An Architecture to provide Guarantee of Service (GoS) to MPLS[1]

M. Domínguez-Dorado[1], F. J. Rodríguez-Pérez[1], J. L. González-Sánchez[1],
J. L. Marzo[2], A. Gazo[1]
[1]Computer Science Department, University of Extremadura
Avda. de la Universidad s/n, E-10071 Cáceres, SPAIN
[2]Institut de 'Informàtica i Aplicacions, Universitat de Girona, Girona, SPAIN
ingeniero@manolodominguez.com, {fjrodri, jlgs}@unex.es, marzo@eia.udg.es, agazo@unex.es

Abstract: - MPLS (Multiprotocol Label Switching) technology provides powerful mechanisms to integrate network technologies like ATM and IP with Quality of Service. Although this technology is becoming mature, there are still some aspects to be solved, such as offering guaranteed services to privileged sources that can require GoS (Guarantee of Service). To do so, on the one hand a mechanism of local recovery or packets retransmission requiring GoS is analysed; on the other hand the implementation of a local LSP (Label Switched Path) recovery system is studied.

Key-Words: - MPLS, Guarantee of Service, Local Retransmission, Active Methods

1 Introduction and related works

Nowadays the data communications networks are gaining importance; the increasing number of users, the growth of the applications for which the data interchange is needed, and the migration from traditional telephony to IP telephony, video on demand, etc, have several effects on the network technological infrastructure providers, leaded to carry out a very hard transformation to be able to respond to the modern society demands.

Simultaneously, the appearance of the optical switching technology, capable of managing large volumes of information, requires the design of new signalling methods and new communication protocols that allow to make good use of its advantages and transform the network into an intelligent entity; a resilient network that provides not only the simply passive information transport but also the resources management and reliability on the information and on the network infrastructure itself. A parameterised network that, furthermore, achieves the target of reducing the network services provider's costs and unifying the great amount of actually deployed technologies whose maintenance is not only a technical problem but also an economic one due to the difficulty of offering broadband services with an acceptable quality of service [1].

Currently, MPLS [2] makes good use of optical technologies and provides fast networks, since there is no need to undergo layer-3 lookups between the LSP end-points. This is done at the expense of assuming that network is not going to fail. The problem arises when that remote possibility happens, because this is the moment when great part of the traffic will be lost [3]; higher level protocols can request the retransmission, but the time lag it can involve is high. For some type of applications sensitive to the reliability, MPLS should be able to assure that the traffic will not be affected or that it will be significantly lesser, but it is not able to assure this. MPLS has two main problems in order to be able to guarantee to some kind of traffic that they will arrive without problems:

- What to do and how to act when a physical path fails and it transports packets belonging to a flow that must be prioritised.
- How to respond in view of nodes congestion when discarded packets do belong to this kind of traffic.

This work presents a technique that brings guarantee of service (GoS) to privileged information flows, allowing discarded frames to be recovered and LSP to be restored in a local environment, avoiding in this way, as far as possible, end to end retransmissions requested by transport layer.

The following section will deal with the subject of what is GoS and how it can be applied to privileged flows in a MPLS environment; In the third paragraph we will study the structure and functioning of the elements responsible of providing GoS in a MPLS domain and finally, this article concludes indicating the contributions of this research.

2 Guarantee of Service over MPLS

The GoS requirements contribution for a MPLS flow can be understood as the capacity of discarded frame local recovery as well as local LSP recovery [4]. In this way, this work proposes the use of four GoS levels, beside the existence or not of a backup LSP (Label Switched Path), so each packet can be marked with these attributes from initial node to end node. Each one of these four levels must be understood like a grade of probability that a frame can be localized in any of the active nodes it has been passing through. So the need of end to end retransmissions is avoided, solving it in a much rather local environment.

The need or not of a backup LSP creation will come specified by a parameter of boolean type included in a three control bit codification. Through the decodification of these three values the packet will be retained and processed in the node with regard to the requirements that these bits show. In table 1 the use of these three bits to obtain every possible option, is shown.

The different GoS level implementation has been realized by means of two aspects: on the one hand, in the MPLS packet header and, on the other hand, in the network level header.

To show in MPLS that a packet is marked with any level of GoS, we have decided to use the 1 value as *label* field because this value has been defined as a special one for MPLS labels [2]. In the *EXP* field of the same label (see figure 1) we have introduced the three bits we need. This mark will be able to be set by ingress LER (Layer Edge Router), a node that allows the access to the MPLS domain, using the information kept in the IP header to do it.

