Telecommunication Systems (2019) 72:11–27
https://doi.org/10.1007/s11235-018-00543-7

An approach to support traffic engineering in IPv6 networks based on IPv6 facilities

Line Y. Becerra Sánchez[1] · Jhon J. Padilla Aguilar[2]

Published online: 5 January 2019

Abstract

IPv6 is an Internet protocol with the ability to provide a large number of addresses to allow the connectivity of each existing thing to the global network. It also allows the deployment of many technologies and services of the next generation. One of the major changes that occurred in the IP header with this new version is the addition of the IPv6 flow label field, which was created with the intention of labeling packets that belong to a particular flow to provide an appropriate treatment by routers. However, this field has not been widely exploited yet, and it is being set to zero in almost all IPv6 packets. The main Internet routing problem is that said routing is based on the shortest path algorithm, which leads to the possibility of some paths being congested while others are underused. To solve the congestion problem, many solutions aiming at traffic engineering support have been proposed, but this topic remains an open issue. This paper describes a new solution to support traffic engineering based on the usage of the IPv6 flow label for providing fast packet switching, which we have called PSA-TE6. In this document, we present the PSA-TE6 operation and evaluation regarding the label space reduction, label stacking cost and its minimization. The results show that PSA-TE6 is cheaper compared to the IP/MPLS solution when there is no label stacking, and that PSA-TE6 also outperforms IP/MPLS when the stacking is enabled until achieving a 40% presence of tunnels for encapsulation levels greater than 1.

Keywords IPv6 · Packet switching · IPv6 flow label · Traffic engineering

1 Introduction

Currently, the world is undergoing a global change in Internet technology because we are in the transition from IPv4 [1] to IPv6 [2, 3]. The IPv6 protocol offers great advantages compared to IPv4 with regard to increased addresses, the provision of quality of service, mobility, and other features. Complete studies of IPv6 deployment and the migration of the Internet to IPv6 are described in [4–6]. A new field in the IPv6 header is the "Flow Label" field; in the IETF (Internet Engineering Task Force), several arguments were presented regarding the purpose of this field. The questions posed to the IETF designers were as follows: Was it to be key in handling fast switching? Was it to be meaningful to applications and used to specify quality of service? Must it be set by the sending host? Could it be set by routers? Could it be modified in route? Must it be delivered with no change? [7] Because of these uncertainties, as well as more urgent work in other areas, the IPv6 flow label was ignored by implementers, and today it is set to zero in almost every IPv6 packet [7]. Due to this reason, several proposals to use the IPv6 flow label field have been made for different purposes such as QoS support, mobility, identifying IPv4-in-IPv6 tunnels, load balancing, packet filtering, security and packet switching. A study of these proposals is presented in [7, 8].

However, an important problem in the current Internet is the use of the shortest path routing algorithm, which leads to congestion of certain common paths for many communications. One solution to this problem is to use switching technologies that allow traffic engineering support. Adopting MPLS [9], one of the most used switching technologies, would require the use of an additional transport layer (layer 2.5).

In this paper, we present a new proposal for the use of the IPv6 flow label for traffic engineering support, called

✉ Line Y. Becerra Sánchez
line.becerra@ucp.edu.co

Jhon J. Padilla Aguilar
jhon.padilla@upb.edu.co

1 Universidad Católica de Pereira, Pereira, Colombia

2 Universidad Pontificia Bolivariana, Bucaramanga, Colombia

PSA-TE6 (Packet Switching Architecture to Support Traffic Engineering in IPv6 Networks). The goal of this architecture is to use the IPv6 flow label field for packet switching in IPv6 networks in a similar manner as MPLS [9] but without the need of an MPLS architecture being installed.

PSA-TE6 takes advantage of the location of the IPv6 flow label field in the first 64 bits of the IPv6 header, as well as the DS and Hop Limit fields; the latter two are also being mapped by MPLS when working over IP (IP/MPLS). This proposal is created as a result of studies of the IPv6 Flow Label field, from its creation to the latest recommendations of the IETF, as well as from analysis of several proposals of use cases of the IPv6 Flow Label field for different purposes; such studies are described in [7, 8]. It also considers the great similarity of the flow label structure to that of the MPLS label with respect to size (20 bits) and contents. The goal of PSA-TE6 is to facilitate the provision of traffic engineering at the IP level in next-generation networks.

In this paper, we first present a summary of the works related to the IPv6 flow label and traffic engineering taxonomy. To evaluate PSA-TE6, we analyze the cost of using IPv6 flow labels during packet switching in situations in which the label stacking is used. We present three evaluations in terms of the label space reduction, label stacking cost and minimization of label stacking costs.

This paper is organized as follows: Sect. 2 describes the related works. Section 3 explains the packet switching architecture to support traffic engineering in IPv6 Networks. Section 4 presents the evaluations and results of label space, stacking costs and minimization of label stacking costs. Finally, Sect. 5 describes conclusions and future works.

2 Related works

2.1 Related specifications with the IPv6 Flow label

The flow label field was initially specified with 28-bit length in RFC1710 [10]; then, it was reduced to 24 bits by RFC1883 [11] and finally established as 20 bits by RFC2460 [2]. Since the definition of this field, there have been several uncertainties about its use such that, in RFC2460, it was defined as experimental and subject to change [7].

The IETF performed several preliminary works to its full specification, such as in [12–15], until a more detailed specification was published in RFC3697 [16], which provided useful information for its use and was adopted for approximately 7 years. Meanwhile, several solutions were published by researchers to propose methodologies for the use of flow label field for various purposes, such as quality of service support, packet switching, and packet filtering, among others. Such solutions violated, in one manner or another, the recommendations given by RFC3697 [16].

Table 1 Evolution of the IPv6 flow label field specifications

Specification	Description
RFC1710 [10]	Simple internet protocol plus white paper
RF1883 [11]	Internet protocol, version 6 (IPv6)
RFC2460 [2]	Internet protocol, version 6 (IPv6) specification (obsolete)
RFC8200 [3]	Internet protocol, version 6 (IPv6) specification (current)
RFC3697 [16]	IPv6 flow label specification (obsolete)
RFC6294 [7]	Survey of proposed use cases for the IPv6 flow label
RFC6436 [17]	Rationale for update to the IPv6 flow label specification
RFC6437 [18]	IPv6 flow label specification (current)

Because of the above, the IETF performed a study of use cases for the “Flow Label,” published in RFC6294 [7]. Such a study was performed because they found that, although every IPv6 packet contained the flow label field, the field was not used in practice. Therefore, RFC6294 [7] describes various published proposals for the use of the flow label and showed the inconsistencies with the RFC3697 standard [16]. Then, the IETF took into account the minimal practical use of the flow label field at that moment, and the IETF was motivated to change that specification to clarify it and introduce some additional flexibility; as a result, RFC6436 [17] was published. Then, the IETF made a recommendation to update RFC3697 [16], and as a consequence, the new specification of the flow label field was published in RFC6437 [18], making RFC3697 [16] obsolete. In Table 1, the summary of the history of the flow label field specifications can be observed.

2.2 Proposals of use cases of the flow label field

Most of the proposals are inclined to use the flow label field to support QoS; some proposals focus on using it to support packet switching, others use the flow label to contribute to mobility functions, while other uses are given for load balancing, packet filtering, and security. In [8], a review of such a proposal is described.

In this paper, we provide a brief description of the proposals using the IPv6 Flow Label for packet switching. In [19, 20], the authors describe a proposal called “IPv6 Label Switching Architecture” (6LSA). In 6LSA, network components identify a flow reviewing the flow label field in the IPv6 packet header. All packets with the same flow label value must receive the same treatment and should be sent to the same hop.

However, 6LSA is similar to MPLS when considering that a label has significance only between 6LSA routers and when it establishes the flow label in every hop. Unlike the tra-

Table 2 Summary of proposals for the use of the IPv6 flow label field

Categories (purposes)	Proposals
Packet switching	[19–24]
QoS support	[25–33]
Mobility	[34–38]
IPv4-in-IPv6 tunnel identification	[39]
Load balancing	[40, 41]
Traffic filtering	[42, 43]
Security	[44–46]

ditional routing techniques, but similarly to MPLS, 6LSA packets are classified within an FEC (Forwarding Equivalence Class), and routers send packets over different paths depending on the FEC.

In [21], the authors propose a mix of the flow label and class of service fields as a switching tag and to support QoS in a manner similar to MPLS. This proposal uses a bit of the DiffServ Code Point (DSCP) RFC2474 [22] to indicate that the flow label is a switching tag. Another similar work based on QoS and packet switching is described in [23], which was designed as a hop-by-hop option. On the other hand, [24] presents a new model of sending packets by flow label to improve fast switching of IPv6 packets, which require service differentiation, and according to the author, provides more effective functions than MPLS.

Other use cases of the flow label field as QoS support are described in [25–33]. Uses for mobility are proposed in [34–38]. Additionally, the flow label was used to identify an IPv4-in-IPv6 tunnel in [39], as a tool for load balancing by equal Cost Multi-Path Routing in [40, 41], as a mechanism of traffic filtering in [42, 43], and for security purposes in [44–46]. In Table 2, we give a summary of the different proposals by category according to the purpose of each.

2.3 Proposals to provide internet traffic engineering

Internet traffic engineering is responsible for the optimization and evaluation of the performance of IP networks in operation; the aim is to improve the network performance by optimizing the use of resources and traffic by applying technologies and scientific principles to allow the measurement, characterization, modeling, and control of Internet traffic [47].

