Agent-Based Self-Management of MPLS DiffServ-TE domain

Rana Rahim-Amoud, Leila Merghem-Boulahia, and Dominique Gaiti
Institut Charles DELAUNAY, University of Technology of Troyes
12, rue Marie Curie, BP 2060, 10 010 TROYES CEDEX, FRANCE
{rana.amoud, leila.boulahia, dominique.gaiti}@utt.fr,
WWW home page: http://www.utt.fr/labos/LM2S/

Abstract. MPLS DiffServ-TE presents the solution awaited so much by the network service providers by allowing a differentiation of services and a traffic engineering based on a fast packet switching technology. However, the management of such a network is not a simple function and could not be done manually. In this paper, we propose a novel architecture based on the Multi-Agent Systems (MAS) capable of managing automatically MPLS DiffServ-TE domains. Based on the network states, our intelligent agents take the appropriate decisions. They, for example, reconfigure the network accordingly.

1   Introduction

The Multi-Protocol Label Switching (MPLS) [28] is an Internet Engineering Task Force (IETF) framework that performs label switching between layer 2 and layer 3 protocols. It is a versatile solution to address the problems faced by present-day networks like speed, scalability, quality of service (QoS) and traffic engineering [19].

In recent years, there has been active research in the field of MPLS and an increasing number of networks are supporting MPLS [3]. One of the most significant applications of MPLS is traffic engineering (TE) [4]. MPLS-TE enables resource reservation, fault-tolerance and optimization of transmission resources [23]. 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 service. The combination of DiffServ and MPLS [16] presents a very attractive strategy to backbone network service providers with scalable QoS and traffic engineering capabilities using fast packet switching technologies. The result is the MPLS

DiffServ-TE which will be able to satisfy users' requests like the QoS guarantees while optimizing the use of network resources.

As networks grow rapidly and traffic conditions change frequently, the management of the above MPLS network presents many complexities and could not be done manually. Therefore, automated management is required to minimize this complexity and to engineer traffic efficiently [8]. Moreover, recent researches showed the effectiveness of Multi-Agent Systems (MAS) for the dynamic management of distributed systems [13, 29].

In this paper, we propose a novel architecture based on MAS capable of managing automatically MPLS DiffServ-TE domains. Based on the network states, our intelligent agents take the appropriate decisions. They, for example, reconfigure the network accordingly.

This paper is organized as follows. In the second section we discuss MPLS-TE and DiffServ. A brief description of the multi-agent systems is done in section 3. Our proposition and our proposed architecture are presented in sections 4 and 5 respectively. The LSP creation strategy is presented in section 6. Conclusion and future work are given in section 7.

2   MPLS-TE and DiffServ

2.1   MPLS

MPLS [28] is a new technology that uses labels to forward packets by specifying the Forwarding Equivalence Class (FEC). A FEC is a representation of a group of packets that share the same requirements for their transport. All packets in such a group receive the same treatment in the domain.

MPLS domain contains two types of equipments: LER (Label Edge Router) and LSR (Label Switch Router). The LERs are also called I-LSR (Ingress LSR) for the LSR that puts the label to an incoming packet and E-LSR (Egress LSR) for the one which removes the label from the outgoing packet to return it to its initial nature. LSR is a high speed router device in the core of the MPLS network. The path between two LERs is called LSP (Label Switched Path).

2.2   MPLS-TE

Traffic engineering is used to achieve performance objectives such as optimization of network resources and placement of traffic on particular links [23]. 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 [25].

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 [30]:
1. Different shortest paths from different sources overlap at some links causing congestion on those links.

    2.   The shortest path between a source and a destination is over-utilized while a longer path between these two routers is under-utilized.

TE is needed to avoid these problems by optimizing resource utilization and network performance. In order to control the path of LSPs effectively, one or more attributes can be assigned to each LSP. Such attributes are Bandwidth, Path attribute, Setup Priority, Holding Priority, Affinity, Adaptability, Resilience, etc. [30].

2.3   MPLS - DiffServ

There are many similarities between the functioning of MPLS and DiffServ. In MPLS domain, the classification of incoming packets is done just at the entry of the domain by the I-LSR router, 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].

It is clear that the functioning of MPLS and DiffServ is very similar and it consists of 3 main steps:
    1.   Traffic classification,
    2.   Labeling of packets after classifying them,
    3.   Traffic forwarding according to labels (DSCP in the DiffServ case).

In addition, both MPLS and DiffServ are based on the aggregation.