Table 1. Codification of Guarantee of Service levels.

LSP	GoS_1	GoS_0	Meaning
0	0	0	Not marked with GoS. A traditional packet.
0	0	1	Level 1 of GoS and without backup LSP.
0	1	0	Level 2 of GoS and without backup LSP.
0	1	1	Level 3 of GoS and without backup LSP.
1	0	0	Not marked with GoS but with backup LSP.
1	0	1	Level 1 of GoS and with backup LSP.
1	1	0	Level 2 of GoS and with backup LSP.
1	1	1	Level 3 of GoS and with backup LSP.

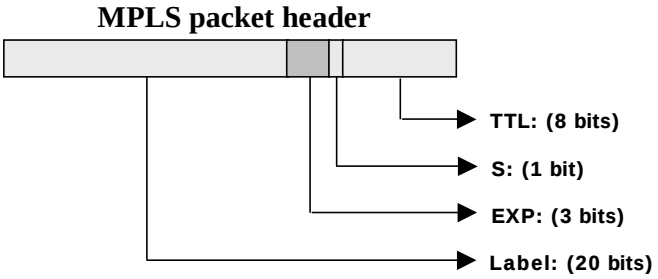

Fig. 1. MPLS packet header structure.

In a primary survey we could have used the *ToS* (Type of Service) field, which is eight bits size. However, its use has been modified sometimes until its disappearance [5].

The possibility of a reinterpretation of this field has been discarded because *ToS* field is used now to specify different *DiffServ* levels and to notify about nodes congestion. The idea of incorporate differentiated services together with our proposal of GoS can result attractive. That is why we do not aim to limit the system having to decide between one option and another one, only because *ToS* serves for both. Because of this, the GoS codification has been implemented over the *options* field, which has a variable size, at most 40 octets.

However we will only need the use of the first byte to codify the three bits that specify our strategy for requirement or not of backup LSP and the different GoS levels.

3 Path marking and lost recovery

During the transfer, data packets will have some information attached to themselves about how they must be handled by the nodes. Thus the functioning of the node, an *active node*, would be dynamic, it would not act always in the same form. Its operation will depend on the traffic that passes through it.

Let us suppose a scene formed by 4 nodes A, B, C and D (see figure 2), among them A and D are MPLS active nodes and B and C are MPLS nodes. Packets coming from A or B can arrive to D, but there are undistinguishable for it because it only has knowledge about the incoming label and the incoming port of these packets. And it only recognizes that C is the sender. It could distinguish their origin based on the label but it would not be reliable enough because C could incorporate aggregation mechanism that merges both flows, coming from A and B, into a unique flow. If at this point D loses a packet due to saturation, it must find out to which it has to request the retransmission. It could not request to C because that is not an active node and so it could not understand it.

Therefore a fundamental aspect in our system is to know the set of nodes by which a packet marked with GoS has passed through because, in case of loss, retransmission could be requested to them**,** without need of doing it to the message source node.

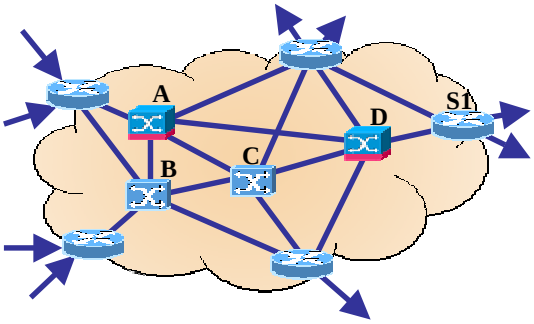

Fig. 2. An example MPLS scene in which traditional nodes and actives nodes coexist.

That is why we have assigned more capacity to the LSR (Label Switch Routers), since it is going to be able to watch further than the MPLS header. Moreover, it is needed that the nodes considered active mark its network level address on the packets. We have decided to perform this marked at network level as due to the fact of using, for example, some bits from the MPLS label would end up with the transparency principle of MPLS, so that classic nodes, non-active, that exist in a network have not difficulties to handle the traffic marked with GoS.