In the last 15 years, different solutions have been proposed to support Internet traffic engineering. A previous study over traffic engineering proposals is described in [48]. We organized these proposals in five categories: TE (Traffic Engineering) based on IP by link weight optimization, TE based on MPLS, TE based on LISP, TE based on segment routing, and TE based on IPv6 facilities.

2.3.1 TE based on IP

IP-based proposals are focused on the IGP link weight adjustment. The first IP-based TE solution was proposed by Fortz et al. in [49–51]. The main goals in their approach were to set the link weights of interior gateway protocols (IGPs), such as OSPF and IS–IS, according to the given network topology and traffic demand to control intradomain traffic and meet TE objectives. In [52–56], algorithms and optimization problems to set link weights are proposed. Alternatively, a generalized routing framework to realize the optimal TE, which can potentially be implemented via OSPF- or MPLS-based approaches, is presented in [57].

2.3.2 TE based on MPLS

The concept of traffic engineering in MPLS-based environments was introduced in [58, 59] by setting up dedicated switched paths (LSPs). The MPLS specifications are detailed in RFC3031 [9]. TE based on MPLS can provide an efficient paradigm for traffic optimization. The most distinct advantage of MPLS-based TE is its capability of explicit routing and arbitrary splitting of traffic, which is highly flexible for both routing and forwarding optimization purposes. Many solutions have been presented in the literature using MPLS for traffic engineering and QoS purposes; many of them are directed to propose constraint-based routing algorithms, and studies over these proposals have been presented in [60, 61].

2.3.3 TE based on LISP

In recent years, a completely different strategy has been proposed by the LISP protocol (Locator/ID Separation Protocol). The LISP Specifications are contemplated in RFC6830 [62]. LISP is a network-layer-based protocol that enables separation of IP addresses into two new numbering spaces: Routing Locators (RLOCs) and Endpoint Identifiers (EIDs). RLOCs are topologically assigned to network attachment points; these locators are used for routing and forwarding of packets through the network [62]. EIDs are assigned independently from the network topology; these identifiers are used for numbering devices and are along administrative boundaries [62]. In [63], the authors describe how LISP re-encapsulating tunnels can be used for traffic engineering purposes. Thus, a packet can take an administratively specified path, a congestion avoidance path, a failure recovery path, or multiple load-shared paths as it travels from the ITR (Ingress Tunnel Router) to the ETR (Egress Tunnel Router). By introducing an Explicit Locator Path (ELP), an ITR can encapsulate a packet to a Re-encapsulating Tunnel Router (RTR), which decapsulates the packet and then encapsulates it to the next locator in the ELP. Some documents dealing with traffic engineering support by LISP include [64–69].

2.3.4 TE based on segment routing

Segment Routing (SR) takes advantages of the source routing paradigm. The IETF is working on standardizing this architecture [70]. In SR a node steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based, and a segment can have a semantic local to an SR node or global within an SR domain. SR allows for enforcement of a flow through any topological path and service chain while maintaining per-flow state only at the ingress nodes to the SR domain [70].

Segment routing and MPLS SR can be directly applied to the MPLS architecture with no change on the forwarding plane. For this, a segment is encoded as an MPLS label, and an ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. Upon completion of a segment, the related label is popped from the stack [70].

Segment routing and IPv6 SR can be applied to the IPv6 architecture with a new type of routing header. A segment is encoded as an IPv6 address. An ordered list of segments is encoded as an ordered list of IPv6 addresses in the routing header. The active segment is indicated by the destination address of the packet. The next active segment is indicated by a pointer in the new routing header [70].

Segment routing for traffic engineering SR has been recently proposed as an alternative traffic engineering technology enabling relevant simplifications in control plane operations. Some works have been proposed to support traffic engineering by segment routing, such as in [71], which details the segment routing policy for traffic engineering. In [72], the authors consider the problem of determining the optimal parameters for segment routing in the offline and online cases. In [73], a Label Encoding Algorithm for MPLS Segment Routing is proposed. In [74], the authors present an SR path assignment algorithm for the flow assignment problem. Finally, in [75], the authors propose ILP models and heuristics that are successfully utilized to assess the TE performance of SR-based packet networks.

2.3.5 TE based on IPv6 facilities

These are proposals that use IPv6 issues such as the flow label and others. Here, there are allocated 6LSA [20] and IPngls [76]. The main advantage of this category of proposals is that there is no other necessary layer 2.5 technology to support TE. Additionally, they try to take advantage of benefits that are not yet being used for IPv6, which is still in deployment and will be the main protocol for the Internet in the future. Our proposal falls in this category. Because it has some similarities with the other mentioned proposals, we provide a summary of the similarities and differences according to important characteristics for supporting traffic engineering in Table 3.

3 Packet switching architecture to support traffic engineering in IPv6 networks (PSA-TE6)

PSA-TE6 is a new solution proposed to support traffic engineering in IPv6 networks. The goal of this architecture is to use the IPv6 flow label field for packet switching in IPv6 networks in a manner similar to how MPLS [9] works but without the need of an MPLS architecture being installed. This proposal is created as a result of the study of the IPv6 flow label field from its creation to the latest recommendations of the IETF. Also from analysis of various proposals for the use of the IPv6 flow label field (see [7, 8]) and by observing its structure, which shows a great similarity to the MPLS label regarding size (20 bits) and contents.

In the literature, two proposals for label switching have been presented; these are described in [20, 76]. In [76], the authors propose the forwarding of IPv6 packets using label switching techniques, with similar advantages to the Multiprotocol Label Switching (MPLS) architecture. This forwarding process is performed using a mapping of all MPLS header fields within the IPv6 header. Alternatively, [20] introduces an architectural framework to use the IPv6 packet header Flow Labels to set up labeled paths like MPLS. Although our proposal and the two mentioned above use the label switching concept via IPv6 flow label switching in a manner similar to how MPLS works, our proposal presents important differences that strengthen traffic engineering support in IPv6 networks. In Table 3, such differences are described.

3.1 Principles of design of the PSA-TE6 proposal

The design of PSA-TE6 started with the study of the IETF specifications for the IPv6 flow label in [18] and the use cases of the IPv6 flow label in [7]. Both documents describe three basic rules for the use of the IPv6 flow label field, which are as follows:

1. IPv6 nodes MUST NOT assume any mathematical or other properties of the flow label values assigned by source nodes [7, 18].
2. Router performance SHOULD NOT be dependent on the distribution of the flow label values. Specifically, the flow

Table 3 Comparison between proposals using the IPv6 flow label to switch packets

Characteristic	6LSA	IPNGLS	PSA-TE6
It uses a packet switching mechanism via the IPv6 flow label	Yes, it is	Yes, it is	Yes, it is
It describes the routing table fields	Yes, it is	No, it is not	Yes, it is
It uses the label switching paths like MPLS	Yes, it is	Yes, it is	Yes, it is
It splits the label value in different fields	Yes. It divides the flow label field into three parts	No. It maps the label value directly from MPLS	No. Uses the 20 bits without dividing it. The value is assigned in a similar manner as MPLS
It uses label operations: push-swap-pop	Yes, it is	It is not described. It is assumed like in MPLS	Yes, it is
It describes how the label is generated	Described three ways: 1. Locally based on a certain algorithm or policy. 2-In the entry packet like a flow label from the source node 3-Distributed through a label distribution process	No, it is not	Yes, it is distributed through a label distribution process in a similar manner as MPLS
It uses a label distribution protocol	It contemplates the option of using it for case 3 but does not assume one in particular	It contemplates the possibility of using it but does not assume one in particular	It contemplates using RSVP-TE
It defines the label-FEC relation in every router	Yes, it is	No, it is not	Yes, it is
It defines the operation within a domain and defines the elements that comprise it	Yes, it is	No, it is not	Yes, it is
It allows label stacking	No, it is not allowed	Yes, it is allowed by an IPv6 option header	Yes, it is allowed by generic packet tunneling in IPv6 or using an IPv6 option header
It uses extended routing protocols to support traffic engineering	No, it is not	No, it is not	Yes, it is
It uses constraint-based routing algorithms	No, it is not	No, it is not	Yes, it is
It defines a flow label restoration mechanism	No, it is not	No, it is not	Yes, it is

label bits only make poor material for a hash key [7, 18].

3. The flow label must not be changed in route but allow routers to set the label on behalf of hosts that do not do so [7, 18].

According to the fundamental rules mentioned above, our proposal would violate the first part of rule (3), i.e., "not to be changed in route." However, the second part of rule (3) states: "but allow routers to set the label on behalf of hosts that do not do so," which is an open issue discussed in RFC6294 [7].

Additionally, RFC6294 mentioned with respect to rule (3) that it does not exclude the Flow Label from being used for switching or routing purposes.

Similarly, RFC6294 makes recommendations for designers who use the IPv6 flow label for packet switching. Such recommendations refer to overlooking the rules within a given domain. Within that domain, routers could establish and interpret the IPv6 flow label field as it was designed, and then in the router of the last hop of the domain, the label should be set to zero; this rule should be enforced for packets arriving at the domain with the label set as zero.

For the case in which packets arrive at the domain with a label value other than zero, an alternative recommendation given in RFC6294 [7] is to define a hop-by-hop option header to carry the original label through the domain so that it can be restored at the output of the domain. All those recommendations were taken into account in the design of the proposed PSA-TE6 solution.