The mapping between DiffServ and MPLS is still an open research [3]. Currently, there are two solutions [16], 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 scenarii [23]. 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 [30]. In this case, three virtual MPLS networks are defined for EF, AF and BE classes. An example is showed in Fig. 1.

The bandwidth set by administrators on each physical link is partitioned among these MPLS virtual networks. As a result, each network has a percentage of the maximum bandwidth. This will provide better resource utilization and each DiffServ level can be treated alone. The most important thing in the management of such network is the management of LSPs.
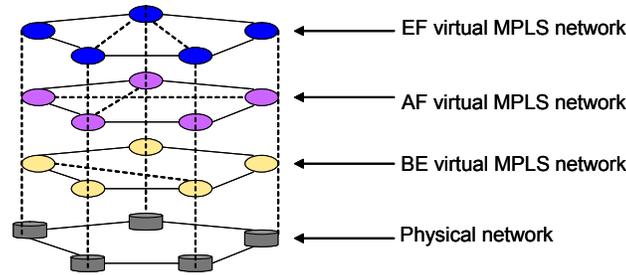
Fig. 1. Virtual MPLS networks

To summarize, we have seen that the TE is needed to avoid some serious problems. DiffServ is also needed to provide a differentiation of services into the MPLS network. However, the management of an MPLS DiffServ-TE network is not a simple function and an automated management is needed to reduce the complexity of the management tasks. Our proposed solution considers that the mapping between DiffServ and MPLS is done by the L-LSP method (multiple virtual networks). In addition, it is based on the use of multi-agent systems which are briefly described in the next section.

3   Multi-Agent Systems

Multi-Agent Systems are an innovative and interesting concept for a great number of researchers in different domains such as road traffic simulation [13, 18, 22], social phenomena simulation [6, 10], biological phenomena simulation [14, 15, 26], negotiation in electronic markets [1, 29], etc.

According to Ferber [17], an agent is a physical or virtual entity having trends and resources, able to perceive its environment, to act on it and to acquire a partial representation of it. It is also able to communicate with other peers and devices, and has a behavior that fits its objectives according to its knowledge and capabilities. Furthermore, agents can learn, plan future tasks and are able to react and to change their behavior according to the changes in their environment. A multi-agent system is a group of agents able to interact and to cooperate in order to reach a specific objective.

Agents are characterized by their properties that determine their capabilities. Different properties are defined like autonomy, proactive-ness, flexibility, adaptability, ability to collaborate and coordinate tasks and mobility. According to its role within its environment, the agent acquires some of these properties. Multi-agent approach is well suited to control distributed systems. Telecommunication networks are good examples of such distributed systems. This explains partly the considerable contribution of agent technology when introduced in this area which motivates us to use it.

4   Our proposition

Since the MPLS functioning is based on the use of LSP in order to forward packets, and the mapping between MPLS and DiffServ is also based on the LSP, it seems that the management of LSPs 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. These attributes will be considered in computing the path for the LSP [30].

   Agents that we will introduce into the MPLS domain have the role to introduce some decision-making abilities in the complex network management tasks. So the introduction of these agents will take place into the decision points of MPLS network. The first step of our research consisted in finding the decision points of the MPLS network which are especially identified on the entry of the domain on the LER routers [27].

   An intelligent agent will be, as a result, introduced on the level of each LER router in the MPLS domain. In order to control and manage effectively the network and to benefit from the distribution feature of MAS, we decide to introduce also an agent on the level of each intermediate LSR. All these agents form a multi-agent system. These agents interact and communicate together and also interact with the routers and switches in the domain. Each agent has a local vision and takes local decisions in order to realize the main goal of the multi-agent system. The architecture of these agents is given in the next section.

5   The proposed architecture

Our principal idea is to control and manage the MPLS network by a simple automated way. This is the reason for which we decide to use Multi-Agent Systems. In fact, the most important feature of a multi-agent system, that we are interested in, is the distribution. An agent has a local view and takes local decisions in order to achieve the global goal of the system. In our case, an agent is introduced on the level of each router in the MPLS domain. Actually, each agent is responsible for the router on which it is introduced and for the corresponding interfaces.

   The architecture of our intelligent agent is shown in Fig.2. It includes two entities: the collector entity (CE) and the management entity (ME) which includes, in its turn, two sub entities: the LSP resource management entity and the LSP route management entity. In addition, the architecture contains a Data Base (DB) which is shared between the CE and the ME.