On the other hand, we have decided to transform the *option* field in a stack of network level address to store the addresses of the active nodes that the traffic has been passing through. So we always know the last *n* nodes by which the packet has passed through. Firstly, it could be n = (40 -1) / 4 = 9 addresses of active nodes, what we think is suitable, because we do not propose the replacement of all the nodes in a domain but so the incorporation of some active MPLS nodes. In this way, in the case that a retransmission was necessary, we could go backwards towards the source at most 9 active nodes, increasing thus the possibilities of finding the lost packet.

Therefore, in order to control the store, search and retransmission tasks, it is necessary the definition of a retransmission protocol, we have called GPSRP (GoS PDU Store and Retransmit Protocol). Moreover, allowing local retransmissions implies the need of having an intermediate, temporal memory in the active nodes. In such buffer the localized packets needed for a possible retransmission can be found. This memory is named DMGP (Dynamic Memory for GoS PDU). In the figure 3 the architecture of the proposed nodes can be appreciated.

The buffers in this node accept incoming traffic that must be served by a Prioritised Round Robin algorithm; so, we assure that the most important traffic will be attended to faster, according to the priority scale previously defined, independently of the arriving moment to the buffer. Same kind of traffic will be served by a traditional Round Robin algorithm until the appearance of most prioritised traffic.

When the packet has been read from the buffer, it is automatically attended to by the appropriated protocol module. If the packet is TLDP (Tiny Label Distribution Protocol, a LDP protocol reduced subset at functional level), the TLDP module will attend to it and, as it is a signalling packet, it will possibly modify the values in the switching array, formed by ILM (Incoming Label Map), FTN (Functional Equivalence Class to Next Hop Label Forwarding Entry) and NHLFE (Next Hop Label Forwarding Entry) if required. If the packet is a GPSRP packet, in charge of GoS packet retransmissions, the corresponding modules will attend to it and in order to do it, it must access to DMGP where the packets marked with some GoS level are stored. GPSRP starts to work also when EPCD (Early Packets Catch and Discard, defined in section 3.4), always

monitoring the incoming buffer, notifies it that a GoS marked packet has been discarded. In this case, in addition to the notification, EPCD gives the MPLS/IP header of the packet to the GPSRP module in order to carry out the retransmission request.

If the packet is a RLPRP (Resilient Local Path Recovery Protocol, commented in sec. 4) packet, whose task is to keep backup LSPs for the flows that require it, it would be this protocol which attends to the packet, notifying the new situation to the involved active nodes and switching to the new path as fast as possible if necessary. After this, it must establish a new LSP that will become the backup LSP one. If the packet is MPLS, the MPLS module will seek an item in the switching array according to the incoming packet label; if it does not exist, TLDP will become active requesting a label and the packet will return to the buffer again until the adjacent node respond. Eventually, if the incoming packet is IPv4, it is classified and checked if there is a coincidental FEC (Functional Equivalence Class) for the packet in the switching array. If it is not, a label will be requested for this packet, it will return to the buffer again and we will wait for a response. In any case, routing algorithm will be available for the protocols that need it at every time, helping to set a switching array according to the routing policy and the protocols over IP to select the adequate route to go to the target node.

When the active node is handling non-active packets, it will use a routing algorithm based on the links delay. When these packets are active, that is, they are marked with some GoS level, the routing algorithm used will be RABAN (Routing Algorithm for Balanced Active Networks, explained in section 4), that will try to select a route not only with few delay but also with few traffic, enough resources and passing through active nodes when necessary.

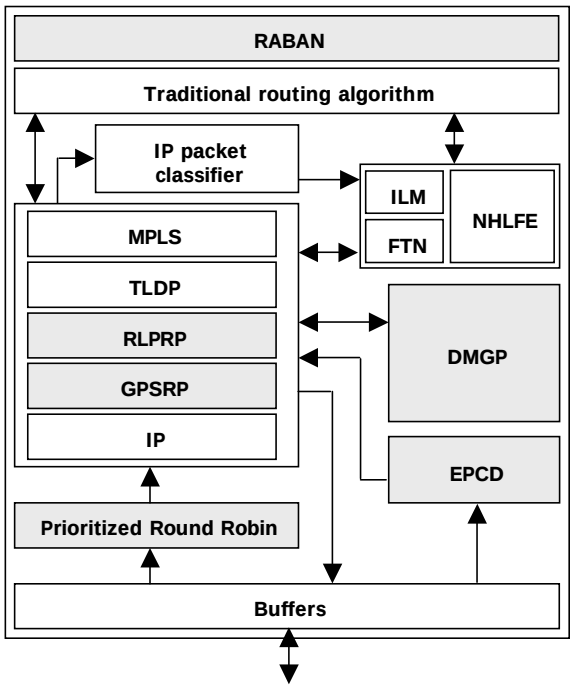

Fig. 3. Internal architecture of an active node with routing features.