On the other hand, OSPF and IS–IS protocols were extended in response to the requirements of RFC2702 [58], and these extensions are also associated with support MPLS traffic engineering (OSPF-TE and IS–IS–TE). Therefore, in the design of PSA-TE6, such protocols also play an important role in the forwarding process since PSA-TE6 uses label switching and its goal is to provide traffic engineering. Thus, our PSA-TE6 proposal includes the use of the OSPFv3-TE [77] protocol, which is the extended protocol for working on IPv6 networks and supporting traffic engineering. Additionally, the RSVP-TE signaling protocol [78], which was extended to the MPLS label distribution, has been taken and defined as a tool for the distribution of IPv6 flow labels in our proposal.

Finally, in the PSA-TE6 proposal, it is necessary to find appropriate constraint-based paths. This functionality is provided by CBR (Constraint-based Routing) algorithms, which select the best route that corresponds to the constraint set. Restrictions can be imposed by administrative policies, quality of service or traffic engineering requirements [79]. In the last 15 years, many CBR algorithms have been proposed, and in [60, 79, 80], a study is presented. We have selected the CSPF (Constraints Shortest Path First) algorithm for use in our proposal because CSPF is one of the most common algorithms used to address this issue [81].

3.2 Architecture of the PSA-TE6 proposal

The PSA-TE6 architecture is composed of the following elements (see Fig. 1): ingress/egress 6DER (Ingress/Egress PSA-TE6 Domain Edge Router); 6DTR (PSA-TE6 Domain Transit Router); and 6DLSP (PSA-TE6 Domain Label Switching Paths). These elements must operate under a PSA-TE6 domain, which we have denominated 6D. The idea of having a domain with PSA-TE6 is referred at RFC6294 [7] in Sect. 4, as a recommendation to proposals using the IPv6 flow labels to switch packets, as stated previously.

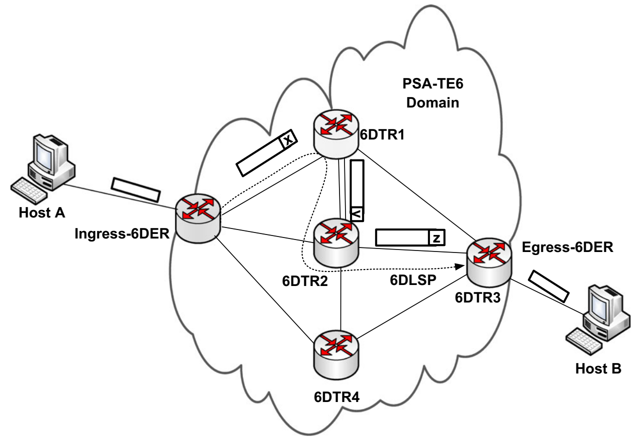

Fig. 1 PSA-TE6 domain

To explain the PSA-TE6 proposal, we assume a network with five nodes as in Fig. 1. It is a network that works under the IPv6 protocol, and it is composed of a 6D domain with routing and signaling protocols as OSPFv3-TE [77] and RSVP-TE [78], respectively. OSPFv3-TE is responsible for routing information flooding regarding the network topology and traffic engineering information. Meanwhile, RSVP-TE is responsible for the establishment of the label-switched paths and the distribution of IPv6 flow label values. It is important to emphasize that the RSVP-TE protocol has an object named "Label," which was created to distribute MPLS labels; however, in our solution, we use the "Label" object of RSVP-TE to distribute IPv6 flow labels. The above can be done because the MPLS labels and IPv6 flow labels have the same length (20 bits). Besides, to establish IPv6 label-switched paths, it is necessary to find an appropriate path not necessarily the shortest, but one based on constraints. Several algorithms have been proposed for this, but in this proposal, we assume that PSA-TE6 works with a CSPF algorithm (Constrained Shortest Path First), which is one of the most used. CSPF is based on the Dijkstra algorithm with a modification: the addition of a bandwidth constraint; this algorithm is explained in [81].

In the 6D domain, Ingress/Egress-6DER routers must be able to read the IPv6 header; they also should set the 6DLSPs (IPv6 Flow Label Switching Paths) and perform label insertion and deletion operations (push and pop operations). Meanwhile, 6DTRs routers must be able to read the first 64 bits of the IPv6 header (see Fig. 2) to switch and to route packets through exchanges of IPv6 flow labels, and they also should be able to do the necessary operations for IPv6 label stacking. The fact that the 6DTRs routers have to read the first 64 bits of the IPv6 header is because the DS (Differentiated Services), IPv6 flow label and Hop Limit fields

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|     DS    |ECN|               Flow Label               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Payload Length          |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 2 IPv6 header RFC2460 [2], RFC8200 [3] and RFC2474 [22]

of the packet are located in this part of the IPv6 header and must be known to support quality of service and label packet switching (see Fig. 2). It is possible to do this in one reading operation with the currently used 64-bit network processors.

3.3 Process to establish 6DLSPs

When it is necessary to establish a new communication and the first packet reaches the Ingress-6DER, it reads the IPv6 header and captures the source and destination address. Then, by means of RSVP-TE, the establishment of the label switching path and label distribution process are accomplished using the standard procedures of such protocols. The label switching path is found by the constraint-based routing algorithm based on the OSPFv3-TE information. Then, the ingress-6DER puts the label value in the IPv6 flow label field in all the packets belonging to the correspondent flow and sends them to the next hop. This proposal initially assumes that packets come from a domain that does not use the IPv6 flow label so that this value will be zero according to RFC6437 [18]. Then, the packet travels on that path, and each 6DTR interior router will exchange the label (swap operation) in each packet and then sends the packet to the appropriate output interface. When the packet arrives at the Egress-6DER, it removes the label (pop operation) and sends the packet to the destination. The router also returns the flow label field to its original value (or zero) according to RFC6294.

3.4 Information bases required in PSA-TE6

In the packet forwarding process, it is necessary to have information for the operations performed on the IPv6 flow label field, which must be analyzed before packet forwarding to the next hop. For this proposal, routers have a FIB (Forwarding Information Base) that is specific for each router, i.e., if it is 6DER or 6DTR. This FIB can be of two types: one that maps a FEC (Forwarding Equivalent Class) to N6FLD (Next Hop IPv6 Flow Label Forwarding Data), which we have called FTN6 (Forwarding Equivalence Class To Next Hop IPv6 Flow Label Forwarding data), and another FIB that maps an incoming IPv6 flow label to N6FLD, which we have called I6LTN (incoming IPv6 flow label to next Hop IPv6 flow label Forwarding data). These tables are similar in content and function to those used in MPLS [9]. These tables will be available in routers of IPv6 flow label switching according to their roles in the PSA-TE6 domain. In IP networks, a router considers that two packets belong to the same FEC if there is any address prefix X in the routing table such that X prefix is the longest match for each packet's destination address. In the PSA-TE6 architecture, the FEC will be determined in the ingress routers where the 6DLSPs are established.

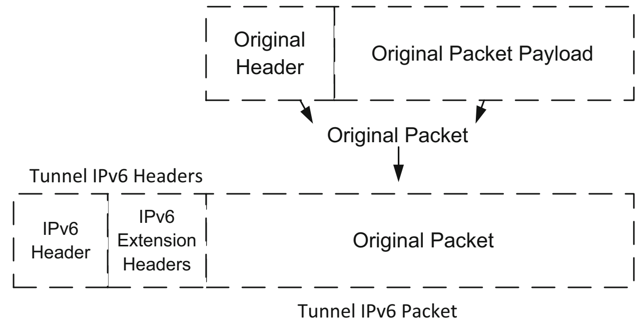

Fig. 3 Encapsulating a packet through generic packet tunneling RFC2473 [83]

3.5 IPv6 flow label stacking

In PSA-TE6, it is necessary to explore mechanisms to stack IPv6 flow labels as a technique to improve management of the limited label space, considering that the size of the IPv6 flow label field is only 20 bits. In MPLS, the same problem occurs, and the solution has been studied and evaluated by creating tunnels [82]. Therefore, in PSA-TE6, we propose two solutions for creating tunnels to add traffic of different 6DLSPs in common segments of the network with two or more internal routers. To make the process of label stacking, two mechanisms are proposed: the first is the IPv6 tunneling by Generic Packet Tunneling in IPv6 [83] (see Fig. 3), and the second method is by using an option header, similar to that described in [76].

3.5.1 Label stacking using generic packet tunneling for IPv6

This is a technique to establish a virtual path between two nodes using IPv6 encapsulation, and it is widely used in different scenarios, such as in IP mobile networks. In this method, to handle the label stacking and thus have those routers understand that there is label stacking, this mechanism will use the Next Header field, which indicates that the next header is another IPv6 header (Next Header = 41 for IPv6 in IPv6 tunneling) [2]. With this tunneling mechanism, the packet switching process via the IPv6 flow label can continue in the outer IPv6 tunnel header, preserving the switching speed and the same forwarding process [83]. IPv6 encapsulation consists of prepending to the original packet an IPv6 header and, optionally, a set of IPv6 extension headers, which are collectively called tunnel IPv6 headers. The encapsulation occurs in an IPv6 tunnel entry point node as the result of an original packet being forwarded onto the virtual link represented by the tunnel (see Fig. 3).

3.5.2 Label stacking using a hop-by-hop option header

This method is referred to support IPv6 option headers [2]. The option header that fits for this purpose is the hop-by-hop option header (coded for Next header = 00, as was described in [2]). As specified in RFC2460, extension headers are not examined by any node along the route except for the hop-by-hop header that can be processed and analyzed by each node along the path of the packet. When the hop-by-hop option header is present, it must be followed immediately by the IPv6 header. Each option header has a length that is an integer multiple of 8 octets.