   The CE collects the information concerning the corresponding router and its interfaces. It collects various parameters like available bandwidth, delay, jitter, queue lengths, and the number of loosed packets. CE collects also the network topology information such as the new created LSPs, if an opened LSP is still useful or not, etc. Furthermore, the interaction between agents is done by their CEs by exchanging the collected information. This information is stored into the DB.
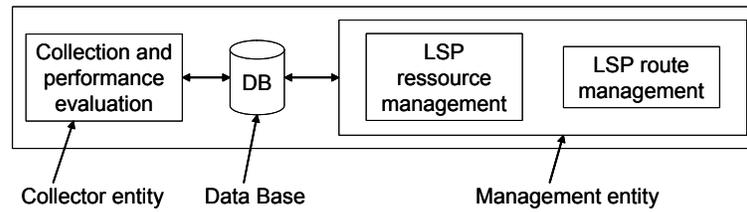
Fig. 2. Intelligent agent architecture

The ME is in charge of taking the corresponding decisions according to the collected information. Possible decisions are:

1. to create a new LSP,
2. to vary the resource allocated to a given LSP,
3. to tear down an LSP,
4. to establish a physical path for an LSP requiring a specified QoS.

After taking the appropriate decision, the next step done by the ME is to automatically implement this decision by configuring the corresponding router accordingly. The last step is done by the CE and consists in checking if the result obtained after the configuration corresponds to the intended one. The succession of these tasks is showed in Fig. 3.
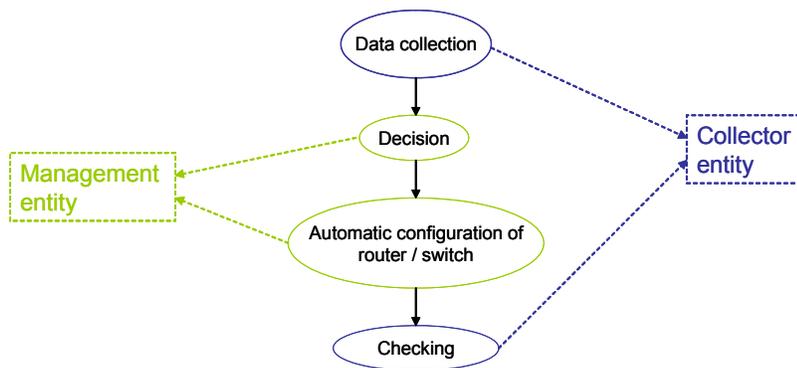
The next section describes the function of the CE.

Fig. 3. The succession of tasks

5.1   Collector entity (CE)

The CEs collect the various parameters of the network and routers. Each CE collects only the information concerning the interfaces of its node. It uses SNMP (Simple Network Management Protocol) [9] to collect information from the MIB (Management Information Base) [21]. The objects collected from the MIB can have a read-only or a read-create access.

The collected information is stored into the DB. The ME which has access to the DB uses this information to take the suitable decisions and to configure the corresponding router accordingly.

We estimate that the available bandwidth is the most important parameter to be treated. It gives us a view of the current network state. At a first step, we will consider that the available bandwidth is the only parameter collected by the CEs and exchanged between agents (Fig. 4).
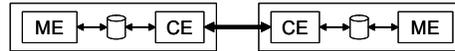

Fig. 4. The interaction between agents is done via the CEs

Each router is requested by the SNMP agent in order to collect the available bandwidth of its interfaces. One of the DB tables, called "LSP table", contains a list of the already created LSPs traversing the corresponding router, their current available bandwidths and the virtual topologies to which they belong. An example of this table can be showed in Fig .5.

Existant LSP	Available bandwith	Virtual topology
LSP1	150	EF
...	...	...

Fig. 5. An example of the LSP table. Each row contains the information concerning an LSP traversing the corresponding router.

5.2   Management entity (ME)

The ME is the most important part of our intelligent agent. In fact, it is responsible for the resource and route management. Precisely, it is responsible for determining when and where an LSP should be created. Based on the information collected by the CE, the ME takes the appropriate decision. The ME contains two sub entities: the LSP resource management entity and the LSP route management entity (Fig. 2).

5.2.1 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 placing many LSPs on a given link. Since the MPLS architecture does not specify any routing protocol, all standardized protocols can be used. The solution is then to choose the most appropriate protocol to be activated according to the network conditions. The decision rules of this entity will not be discussed in this paper.

5.2.2 LSP resource management entity

The role of this entity is to manage the LSP resources. In other terms, it is responsible for creating and re-dimensioning LSPs, etc. To take the suitable decision, the LSP resource management entity has to follow a strategy that we have proposed and called the "LSP creation strategy". This strategy is described in the next section.