3.1 High level layers protocols

In a MPLS communication the implied levels are those of network, link and level 2+ or MPLS. However, we have to bear in mind the possibility of marking the whished GoS level in the transport layer for the application level packets. Thus, following the TCP/IP model, we would find that data would be marked at application level directly by users and after the network application would mark the TCP segments that, being encapsulated over IP packets, would results in processed packets.

At application level, the user can start a session for the GoS packets retransmission; the user indicates this option by selecting the receiver port when opening TCP socket (when accessing to the transport layer). In the same way that, for instance, in order to make use of an electronic mail service we access to the port 110 or to use a SSH services, to port 22, we will dedicate seven ports to open TCP sessions with each one of the seven GoS available levels (GoS + backup LSP). This will cause the transport level to be marked with the three bits needed to include in this level.

In the TCP header there are six bit reserved since the initial development of TCP. For a long time that field has remained intact, but in the recent years, some of its bits have started to be used, in concrete two of them, to be able to mark some of the differentiated services options [6]. We have still four available bits, from which we would use three and there would still be one left for other uses (see figure 4). In this form, we can specify the order of prioritising the packet from the application level to the network level passing by the transport level, without any problem.

3.2 The temporal DMGP memories

The analysis of the DMGP memory size (see figure 3) requires a detailed study. The variable size of IP frames implies to realise complex calculations to obtain the optimum size for the DMGP in the active nodes. On the other hand, we must take into account the distribution of the memory between the different kinds of incoming flows, so we always can assure that a number of packets belonging to a privileged flow can be stored in the memory for its likely local retransmission. This circumstance limits the maximum number of packets that can be referenced in memory as the use of a fixed identifier can suppose a disadvantage for a network in which a lot of prioritised flows has been marked (with GoS). Summarizing, in addition to take into account the possible packets size, some aspects such as kinds of traffics, transfer rates, etc, of the traffic that is really passing round Internet, must be borne in mind.

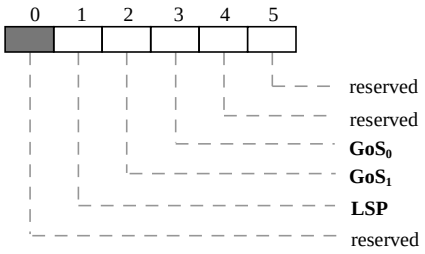

Fig. 4. Some bits unused in the TCP header.

3.3 Global packets identifying

During a retransmission, the identification of each packet stored on the intermediate DMGP memories is necessary. In order to achieve it, the PDU marked with guarantee of service must be indexed on these memories. In that form we will have each one of the globally sent and received packets identified in the MPLS domain. So**,** we need an identifier that permits to recognize each packet whose retransmission is desired, from the source side as well as from the side of the node that stores in its DMGP buffer the GoS marked packets.

The IP address from network layer allows identifying each node in a network topology. However, it can not identify unmistakeably by itself each packet generated by a specific node. This is why we will need an *id* identifier that will go with each GoS marked packet and that will be assigned by the node that generates it. In short, we will consider as unique identifier for a GoS marked packet to the pair of values formed by the *network address* of the packet sender together with the *id* identifier with which such node marks each packet.

A 4 octets *id* identifier allows us to recognize at most 2^{32} = 4.294.967.296 packets generated by the same node. From this moment on it would start to assign *ids* from the beginning, allowing the existence of two packets carrying out the same identifier. However it is likely that before starting to repeat identifiers, the supposed "repeated" packets, have abandoned the MPLS domain, what is less likely if the addressing is lesser than 2^{32}, because we are planning an architecture suitable for using in backbones networks in which the information volume will be predictably high. This four bytes value will be also stored on the *options* field, after the octet concerning the GoS levels and before the stack of addresses of actives nodes passed through. Thus, in order to support GoS, IP *options* field will be formatted like it is shown in figure 5.

3.4 Packets discard in the buffers of an active node

In order to attain a fair treatment of the packet that come in to a specific buffer, the use of a scheduling algorithm is needed. So, we will use a circular Prioritised Round Robin in such a way that in case of the existence of some packets with the same priority, those indicated by Round Robin will be processed and in the opposite case, packets marked with more priority will receive a preferential treatment.