The process of using the hop-by-hop option header to create tunnels or label stacking is as follows: When a packet arrives at the tunnel ingress point, it creates a hop-by-hop option header, which will save the label value with which the packet comes, and then it puts the new label value in the IPv6 flow label field of the IPv6 header. That label will be the outermost label, and it will be used to perform the switching to the next router based on the FIBs information. This allows for continued label switching on the first 64 bits of the IPv6 header in the PSA-TE6 architecture, maintaining the same delivery speed as when there is no stacking along of the path. A similar proposal was initially described in [76]. Figure 4 shows the format of the hop-by-hop option header used for label stacking in the PSA-TE6 architecture.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  | Hdr Ext Len=1 |Option Type=X | Opt Data Len=Y |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Original IPv6 Flow Label          |     Future Use    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IPv6 Flow Label 2              |     Future Use    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IPv6 Flow Label 3              |     Future Use    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IPv6 Flow Label 4              |     Future Use    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 4 Hop-by-hop option header with IPv6 flow label stacking for PSA-TE6

4 Evaluation of the PSA-TE6 solution

In this section, we present three evaluations of the PSA-TE6 solution. We have oriented such evaluations towards label space reduction and label stacking cost analysis and minimization of label stacking cost. We consider in our proposal that it is important to probe such issues since, in MPLS, those were also topics to be solved. Each evaluation and its results are described in the following sections.

4.1 Evaluation of the label space reduction in PSA-TE6

Here, we analyze the behavior of PSA-TE6 when it uses label stacking and when label stacking is not used. This analysis is necessary and important because the IPv6 flow label field has only 20 bits. Therefore, saving of label space is required. The label stacking process by tunneling is a useful solution to reduce the label space when a network has segments that are common for many communications, allowing the use of fewer labels in the forwarding process. To evaluate such situations, we perform the abstraction of a network topology through which three flows are sent; the establishment of tunnels in common segments of the IPv6 flow label switched path for those flows is shown, and the label number involved in the forwarding process is calculated.

In Fig. 5, there is a simple example of three communications being sent by nodes I, A, and M to the nodes J, G, and N, which are the ingress and egress nodes of the PSA-TE6 architecture. As stated above, in this proposal, the label switching paths followed for these flows are called 6DLSPs (PSA-TE6 Domain Label Switching Paths). In the case of not using tunneling, the label-switched paths are the only ones established, which correspond to level 1 (L = 1; L will be used to assign different levels of stacking). If the label stacking is enabled, in this example, there could be established two level-2 tunnels (L = 2) at maximum, which are: B–F that tunneled A–G and I–J flows. The other is tunnel C–E that establishes a level-2 tunnel regarding M–N and a level-3 tunnel, which would also be in the C–E segment with respect to B–F and, in turn, the A–G and I–J flows. Table 4 shows the number of labels used for the cases when there is no tunneling enabled and when tunneling is enabled.

4.1.1 Analysis of results of the labels' reduced space in PSA-TE6

As shown in Fig. 5 and in the data of Table 4, for the three communications, there are segments that are common, and

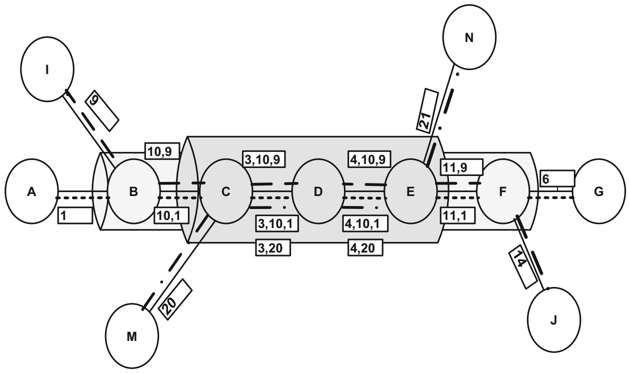

Fig. 5 Possible tunnels formed for three communications

Table 4 Total number of labels used with and without label stacking

State	Tunnels	Paths	# labels
No stacking	L = 1	A–G	16
		I–J	
		M–N	
With stacking	L = 2, 3	A–G/B–F/C–E	10
		I–J/B–F/C–E	
		M–N/C–E	

there are tunnels of a level higher than 1 (L > 1) that were established. Table 4 shows the number of labels used for the cases when there is no tunneling enabled and when tunneling is enabled. It is observed in the example in Fig. 5 that the stacking process allows considerable savings in label space (37.5%).

Table 5 List of notations and operations

C_{in}	Cost of operations in ingress nodes. At the ingress nodes, Ingress-6DER conducts mapping operations of an unlabeled packet to an FEC (FTN6 table), and then it consults N6FLD data and puts the label on the packet
C_{eg}	Cost of operations in egress nodes. In Egress routers, Egress-6DER performs a lookup of the label in the I6LTN table, and then it consults N6FLD data, deletes the label and passes the packet to the upper layers
C_{tr}	Cost of operations in transit nodes. The transit nodes, 6DTRs, conduct search operations of the label in the I6LTN table, consulting N6FLD data, and swapping the top label, i.e., they delete the top label and put a new label
C_{push}	Cost of operations in entry point nodes of the tunnel and operations to push a new label when label stacking is enabled. Push operations include searching the label in the I6LTN table to consult the N6FLD data and to put a new label value on the top of the label stack
C_{pop}	Cost of operations in exit point nodes of the tunnel and deletion of the top label within the label stack. Pop operations include lookups of the label in the I6LTN table to consult N6FLD data, deletion of the top label from the label stack, searching for the new top label in the I6LTN table, and consulting the N6FLD data again
$NT_{L>1}$	Number of tunnels present in each level L
Ntr_L	Number of transit nodes of the level L tunnel
N_L	Number of nodes of the level L tunnel
$/N_L, Ntr_{L-1}/$	It is the number of nodes of the level L tunnel that are transit nodes of the level L-1 tunnel

4.2 Evaluation of label stacking cost

To evaluate the label stacking cost for different stacking levels, and to have an approximation of how the PSA-TE6 architecture operates in this aspect, an analysis regarding the operation cost in each router according to their role within the PSA-TE6 architecture has been realized. The same analysis is done with IP/MPLS to compare results. Both architectures have routers that play the same role in the path establishing process. Additionally, the creation of tunnels by stacking signaling protocols, routing algorithms based on restrictions and extended routing protocols for traffic engineering are similar, as is described in Sect. 3.2. It is not necessary to compare our proposal with the 6LSA and IPNGLS proposals (described in Table 3) because they are not standardized at the moment. This is why we compared our proposal only with MPLS, as it is a widely applied standard.

To determine the operating costs in each router, the following assumptions have been made: to each row reading in a table of the respective router has been assigned a value of 1 operation. We assume that the evaluated architectures work with the same search method in routing tables, so there is no difference in the lookup process. Thus, this aspect does not affect the cost difference between both studied technologies, and therefore, it was not taken into account in our models. In the case of inserting or deleting packet headers, it has been assigned a value of 1 for every 64 bits, in reading or writing operations, assuming that routers operate with 64-bit words (the first 64 bits of the IPv6 header include the DS (Differentiated Services), IPv6 flow label and hop limit fields, as seen in Fig. 2). Every operation has an associated unit of time, so the time is directly proportional to the cost. Then, the cost parameters, operations, and definitions are described in Table 5. Moreover, the costs and their location according to the roles of the routers in the forwarding process are shown in Fig. 6.

In Tables 6, 7 and 8, the cost values can be observed for each case according to the operations performed in each router. Costs for the PSA-TE6 architecture are in Tables 6 and 7 for both label stacking mechanisms explained in the previous section (GPT and hop-by-hop options header). The operation costs of the IP/MPLS are described in Table 8.

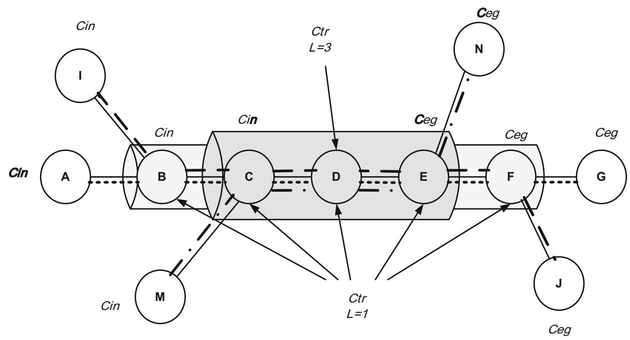

Fig. 6 Location of costs according to the role in routers

Costs are calculated based on the number of required memory reads/writes, which are specified as the number of operations in parentheses (op).