6   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 [7].

   To design the MPLS layout, there are on-line and off-line proposed approaches. Off-line approaches are based on the estimation of the traffic demand over time. According to Kodialam [20], off-line approaches are not appropriate to MPLS networks due to the high unpredictability of the Internet traffic. Another approach is based on the creation of a fully connected MPLS network. This approach consists of creating an LSP between each pair of nodes. This provides a large number of LSP introducing, as a result, high control traffic, high signaling cost, high management cost and high management complexity. Since these two methods present many disadvantages, they are to be avoided.

   On-line methods calculate paths as demands arrive. Three different approaches can be distinguished:
   1.   Request-driven
   2.   Topology-driven
   3.   Traffic-driven

   The request-driven approach is used when MPLS transmits multicast traffic [24]. We have avoided 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, LSP already exists before traffic is transmitted. Thus, a constructed path may be not used because its creation was based only on the routing information. That is the big disadvantage of this approach.

   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 request 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 link is equal to the maximum bandwidth of the physical link minus the total bandwidth reserved by LSPs traversing the link. It does not depend on the actual amount of available bandwidth

on that link [30]. The available bandwidth on a physical link is given by the following equation (1):

$$B_a = B_{rt} - \sum_{i=1}^{n} B_i \tag{1}$$

Where B_a is the available bandwidth on the link, B_{rt} is the maximum reserved bandwidth of the physical link and B_i is the bandwidth reserved for the LSP_i.

We remark 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 technology.

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. A topology change will take place, by consequence, 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
3. The cost

A request can be a new bandwidth request, a disabled bandwidth request or a request for disabling an existing LSP. The request is a very important factor because the type and the number of requests occurred at a precise instant have a big influence on the decision to be taken.

The network state includes the current MPLS topology (the virtual topology) 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, it includes the physical link attributes (i.e. the available bandwidth, the delay, etc.). As we have mentioned above concerning the LSP attributes and the physical link attributes, our strategy takes into account only the available bandwidth. The other parameters will not be treated in this paper.

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 switching cost defined by the operator. (3) The bandwidth cost which depends on the carried bandwidth and the number of traversed nodes.

Taking into account these factors, we can distinguish many cases. The first and trivial case is when a request for disabling an already created LSP arrives. In this case, the LSP resource management entity takes the decision (1) to tear down the LSP, (2) to release the corresponding labels and (3) to liberate the bandwidth. As a result, the available bandwidth on the physical link is increased by the value of the liberated bandwidth.

A second trivial case is when a request for bandwidth is deactivated. In this case, the LSP resource management entity takes the decision (1) to liberate the corresponding bandwidth and (2) to increase the available bandwidth of the

corresponding LSP by the value of the liberated bandwidth. Thus, the available bandwidth on the physical link remains the same.

Another case more complex than the previous ones is when a new bandwidth (BW) request arrives. The LSP creation strategy for this case is defined in the diagram showed in Fig.6.

Consider that a new BW request arrives between a pair of routers demanding a certain level of QoS. 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 the LSP exists, the next step is to compare the available BW of that LSP with the requested BW. If the available BW is higher than the requested one, the requested BW is allocated on that LSP and its available BW is reduced accordingly.
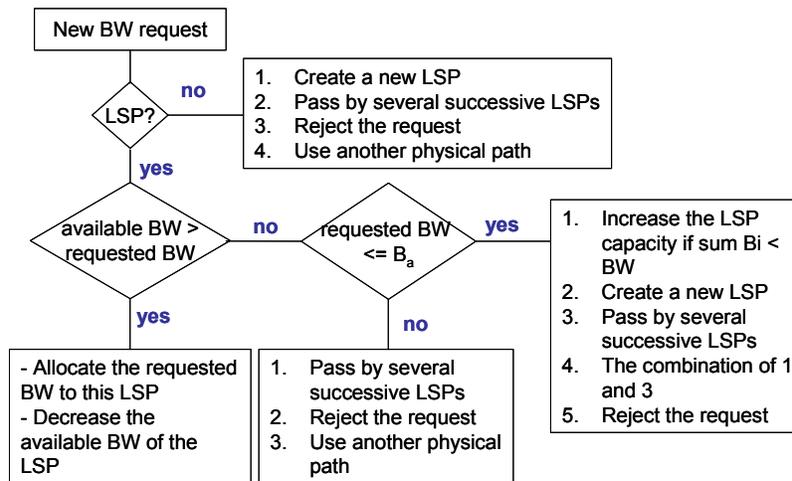

Fig. 6. LSP creation strategy diagram

If the available bandwidth is lower than the requested one, two cases can be distinguished:

a.   The requested BW is lower than or equal to the available BW on the physical link. In this case, several possibilities take place:

1.   According to Eq. 1 it is possible to increase the capacity of the LSP. In other words, the BW reserved for the LSP in question will be increased by a value equal to the requested BW to be able to forward the new traffic.