In the table 2 the different considered priorities are shown. The different priority levels have been assigned depending on the importance that the loss of such kind of packets would have for the whole communication or for the well network functioning.

In this way, when saturation exists in the buffer of a determined node, some packets will be able to be discarded. But in this circumstance no-GoS MPLS packets have higher probability of being discarded whereas those belonging to TLDP traffic (LDP protocol reduced subset at functional level) will be only discarded if there is no other option.

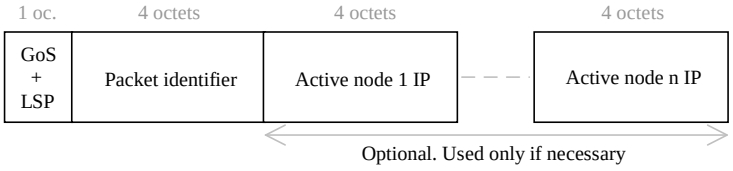

Fig. 5. Proposed format for the IP options field.

Table 2. Packets classification according to its priority.

Level	Type of packet
PRIORITY 10	TLDP packet
PRIORITY 9	GPSRP packet
PRIORITY 8	RLPRP packet
PRIORITY 7	MPLS packet with GoS 3 and backup LSP
PRIORITY 6	MPLS packet with GoS 3 and not backup LSP
PRIORITY 5	MPLS packet with GoS 2 and backup LSP
PRIORITY 4	MPLS packet with GoS 2 and not backup LSP
PRIORITY 3	MPLS packet with GoS 1 and backup LSP
PRIORITY 2	MPLS packet with GoS 1 and not backup LSP
PRIORITY 1	MPLS packet without GoS and with backup LSP
PRIORITY 0	Traditional MPLS packet

In the case of a packet being discarded and in order to avoid requesting its end to end retransmission, GoS marked packets are stored for some time in the active nodes in order to be recovered inside the MPLS domain, avoiding in this way a higher global traffic. Nevertheless, to request a local retransmission to an active node, we need to recover at least the IP header from the discarded packet, where its identification as well as the last n active nodes the packet has passed through, are stored.

We need to use a special buffering management algorithm, to recover this information from packets discarded due to saturation and that will be named EPCD *(Early Packet catch and Discard)*.

4 Packets routing

The different routing strategies that can be used to make a message go from the source to the receiver node can also contribute to the performance improvement. To do it they must select the most suitable routes for the kind flow being transported as well as the present network status. In this form we will be able to distinguish between normal MPLS traffic or GoS marked MLPS traffic. A traditional MPLS node will implement an algorithm in which any links weight will be simply its delay. Nevertheless, an active node will run an algorithm in which the links weight will represent a weighted calculation of different parameters:

- Link delay.
- Number of LSP supported by the link.
- Number of established backup LSP over the link.
- Saturation state for the nodes connected by the link.
- Packets on-fly estimation.

Through this routing algorithm with weighted values we aim to obtain an equilibrated network in which the load has been balanced. In this way the network resources over-exploitation and under-use are avoided, trying also to reduce the number of collisions. We will call this algorithm RABAN *(Routing Algorithm for Balanced Active Networks)*.

On the other hand, when we need to create a backup LSP, it must comply with some requirements such as to coincide as less as possible with the original LSP route. It is also of great interest that the backup LSP passes through MPLS active nodes because

there is more probability that a service requiring backup LSP also requires GoS. RABAN algorithm must determine if some gain will be obtained by passing through active nodes at the expense of accepting possibly slower routes. So, we need a protocol in charge of backup LSP establishment and switch between them when a fail is detected. It is complex to obtain an efficient behaviour that avoids the chained data loss reaction and above all it is complex to maintain the switches and routers label coherence in an adequate time period. The developed protocol in this proposal is ***RLPRP (Resilient Local Path Recovery Protocol)*** and it will be deal with the main LSP fail detection, notifying to the active nodes in charge of the backup LSP maintenance and switching to it as soon as possible. After this, it will establish a backup LSP again as the previous one has become the main LSP now.

Eventually, we will choose for the creation of partial backup LSP inside the domain, locally, to solve link fails between active nodes inside the domain. That implies that active LSR must have features typical of LER, since they will function like ends of such path; they will also have to generate labels and possess routing skills. However, this is a faster solution and is lower resource-consumer that the end to end LSP establishment solving, indeed, the problems in a much more local way.