Analysis of common costs in the forwarding process The ingress and egress costs (C_{in}, C_{eg}) are present in the tunnels of level 1, in the edge routers of the PSA-TE6 architecture, and in the entry and exit point nodes of the tunnels with a level higher than 1 (L > 1). Costs of label exchange in transit nodes are also present at all tunneling levels. To calculate the total cost of transit nodes, we must know the number of transit nodes from the respective path, which is represented in the equations by Ntr_L=1,2,3. The Eqs. (1), (2) and (3) described below take into account the costs that are common to all tunnels in the forwarding process for the PSA-TE6 and IP/MPLS solutions. For the PSA-TE6 architecture with both stacking mechanisms explained in the previous section, Generic Packet Tunneling and by the IPv6 hop-by-hop options header, have been called $Cost_{PSA\text{-}TE6\_GPT}$ and $Cost_{PSA\text{-}TE6\text{-}HBH}$, respectively, which are described in Eqs. (1) and (2). For IP/MPLS, Eq. (3) describes $Cost_{IP/MPLS}$. The costs of Eqs. (1), (2) and (3) have been calculated by taking into account the values determined in Tables 6, 7 and 8, which were located as coefficients. *NT* is the number of tunnels for each level being analyzed.

$$\underset{L=1,2,3...}{Cost_{PSA-TE6\_GPT}} = \left(2C_{in} + 2Ctr * \left(Ntr_{L=1,2,3}\right) + 2C_{eg}\right) * NT_{L=1,2,3..} \tag{1}$$

$$\underset{L=1,2,3...}{Cost_{PSA-TE6\_HBH}} = \left(2C_{in} + 2Ctr * \left(Ntr_{L=1,2,3}\right) + 2C_{eg}\right) * NT_{L=1,2,3..} \tag{2}$$

$$\underset{L=1,2,3...}{Cost_{IP/MPLS}} = \left(4C_{in} + 3Ctr * \left(Ntr_{L=1,2,3}\right) + 4C_{eg}\right) * NT_{L=1,2,3..} \tag{3}$$

Analysis of the label stacking costs Now, we will describe the costs present at the endpoints of the tunnel. Costs related to push and pop operations (C_{push} and C_{pop}) are only part of the label stacking process or tunnels for more than 1 level (L > 1). When label stacking is done, the cost associated with the transit nodes is considered only for the transit nodes belonging to the upper tunnel. Therefore, an adjustment in the number of transit nodes is necessary for its calculation according to existing tunnels that have more than 1 level with respect to the tunnels that are tunneled within these. To make the adjustment, it is necessary to know the number of nodes in the upper tunnel, which are transit nodes in the lower-level tunnels, and then we should subtract the cost associated with that number of nodes. This operation is denoted as $|N_L, Ntr_{L-1}|$ for this analysis. As an example, the number of nodes of the level-2 tunnel, L = 2, in the tunnel B–F of the Fig. 6, is equal to 5, which are the number of transit nodes in tunnel level 1 (tunnel A-G). This number of nodes (5 nodes) multiplied by the associated cost should be subtracted in Eq. (4) since they were taken into account in Eq. 1 and are now tunneled [similarly for Eqs. (5) and (6)]. The calculation of the

Table 6 Operation cost in the PSA-TE6 solution when generic packet tunneling is used

Cost	Operations	Value
PSA-TE6 with GPT		
C_{in}	Search in FIB of FTN6 type FEC-to-N6HLFE (1 op.)	2
	Packet labeling (put the label value in the IPv6 flow label field) (1 Op.)	
C_{eg}	Search for the label in the FIB table I6LTN type (1 op.)	2
	Deleting label (pop operation, to put in zero IPv6 flow label field) (1 op.)	
C_{tr}	Search for the label in the I6LTN Table (1 op.)	2
	Exchange of label (swap operation, put a new, modification of the IPv6 flow label field of the top header) (1 op.)	
C_{push}	Search for the label in the I6LTN Table (1 op.)IPv6 Encapsulation (push operation) (5 op.)	6
	This operations number is calculated taking into account that the IPv6 Header has 40 bytes and each writing operation uses 64 bits (8 bytes), therefore to write 40 bytes are performed 5 writing operations.	
C_{pop}	Search for the label in the I6LTN Table (1 op.)	8
	Decapsulation of the IPv6 Header (pop operation) (5 op.) This operations number is calculated as 40 bytes of the IPv6 header divided in 8 bytes (64 bits in each reading operation)	
	Search for the new top label of the stack in I6LTN, consulting N6HFLD (1 op), put an outgoing label (1 op.)	

Table 7 Operation cost in the PSA-TE6 solution when the hop-by-hop option header is used

Cost	Operations	Value
PSA-TE6 with hop-by-hop		
C_{in}	Search in FIB of FTN6 type FEC-to-N6HLFE (1 op.)	2
	Packet labeling (putt the label value in the Ipv6 flow label field) (1 op.)	
C_{eg}	Search for the label in the FIB table of I6LTN type (1 op.)	2
	Deleting label (pop operation, set to zero the IPv6 flow label field) (1 op.)	
C_{tr}	Search for the label in the I6LTN Table (1 op.)	2
	Label exchange (swap operation, to put a new label, modification of the IPv6 flow label on the top header) (1 op.)	
C_{push}	Search for the label into the I6LTN Table (1 op.)	4
	Push operation, router copies the original label value of the packet in the hop-by-hop option header (2 op.), and it puts the new label in the IPv6 header (1 op.)	
C_{pop}	Search the label in the I6LTN Table (1 op.)	6
	Pop operation (Reading of the last label of the stack in the option header (1 op.), writing of the new value of the length of the option header (1 op.), writing of the read label in the flow label field of the IPv6 header (1 op.)	
	Search the top label of the stack in the I6LTN table consulting the N6HFLD (1 op.). Router puts an outgoing label (1 op.)	

push, pop and transit costs are defined in Eqs. (4), (5) and (6) for the PSA-TE6 proposal and IP/MPLS architecture. In the equations presented in this analysis, the cost must be multiplied by the number of tunnels present in each level, and this number is denoted by $NT_{L>1}$.

$$Cost_{\substack{PSA-TE6\_GPT \\ L>1}} = \left(6C_{push} - 2Ctr * |N_L, Ntr_{L-1}| + 8C_{pop}\right) * NT_{L>1} \quad (4)$$

Table 8 Operation cost in IP/MPLS

Cost	Operations	Value
IP/MPLS		
C_{in}	Search in the FIB table of FTN type (1 op)	4
	Mapping the TTL and DS fields from the IPv6 header to the MPLS header (reading and writing) (2 op.)	
	Packet labeling (Insertion of the MPLS header) (1 op.)	
C_{eg}	Search for the label in the ILM Table (1 op.)	4
	Deleting the label (pop operation, extraction of the MPLS header) (1 op.)	
	Mapping of the TTL field from the MPLS header to the IPv6 header (2 op.)	
C_{tr}	Searching for the label in the ILM Table (1 op.)	3
	Label Exchange (swap operation: extraction of the MPLS header (1 op.), insertion of the MPLS header with the outgoing label (1 op)	
C_{push}	Searching for the label in the ILM Table (1 op.)	2
	Push operation: put a new label value-insertion of the MPLS header. (1 op.)	
C_{pop}	Searching for the label in the ILM Table (1 op.)	4
	Pop operation: deleting the top label of the label stack-extraction of the MPLS header (1 op.)	
	Search the new top label of the stack in ILM consulting the NHFLE (1 op.). Adding the outgoing label (1 op.)	

$$Cost_{\substack{PSA-TE6\_HBH \\ L>1}} = \left(4C_{push} - 2Ctr * |N_L, Ntr_{L-1}| + 6C_{pop}\right) * NT_{L>1} \quad (5)$$

$$Cost_{\substack{IP/MPLS \\ L>1}} = \left(2C_{push} - 3Ctr * |N_L, Ntr_{L-1}| + 4C_{pop}\right) * NT_{L>1} \quad (6)$$

Total cost calculation in which tunnels are established over different stacking levels To calculate the total final cost, both parts in each case are added. These total costs are defined in Eqs. (7), (8) and (9), which are the sums of the common costs in the forwarding process in Eqs. (1–3) and the label stacking costs in Eqs. (4–6).

$$Total\mathrm{Cost}_{PSA-TE6\_GPT} = \mathrm{Cost}_{\substack{PSA-TE6\_GPT \\ L=1,2,3..}} + \mathrm{Cost}_{\substack{PSA-TE6\_GPT \\ L>1}} \quad (7)$$

$$Total\mathrm{Cost}_{PSA-TE6\_HBH} = \mathrm{Cost}_{\substack{PSA-TE6\_HBH \\ L=1,2,3..}} + \mathrm{Cost}_{\substack{PSA-TE6\_HBH \\ L>1}} \quad (8)$$

$$Total\mathrm{Cost}_{IP/MPLS} = \mathrm{Cost}_{\substack{IP/MPLS \\ L=1,2,3..}} + \mathrm{Cost}_{\substack{IP/MPLS \\ L>1}} \quad (9)$$

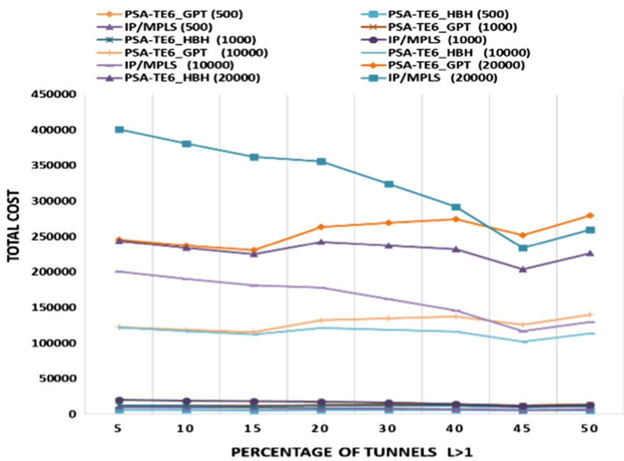

Fig. 7 Total costs for a different number of communications and percentages of level tunnels higher than 1, L > 1