2.   Another possibility will be to create a new LSP and to reserve a BW to him equal to that requested.

In these two cases the available BW of the physical link is decreased by the value of the requested BW.

3.   A third possibility will be to pass by several successive LSPs by using the label stack, an example can be showed in Fig. 7.

In the case of Fig.6 there are two LSPs: LSP1 (Igress-LSRx-LSRy) and LSP2 (LSRy-LSRz-Egress). These two LSPs have one LSR in common and the idea is to use the LSP (Igress-LSRx-LSRy-LSRz-Egress).

The Ingress LSR pushes two labels onto the label stack of the incoming packet, label1 to be used to forward the traffic in LSP1 and label 2 to be used to forward the traffic in LSP2. When the packet arrives to the LSRy, LSRy pops off label1 and forwards the packet according to the label2.

4. The combination of the possibilities 1 and 3. An example can be to increase the capacity of LSP1 and use LSP1+LSP2 to forward the incoming traffic.

5. The last possibility will be to reject the request.

b. The requested BW is higher than the available capacity of the physical link. In this case, only the two possibilities (3 and 5) will be valid. Another solution is to pass by other physical links indicated by the routing protocol. In this case, a new LSP will be created between the pair of routers traversing another physical path.
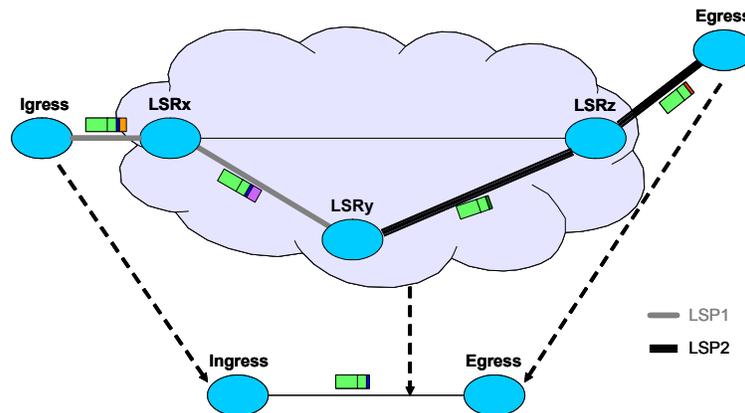

Fig. 7. Using multiple LSPs

If there is no LSP between the pair of routers, only the possibilities 2, 3 and 5 can be valid. Another solution is also to pass by other physical links as in b.

The previous cases correspond to the arrival of only one request. If several requests having the same type or different types arrive at the same moment, the decision will be much more complex. The decision will be a generalization of the previous cases. Considering the multitude of the possible cases, the most practical method which appears to us is the use of an optimization method which enables us to choose the best decision according to the various conditions and various factors. To do that, we have to choose the suitable optimization method and to formulate our problem as a mathematical model.

7   Conclusion and future work

In this paper, we propose a novel architecture based on the Multi-Agent Systems capable of managing automatically MPLS DiffServ-TE domains. Based on the network states, our intelligent 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. Our decision rules consider many factors like the cost, the request, the available bandwidth and the network topology information. As future work, we are intended to complete our proposed LSP creation strategy by choosing an optimization method and formulating our problem as a mathematical model. Once done, we will test this proposition and compare the results. As long term future work, we will define the decision rules of the LSP route management entity and test it.

