. Simulations show that for a number of requests lower than 500 there is no requests blocking in both solutions. Thus in Fig.4, we only plot the blocking rate for requests varying from 500 to 1000 requests. We show that the rate of request blocking is very closed in both solutions. These results mean that our proposition can significantly reduce the number of LSPs and the number of signaling operations. Furthermore, our solution does not degrade the network performance regarding the blocking rate.

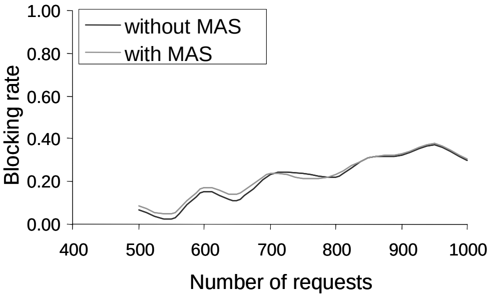

Fig. 4. The blocking rate

7 Conclusion and Future Work

In this paper, we discuss the essential characteristics needed to build an autonomic network. We prove that the multi-agent systems present a good tool in order to reach this purpose. Furthermore, we propose a novel architecture based on the multi-agent systems to automatically manage an MPLS DiffServ-TE domain. Based on the network state, our agents take the appropriate decisions. In our approach, we determine an initial MPLS network topology and then we adapt it to the traffic load. The challenge is to determine when an LSP should be created and when to pass a new traffic in an already created LSP. In order to do that, we propose an LSP creation strategy based on the traffic-driven approach.

Currently, our proposition considers three of the multi-agent systems' features: the decentralization, the reactivity and the sociability. At now, our solution does not consider the proactive and the adaptable features. These two features will be treated in our future work.

Simulation results show that our solution can significantly improve the performance of MPLS by reducing the number of LSPs and the number of signaling operations. Furthermore, our solution does not degrade the network performance regarding the blocking rate.

As future work, we are intended to improve our strategy in order to take into account the proactive and the adaptable features of multi-agent systems. In addition, we will address the possibility of preemption between LSPs. As long term future work, we will define the rules of the route management entity.

Acknowledgment. This is a PhD research supported in part by "Conseil Régional Champage-Ardenne" (district grant) and the European Social Fund. The development is partly realized by two engineer students: Ahmad Sardouk and Wissam Chehadé.

References

1. Aknine, S., Pinson, S., Shakun, M.F.: An Extended Multi-agent Negotiation Protocol. International Journal on Autonomous Agents and Multi-agent Systems, Sycara, K., Wooldridge, M. (eds.), Vol. 8, no. 1 (2004) 5–45
2. Armitage, G.: MPLS: The magic behind the myths. IEEE Communications Magazine, Vol. 38, no. 1 (2000) 124–131
3. Anjali, T., Scoglio, C., de Oliveira, J.C., Akyildiz, I.F., Uhl, G.: Optimal Policy for LSP Setup in MPLS Networks. Comp. Networks, vol. 39, no. 2 (2002) 165–183
4. Awduche, D. O.: MPLS and traffic engineering in IP networks. IEEE Communications Magazine, vol. 37, no.12 (1999) 42–47
5. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: Architecture for Differentiated Services. RFC2475 (1998)
6. Beker, S., Kofman, D., Puech, N.: Off Line MPLS Layout Design and Reconfiguration: Reducing Complexity Under Dynamic Traffic Conditions. International Network Optimization Conference, INOC, Evry, France, N. ISSN: 1762 5734 (2003) 61–66
7. Berger, M., Rosenschein, J.S.: When to Apply the Fifth Commandment: The Effects of Parenting on Genetic and Learning Agents. AAMAS'2004, New York, USA (2004) 19–23
8. Brooks, R. A.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, Vol. 2, No. 1 (1985) 14–23
9. Callon, R.: Predictions for the Core of the Network. IEEE Internet Comp., vol. 4, no. 1 (2000) 60–61

10. Case, J., Fedor, M., Schoffstall, M., Davin, J.: A Simple Network Management Protocol (SNMP). RFC 1157 (1990)
11. Cisco Systems, Inc.: Quality of Service for Multi-Protocol Label Switching Networks. Q & A (2001)
12. Cisco Systems: Implementing Quality of Service Policies with DSCP. (2005)
13. David, E., Azulay-Schwartz, R., Kraus, S.: Bidders` strategy for Multi-Attribute sequential English auction with a deadline. AAMAS-2003, Melbourne, Australia (2003)
14. Dutta, P.S., Jennings, N., Moreau, L.: Cooperative Information Sharing to Improve Distributed Learning in Multi-Agent Systems. Journal of Artificial Intelligence Research, Vol. 24 (2005) 407–463
15. Le Faucheur F. et al.: Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. RFC3270 (2002)
16. Ferber, J.: Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Harlow: Addison Wesley Longman (1999)
17. Kodialam, M., Lakshman, T.V.: Minimum interference routing with applications to MPLS traffic engineering. IEEE INFOCOM 2000, Tel Aviv, Israel (2000)
18. McCloghrie, K., Rose, M.: Management Information Base for Network Management of TCP/IP-based internets. (1990)
19. Minei, I.: MPLS DiffServ-aware Traffic Engineering. Juniper Networks (2004)
20. Ooms, D., Sales, B., Livens, W., Acharya, A., Griffoul, F., Ansari, F.: Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment. RFC 3353 (2002)
21. Osborne, E., Simha, A.: Traffic Engineering with MPLS. Cisco Systems (2003)
22. Rahim-Amoud, R., Merghem-Boulahia, L., Gaiti, D.: Improvement of MPLS Performance by Implementation of a Multi-Agent System. Intelligence in Communication Systems - IntellComm 2005, Springer, Montreal, CANADA (2005) 23–32
23. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC3031 (2001)
24. Sandholm, T. eMediator: A Next Generation Electronic Commerce Server. Computational Intelligence Vol. 18, no. 4, Special issue on Agent Technology for Electronic Commerce (2002) 656–676
25. Xiao, X., Hannan, A., Bailey, B., Ni, L.: Traffic engineering with MPLS in the Internet. IEEE Network Magazine (2000) 28–33