5 Results

The topology of the figure 2 has been used for the validation of the system. *A* and *D* are the unique active nodes, with DMGP size of 1 MB and 100 KB, respectively. In the figure 6 can be observed that approximately 350 packages have been rejected in the active node *D*. This has caused 350 retransmission requests, which have allowed to recover locally 280 packets. At the final moment of the simulation, 70 packages were staying in fly (GPRS requests without answering yet), but there was no packet without possibilities of recovery in that moment. Thus we can verify that the system is capable of locally recover a high percentage of packets rejected by saturation, avoiding this way the end-to-end retransmission of the same ones.

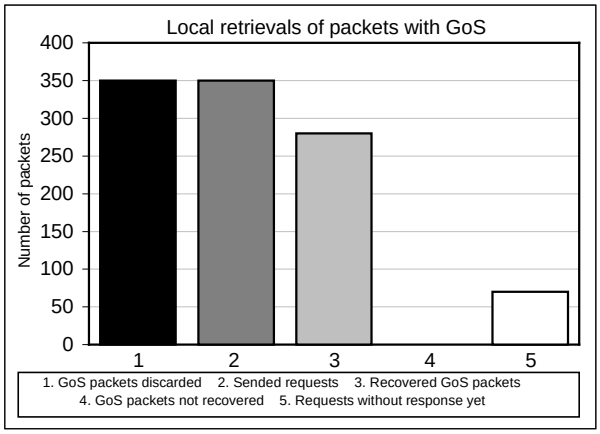

Fig. 6. Number of packets locally recovered in a congested active node

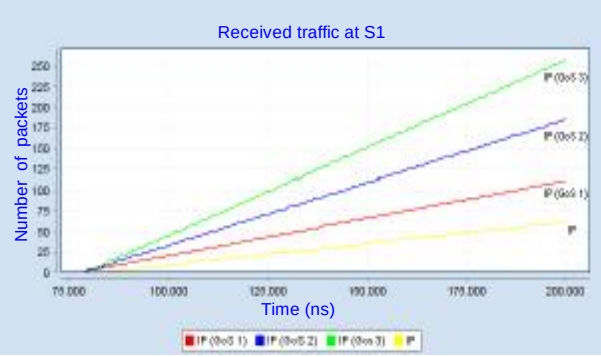

Fig. 7. Traffic volume received in a sink node in function of the GoS level

On the other hand we have analysed the priority of the traffic with GoS (see figure 7). We have raised an environment with four flows (*GoS 1*, *GoS 2*, *GoS 3* and *No GoS*); identical in size, number of packets and generation speed. Also there is no node saturated in the network, existing three intermediate active nodes. In this case, the aim of the simulation consists of analysing the number of packets of every flow that will arrive to a sink node *S1*.

The result is that independently of the quantity of packets that arrive to an active node, this one processes more packets of those traffics with major GoS level than those with minor priority.

6 Conclusions and future works

This work proposes a local packets recovery mechanism in a MPLS domain environment. Thus, it brings GoS to privileged traffic sources that require reliability. The proposed architecture has been validated by means of simulations which demonstrate that a great number of packages can be recovered locally, which, without our proposal, would have to be retransmitted in an end-to-end way. On the other hand, simulations also demonstrate that with better GoS, more traffic arrives to the receivers.

References

[1] Janus Gozdecki, Andrzej Jajszczyk, and Rafal Stankiewicz, “Quality of Service Terminology in IP Networks,” IEEE Communications Magazine, March 2003.

[2] E. Rosen et al., “Multiprotocol Label Switching Architecture,” RFC 3031, January 2001.

[3] Jose L. Marzo, Eusebi Calle, Caterina Scoglio, and Tricha Anjali, “QoS Online Routing and MPLS Multilevel Protection: A Survey,” IEEE Communications Magazine, October 2003.

[4] M. Kodialam and T. V. Lakshman, “Restorable Dynamic QoS Routing,” IEEE Communications Magazine, June 2002.

[5] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, September 2001.
[6] DARPA Internet Program, "Transmission Control Protocol," RFC 793, September 1981.

[1] This work is sponsored in part by the Regional Government of Extremadura (Education, Science and Technology Council) under Grant No. 2PR03A090 and by the CICYT under Grant No. TIC2003-05567.