Acknowledgment

References

1. Aknine S, Pinson S, Shakun MF (2004) An Extended Multi-agent Negotiation Protocol. International Journal on Autonomous Agents and Multi-agent Systems, Sycara, K., Wooldridge, M. (eds.), Vol. 8, no. 1, pp.5-45
2. Armitage G (2000) MPLS: The magic behind the myths. IEEE Communications Magazine, Vol. 38, no. 1, pp. 124-131
3. Anjali T, Scoglio C, de Oliveira JC, Akyildiz IF, Uhl G (2002) Optimal Policy for LSP Setup in MPLS Networks. Comp. Networks, vol. 39, no. 2, pp. 165-183
4. Awduche D O (1999) MPLS and traffic engineering in IP networks, IEEE Communications Magazine, vol. 37, no.12, pp. 42-47
5. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W (1998) Architecture for Differentiated Services. RFC2475
6. Ben Said L, Bouron T, Drogoul A (2002) Agent-based interaction analysis of consumer behavior. AAMAS' 2002, ACM, Bologna, Italy, pp. 184-190
7. Beker S, Kofman D, Puech N (2003) Off Line MPLS Layout Design and Reconfiguration: Reducing Complexity Under Dynamic Traffic Conditions. International Network Optimization Conference, INOC, Evry, France, N. ISSN: 1762 5734, pp. 61-66
8. Callon R (2000) Predictions for the Core of the Network. IEEE Internet Comp., vol. 4, no. 1, pp. 60-61
9. Case J, Fedor M, Schoffstall M, Davin J (1990) A Simple Network Management Protocol (SNMP). RFC 1157
10. Conte R, Gilbert N, Sichman JS (1998) MAS and Social Simulation: A Suitable Commitment. MABS'98. Lecture Notes in AI, Vol. 1534, Paris, France, pp. 1-9
11. Cisco Systems, Inc. (2001) Quality of Service for Multi-Protocol Label Switching Networks. Q & A
12. Cisco Systems (2005) Implementing Quality of Service Policies with DSCP.
13. de Oliveira D, Bazzan ALC, Lesser VR (2005) Using cooperative mediation to coordinate traffic lights: a case study. AAMAS, Utrecht, Netherlands, pp. 463-470
14. Doran J (2001) Agent-Based Modelling of EcoSystems for Sustainable Resource Management. 3rd EASSS'01. Lecture Notes in AI, Vol. 2086, Prague, Czech Republic, pp. 383-403

15. Drogoul A (1995) When ants play chess. In From reaction to cognition. Lecture Notes in AI, Vol. 957. C. Castelfranchi & J.P. Müller (Eds). Springer-Verlag, Berlin-Heidelberg, pp. 13-27
16. Le Faucheur F et al (2002) Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. RFC3270
17. Ferber J (1999) Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Harlow: Addison Wesley Longman
18. El Hadouaj S, Drogoul A, Espié S (2000) How to Combine Reactivity and Anticipation: the Case of Conflicts Resolution in Simulated Road Traffic. MABS'2000 workshop. Lecture Notes in AI, Vol. 1979, Boston, USA, pp. 82-96
19. The Internal Engineering Consortium (2005) Multi Protocol Label Switching. Web ProForums Tutorials, http://www.lec.org
20. Kodialam M, Lakshman TV (2000) Minimum interference routing with applications to MPLS traffic engineering. IEEE INFOCOM 2000, Tel Aviv, Israel
21. McCloghrie K, Rose M (1990) Management Information Base for Network Management of TCP/IP-based internets.
22. Moukas A, Chandrinos K, Maes P (1998) Trafficopter: A Distributed Collection System for Traffic Information. CIA'98. Lecture Notes in AI, Vol. 1435, Paris, France, pp 34-43
23. Minei I (2004) MPLS DiffServ-aware Traffic Engineering, Juniper Networks
24. Ooms D, Sales B, Livens W, Acharya A, Griffoul F, Ansari F (2002) Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment. RFC 3353
25. Osborne E, Simha A (2003) Traffic Engineering with MPLS. Cisco Systems
26. Pave A, Bousquet F, Cambier C, Mullon C, Morand P, Quensiere J (1993) Simulating the Interaction between a Society and a Renewable Resource. Journal of Biological Systems, Vol. 1, pp 199-213
27. Rahim-Amoud R, Merghem-Boulahia L, Gaiti D (2005) Improvement of MPLS Performance by Implementation of a Multi-Agent System. Intelligence in Communication Systems - IntellComm 2005, pp. 23-32, Springer. Montreal, CANADA, Oct. 17-19
28. Rosen E, Viswanathan A, Callon R (2001) Multiprotocol Label Switching Architecture. RFC3031, IETF
29. Sandholm T eMediator (2002) A Next Generation Electronic Commerce Server". Computational Intelligence Vol. 18, no. 4, pp. 656-676, Special issue on Agent Technology for Electronic Commerce
30. Xiao X, Hannan A, Bailey B, Ni L (2000) Traffic engineering with MPLS in the Internet. IEEE Network Magazine, pp. 28-33