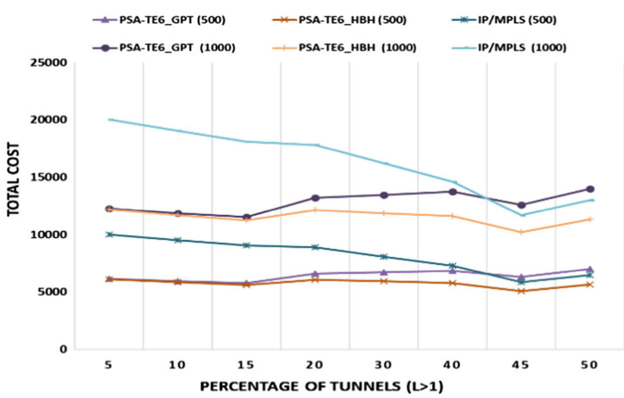

Fig. 8 Total number of communications costs for 500 and 1000 and percentages of level tunnels higher than 1, L > 1

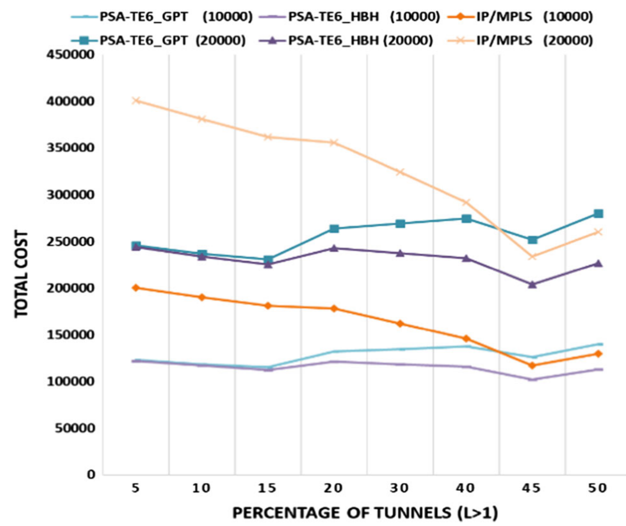

Fig. 9 Total costs for communications number 10,000 and 20,000 and percentages of level tunnels higher than 1, L > 1

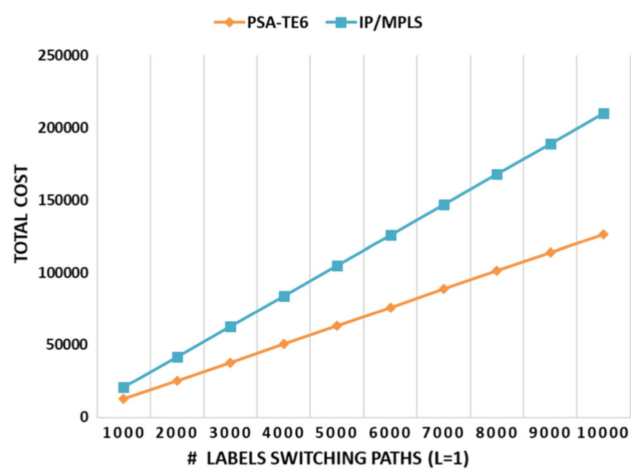

Fig. 10 Total cost of PSA-TE6 Vs. IP/MPLS for tunnels L = 1

4.2.1 Analysis of results for stacking model costs

Figures 7, 8 and 9 show the results obtained for the costs of the PSA-TE6 architecture using two mechanisms for label stacking: PSA-TE6-GPT and PSA-TE6-HBH. These proposals have been compared with the existing MPLS architecture when it works at the IP level (IP/MPLS). These data were obtained using Eqs. (1)–(9) in an algorithm coded in MATLAB [84] for 500, 1000, 10,000, and 20,000 communications and varying percentages of tunnels of levels higher than 1 (L > 1) that are present in the forwarding process, from 5 to 50%. Figure 7 shows the complete results (for 500, 1000, 10,000 and 20,000 communications), Fig. 8 shows partial results for 500 and 1000 communications, and, Fig. 9 shows partial results for 10,000 and 20,000 communications.

For this analysis, the tunnels with L > 1 have the same characteristics in regard to the level and number of transit nodes of the example of Fig. 5. Results in Figs. 7, 8 and 9 show that our PSA-TE6 proposal, using the two label stacking mechanisms, outperforms IP/MPLS when there is a presence up to 40% of tunnels with levels higher than 1. Additionally, such a result indicates that, when label stacking is not necessary, PSA-TE6 costs are approximately 45% lower than those of the IP/MPLS solution (see Fig. 10). When the number of tunnels of level higher than 1 is increasing, between 40% and 45%, the two solutions, PSA-TE6 and IP/MPLS, have similar costs, as is observed in Figs. 7, 8 and 9. These results are a consequence of a low value in ingress, transit and egress costs in PSA-TE6 (C_{in}=2, C_{eg}=2, C_{tr}=2) compared with IP/MPLS (C_{in}=4, C_{eg}=4, C_{tr}=3), as seen in Tables 6, 7 and 8.

When we compare GPT and HBH methods at evaluated scales (500, 1000, 10,000 and 20,000 communications), we observe the behavior is similar: HBH method is cheaper than GPT method. It is due to GPT method adds an IPv6 header in the encapsulation process, while HBH method uses an extension IPv6 header which is shorter than IPv6 header.

4.3 Minimization of label stacking cost

To optimize the label stacking cost for our PSA-TE6 proposal, an ILP formulation based on [85] is presented. We have already analyzed the costs of the forwarding process for the proposed PSA-TE6 solution and IP/MPLS. It is observed that PSA-TE6 has lower costs in the ingress, egress and transit nodes. In addition, the IP/MPLS solution has lower push and pop costs than the PSA-TE6 proposal (as we can see in Tables 6, 7, 8). Because of the above, the ILP formulation is presented to evaluate the effect of minimizing the total cost when the ingress and egress costs change, since these costs are present in both cases when the level of the tunnels is higher than 1 and when there is no label stacking, i.e., the proposal working in its base form. It is also evaluated whether, in the presence of stacking, there are changes in push costs. This evaluation is important because the phenomenon of the tunnels is quite common, and it is the solution more used to reduce the label space in proposals using label packet switching. It is essential to know the effect of push costs, as these are present in tunnels higher than 1. We perform tests with up to level-3 tunnels since the typical values for the nesting number range go from 1 to 3, as is expressed in [86].

For ILP formulation, it is assumed that the label switching paths (or 6DLSPs) for our PSA-TE6 proposal supporting end-to-end traffic flows were established and selected a priori according to a constraint-based routing algorithm such as CSPF [81]. In Table 9, the list of notations for ILP formulation is given, and in Table 10, the ILP formulation is given. Cost parameters, operations, and definitions were previously described in Table 5.

The ILP problem consists of selecting the set of 6DLSPs to support the given set of end-to-end traffic flows and their hierarchical level. The objective function in Eq. 10 is the minimization of the total cost of node operations for different levels of tunneling. The constraint in Eq. 11 forces the level selection of each 6DLSP on the P1 set (see Table 10). The constraint in Eq. 12 imposes that only a candidate LSP can be selected at any stack level for a given pair source–destination. The constraint in Eq. 13 imposes that a candidate 6DLSP can create a tunnel through a candidate path of the highest level only if both 6DLSPs have been selected. The restriction in Eq. 14 requires that any selected 6DLSP may be assigned to only one higher level LSP. Variables must be Boolean since only one 6DLSP can exist for a given path for a source–destination pair in Eq. 15, and each LSP can create a tunnel at most through a higher level LSP on a given link in Eq. 16.

The testing topology for the ILP formulation is shown in Fig. 11. This topology represents an IPv6 network to support intra-domain traffic engineering with PSA-TE6. Every node represents an IPv6 router in a PSA-TE6 domain. We have

Table 9 List of notations

SD_PAIRS:	Set of source and destination pairs of traffic flows or FECs
P1:	Set of 6DLSPs paths, where pi represents the path between the source–destination pair that supports traffic flows end to end, where i ϵ SD_PAIRS
P2:	It is the set of candidate paths of the upper level to 1 (P2 = {Ø} where L = 1, i.e., without label stacking).
P:	P1 ∪ P2
i, j:	Indices for 6DLSPs
l :	Index for the stack level
|pi,pj|:	It is a function that takes the nodes in the path pi and returns the number of those that are intermediaries or transit nodes on the path pj. Thereby, |pi, pi| is the transit node number in the path pi. |pi, pj| is the node number of pi that are transit nodes in pj
L:	Maximum label stack depth
δ_i^l:	Binary variable, 1 if the candidate 6DLSP is selected for level l, otherwise 0. (pi ϵ P)
$\gamma_{i,j}^l$:	Binary variable, 1 if the candidate 6DLSP j of level l-1 (pj ϵ P) is tunneled through of the candidate 6DLSP of level l (p_i ϵ P, $p_i \neq p_j$)
C_i:	Ingress node operations cost
C_{eg}:	Egress node operations cost
C_{tr}:	Transit node operations cost
C_{push}:	Operation cost of pushing a new label
C_{pop}:	Operation cost of popping a label

Table 10 ILP formulation

Objective:

$$Min : \sum_{l=1}^{L} \sum_{p_i \in P} (C_{tr} * |p_i, p_i| + C_{in} + C_{eg}) * \delta_i^l +$$

$$\sum_{p_i \in P} \sum_{r=1}^{2} \sum_{\substack{p_j \in P_r : \\ p_j \supset p_i}} \sum_{l=r+1} (C_{push} + C_{pop} - C_{tr} * |p_i, p_j|) * \gamma_j^l \quad (10)$$

Constraints:

$$\sum_{l=1}^{L} \delta_i^l = 1 \; \forall_i : p_i \in P_1 \quad (11)$$

$$\sum_{l=2}^{L} \delta_i^l \leq 1 \; \forall_i : p_i \in P_2 \quad (12)$$

$$2\gamma_{i,j}^l \leq \delta_i^l + \delta_j^{l-1} \; \forall p_i \in P, \forall p_j \in P_r : p_j \supset p_i \quad (13)$$

$$\forall l = r+1, \ldots, L, \quad \forall r = 1, 2$$

$$\sum_{r=1}^{2} \sum_{l=r+1}^{L} \sum_{\substack{p_i \in P_r \\ p_i \subset p_j}} \gamma_{i,j}^l \leq \sum_{l=s}^{L} \delta_j^l \; \forall p_j \in P_s, \quad \forall s = 1, 2 \quad (14)$$

$$\delta_i^l \in \{0, 1\} \; \forall p_i \in P_r, \forall l = r, \ldots, L, \quad \forall r = 1, 2 \quad (15)$$

$$\gamma_{i,j}^l \in \{0, 1\} \quad \forall p_i \in P, \forall p_j \in P_r : p_j \supset p_i \quad (16)$$

$$\forall l = r+1, \ldots, L, \quad \forall r = 1, 2$$

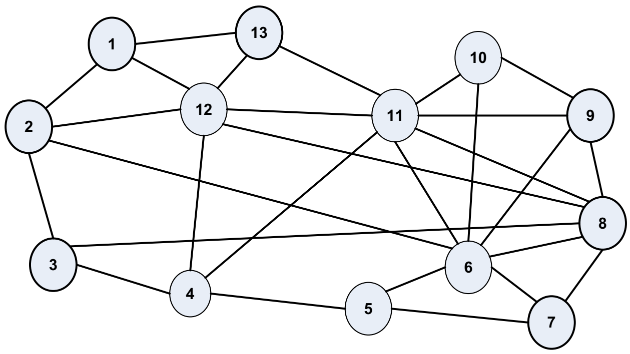

Fig. 11 Test network

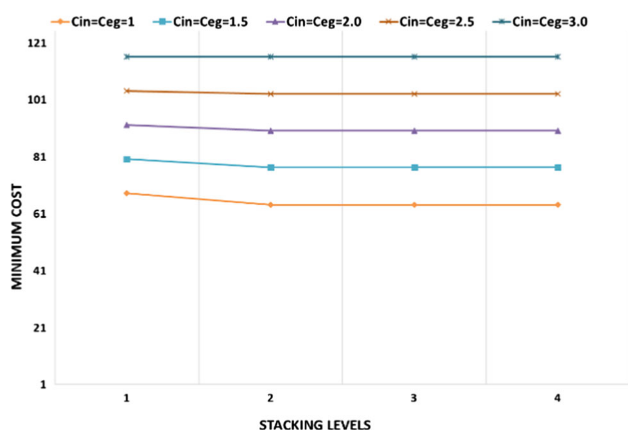

Fig. 12 Minimization of the cost in the forwarding process for several stacking levels with varying ingress and egress costs

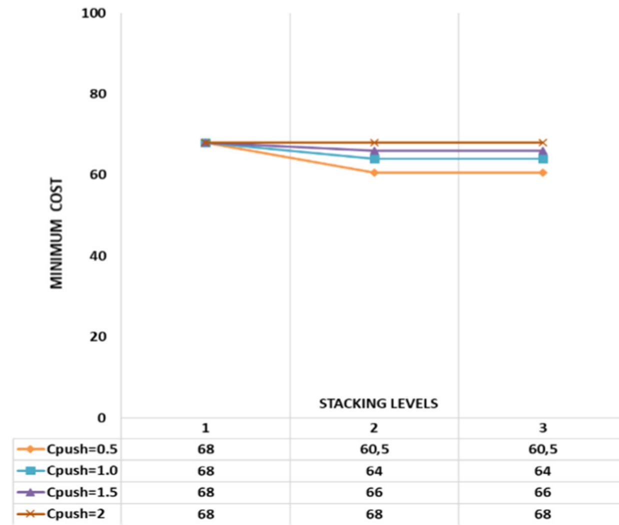

Fig. 13 Minimization of the forwarding cost for changes of C_{push} for three stacking levels

selected 1, 2, 3, and 13 as ingress nodes and 7, 8 and 9 as egress nodes.

4.3.1 Analysis of the results of the optimization model

We use the AMPL tool [87] and CPLEX solver [88] to run the ILP formulation using the test network of 13 nodes shown in Fig. 11. In Fig. 12, the minimum cost versus stacking levels is shown for different values of ingress and egress costs, with the remaining costs normalized to 1 (C_{tr}=C_{push}=C_{pop}=1). This test was performed to evaluate the effect of increasing the ingress and egress costs in a label packet switching network as our proposal PSA-TE6 and IP/MPLS. Results show that, by increasing the ingress and egress costs, there is a resulting increase in the minimum total cost for both cases: when it works in its base form, i.e., when there are only label-switched paths (L = 1) and when there is the presence of tunnels at different levels (L > 1) (see Fig. 12). As mentioned before, the ingress and egress costs are common costs of the forwarding process, and they are present for both methods when the label stacking process is not enabled and when label stacking is enabled.

In Fig. 13, the results obtained with the ILP formulation are shown, but in this case, the push cost values are changed from 0.5 to 2 in 0.5 steps. Push costs are only present in the stacking process. For this scenario, it can be noticed that, when all costs are equal (equal to 1) and stacking is enabled (i.e., L > 1), the minimum cost is lower compared to when there is no stacking (i.e., L = 1), which is due to the change in the number of transit nodes because only the transit node cost of the upper tunnel is present. This formulation finds a minimum when the push cost is between 50% above or below the average value of the other costs ($0.5 \leq C_{push} < 1.5$). In consequence, taking into account that push cost is only present in the stacking process, in this model it is desirable that the number of tunneling levels be 2 or 3, to reduce the total cost (L = 3 is a typical number of tunnel levels in current networks as is mentioned in [86]). This conclusion is valid for both technologies (MPLS and PSA-TE6).

5 Conclusions and future works

This article presented a new proposal to support traffic engineering based on the use of the IPv6 flow label field for packet switching, which we have called PSA-TE6. First, we presented a summary of the works related to the IPv6 flow label and traffic engineering taxonomy; then, we compared our proposal with others that use a similar concept. Finally, we presented three evaluations of our proposal where an analysis of the label space reduction and label stacking cost was performed. We also presented an ILP formulation in which the effects of the ingress and egress costs were analyzed. In these evaluations, we compared the results with MPLS since our proposal uses the same concept of label switching.

The results of the evaluation regarding the processing costs showed that PSA-TE6 has lower costs by 45% with respect to IP/MPLS when both (PSA-TE6 and IP/MPLS) are operating in their base forms, i.e., without label stacking. For the two proposed label stacking mechanisms, PSA-TE6 outperforms IP/MPLS when the stacking is enabled until a 40% presence of tunnels with levels higher than 1 with respect to the number of established communications, while the PSA-TE6 proposal maintains good performance in the label switching cost.

With respect to ILP formulation, the effect of the ingress, egress and push costs could be seen in the minimum total cost. It can be inferred from the results in the minimum cost that it is more harmful to have higher ingress and egress costs as these costs are present when there is no stacking and for all stacking levels. On the other hand, the higher push costs have a lower effect because the push cost is only present in the case of stacking, and thus the stacking percentage can be minimized in some cases where strictly necessary. Here, it is important to mention that, in the cost analysis by reading/writing, it allows deducing that the PSA-TE6 performance presents high benefits with regard to IP/MPLS since PSA-TE6 has lower ingress, egress, and transit costs compared to IP/MPLS.

It is important to highlight that all evaluations presented are essential to determine the performance of routers during the routing of the packets. Another highlight is that the PSA-TE6 solution satisfies the recommendations of the IETF for using the IPv6 flow label for switching control. PSA-TE6 establishes that the IPv6 flow label can be used for packet switching in combination with routing and signaling protocols extended to support traffic engineering. Additionally, it could define paths using constraint-based routing algorithms without the need for IP/MPLS. However, this situation has not been proved yet, and it will be part of future studies. Therefore, future works may be related to the creation and analysis of constraint-based routing algorithms to operate properly with this new proposal.

Also, as future work, development of testbeds will be implemented by using PSA-TE6 architecture. It could be achieved utilizing programmable routers or open source frameworks such as Netfpga. Other future work may aim at studying load balancing and multipath routing for the provision of traffic engineering. Another study could be focused on to evaluate how our proposal affects QoS parameters such as packet delay, bandwidth, and loss of packets by means statistical and optimization models. Also, could be an interesting issue to evaluate PSA-TE6 behavior under a burst of packets of different sizes from the source to the destination. Also, another scenario could be focused on the study the behavior of PSA-TE6 over networks with a big number of routers. Finally, future works could be focused on the evaluation of our proposal in combination with other strategies, such as segment routing or LISP.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Postel, J. (1981). *Internet protocol*. IETF RFC791.
2. Deering, S., & Hinden, R. (1998). *Internet protocol, version 6 (IPv6) specification*. IETF RFC2460.
3. Deering, S., & Hinden, R. (2017). *Internet protocol, version 6 (IPv6) specification*. IETF RFC8200.
4. Dhamdhere, A., Luckie, M., & Huffaker, B. (2012). Measuring the deployment of IPv6: Topology, routing, and performance. In *Proceedings of the ACM internet measurement conference* (pp. 537–550).
5. Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., & Bailey, M. (2014). Measuring IPv6 adoption. In *Proceedings of 2014 ACM conference on SIGCOMM* (Vol. 44, no. 4, pp. 87–98).
6. Nikkhah, M., & Guérin, R. (2016). Migrating the internet to IPv6: An exploration of the when and why. *IEEE/ACM Transactions on Networking, 24*(4), 2291–2304.
7. Hu, Q., & Carpenter, B. (2011). *Survey of proposed use cases for the IPv6 flow label*. IETF RFC6294.
8. Becerra, L. Y., & Padilla, J. J. (2014). Review of approaches for the use of the flow label of IPv6 header. *IEEE Transactions Latin America, 12*(8), 1602–1607.
9. Rosen, E., Viswanathan, A., & Callon, R. (2001). *Multiprotocol label switching architecture*. IETF RFC3031.
10. Hinden, R. (1994). *Simple internet protocol plus white paper*. IETF RFC1710.
11. Deering, S., & Hinden, R. (1995). *Internet protocol, version 6 (IPv6) specification*. IETF RFC1883.
12. Metzler, J., & Hauth, S. (2000). *An end-to-end usage of the IPv6 flow label.* Work Prog.
13. Conta, A., & Carpenter, B. (2001). *A proposal for the IPv6 flow label specification*. IETF internet-draft.
14. Conta, A., & Rajahalme, J. (2001). *A model for Diffserv use of the IPv6 flow label specification*. Work Prog.
15. Hagino, J. (2001). *Socket API for IPv6 flow label field*. IETF internet-draft.
16. Rajahalme, J., Conta, A., Carpenter, B., & Deering, S. (2004). *IPv6 flow label specification*. IETF RFC3697.
17. Amante, S., Carpenter, B., & Jiang, S. (2011). *Rationale for update to the IPv6 flow label specification*. IETF RFC6436.
18. Amante, S., Carpenter, B., Jiang, S., & Rajahalme, J. (2011). *Ipv6 flow label specification*. IETF RFC6437.
19. Chakravorty, S. (2008). Challenges of IPv6 flow label implementation. In *Proceedings of IEEE MILCOM2008.*
20. Chakravorty, S., Bush, J., & Bound, J. (2008). *IPv6 label switching architecture*. Work Prog.
21. Beckman, M. (2007). *IPv6 dynamic flow label switching (FLS).* IETF internet-draft.
22. Nichols, K., Blake, B., Baker, F., & Black, D. (1998). *Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers*. IETF RFC2474.
23. Roberts, L.,& Harford, J. (2005). *In-band QoS signaling for IPv6.* Work Prog.

24. Chin-Ling, C. (2011). A study of IPv6 labeling forwarding model supporting Diffserv. *Procedia Engineer Science Elsevier, 15,* 5590–5594.
25. Banerjee, R., Malhotra, S. P., & Mahaveer, M. (2002). *A modified specification for use of the IPv6 flow label for providing an efficient quality of service using a hybrid approach*. Work Prog.
26. Lin, C., Tseng, P., & Hwang, W. (2006). End-to-end QoS provisioning by flow label in IPv6. In *ICIS*.
27. Lee, I., & Kim, S. (2004). A QoS improvement scheme for real-time traffic using IPv6 flow labels. In *Lecture notes* in *computer science* (Vol. 3043).
28. Prakash, B. (2004). *Using the 20 bit flow label field in the IPv6 header to indicate desirable quality of service on the internet*. University of Colorado (MSc Thesis).
29. Aazam, M., Syed, A. M., & Huh, E.-N. (2013). Redefining flow label in IPv6 and MPLS headers for end to end QoS in virtual networking for thin client no title. In *19th Asia-Pacific conference on communications. (APCC)*, *Bali—Indones* (pp. 585–590). https://doi.org/10.1109/apcc.2013.6766016.
30. Hassan, R., & Jabbar, R. (2017). End-to-end (e2e) quality of service (QoS) for IPv6 video streaming. In *IEEE 19th international conference on advanced communication technology (ICACT).*
31. Glennan, T., Leckie, C., & Erfani, S. M. (2016). Improved classification of known and unknown network traffic flows using semi-supervised machine learning. In *Australasian conference on information security and privacy* (pp. 493–501). Springer.
32. Yin, A., & Zhang, S. (2015). Design and implementation of trusted routing strategy based on IPv6 flow identification. In *10th international conference on communications and networking in China (ChinaCom)* (pp. 887–892).
33. Padilla, J., & Paradells, J. (2007). Intserv6: An approach to support QoS over IPv6 wired and wireless networks. *European Transactions on Telecommunications, 19*(6), 635–652.
34. Padilla, J. J., Paradells, J., & Rodriguez, A. (2006). Supporting QoS over IPv6 wireless networks with IntServ6. In *IEEE 17th international symposium on personal*, *indoor and mobile radio communications* (pp. 1–6).
35. Doan, H., et al. (2006). Flow-based forwarding scheme and performance analysis in mobile IPv6 networks. In *International conference of advanced communications technology* (Vol. 3, pp. 1490–1496).
36. Zheng, T., Wang, L., & Gu, D. (2012). A flow label based QoS scheme for end-to-end mobile services. In *ICNS 2012-Ehte eighth international conference on networking and services* (pp. 169–174).
37. Ouellette, S., & Pierre, S. (2006). HPMRSVP-TE: A hierarchical proxy mobile resource reservation protocol for traffic engineering. In *IEEE 64th vehicular technology conference 2006* (pp. 1–5).
38. Yee,T. W., Eng, T. Ch., & Ping, L. S. (2012). Towards utilizing flow label IPv6 in implicit source routing for dynamic source routing (DSR) in wireless ad hoc network. In *IEEE Symposium* on *computers* and *informatics* (pp. 101–106).
39. Donley, C., & Erichsen, K. (2011). *Using the flow label with dual-stack lite*. Work Prog.
40. Carpenter, B., & Amante, S. (2011). *Using the IPv6 flow label for equal cost multipath routing and link aggregation in tunnels*. IETF RFC6438.
41. Hartmond, F., Rouhi, M., &Scholz, D. (2017). Detecting load balancers in the Internet. In *Proceedings of the seminars future internet (FI) and innovative internet technologies and mobile communication (IITM) focal topic: Advanced persistent threats.* Munich, Germany (pp. 17–23).
42. Braden, R., Clark, D., & Shenke, S. (1994). *Integrated services in the internet architecture: An overview*. IETF RFC1633.
43. Melnikov, D. A., Lavrukhin, Y. N., Durakovsky, A. P., Gorbatov, V. S., & Petrov, V. R. (2015). Access control mechanism based on entity authentication with IPv6 header 'flow label' field. In *3rd international conference on future internet of things and cloud* (pp. 158–164).
44. Blake, S. (2009). *Use of the IPv6 flow label as a transport-layer nonce to defend against off-path spoofing attacks*. Work Prog.
45. Hendriks, L., Velan, P., Schmidt, R. de O., de Boer, P. T., & Pras, A. (2017). Flow-based detection of ipv6-specific network layer attacks. In *IFIP international conference on autonomous infrastructure*, *management and security. AIMS 2017: Security of networks and services in an all-connected world* (pp. 137–142).
46. Bobade, S., & Goudar, R. (2015). Secure data communication using protocol steganography in IPv6. In *International conference on computing communication control and automation* (pp. 275–279).
47. Awduche, D., et al. (2002). *Overview and principles of internet traffic engineering*. IETF RFC3272.
48. Becerra, L. Y., & Padilla, J. J. (2012). Study of proposals for supporting internet traffic engineering. *Entre Ciencia e Ingeniería, 6*(11), 53–76.
49. Fortz, B., & Thorup, M. (2000). Internet traffic engineering by optimizing OSPF weights. In *Proceedings of IEEE INFOCOM* (pp. 519–528).
50. Fortz, B., Rexford, J., & Thorup, M. (2002). Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine, 40*(10), 118–124.
51. Thorup, M., & Fortz, B. (2002). Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications, 20*(4), 756–767.
52. Ericsson, M., Resende, M., & Pardalos, P. (2002). A genetic algorithm for the weight setting problem in OSPF routing. *Journal of Combinatorial Optimization, 6*(3), 299–333.
53. Gojmerac, I., Ziegler, T., Ricciato, F., & Reichl, P. (2003). Adaptive multipath routing for dynamic traffic engineering. In *Proceedings of IEEE GLOBECOM* (pp. 3058–3062).
54. Wang, J., et al. (2005). Edge based traffic engineering for OSPF networks. *Computer Networks, 48*(4), 605–625.
55. Abrahamsson, H., & Bjorkman, M. (2009). Robust traffic engineering using L-balanced weight-settings in OSPF/IS-IS. In *Broadband communications*, *networks*, *and systems BROADNETS. Sixth international conference* (pp. 1–8).
56. Xu, K., Liu, H., Liu, J., & Shen, M. (2011). One more wight is enough: Toward the optimal traffic engineering with OSPF. In *IEEE Computer Society 31st international conference on distributed computing systems* (pp. 836–846).
57. Xu, K., Shen, M., Liu, H., Liu, J., Li, F., & Li, T. (2016). Achieving optimal traffic engineering using a generalized routing framework. *IEEE Transactions on Parallel and Distributed Systems, 27*(1), 51–65.
58. Awduch, D., et al. (1999). *Requirements for traffic engineering over MPLS*. IETF RFC2702.
59. Awduche, D. (1999). MPLS and traffic engineering in IP networks. *IEEE Communications Magazine, 37*(12), 42–47.
60. Karaman, A. (2006). Constraint-based routing in traffic engineering. In *Computer networks* (pp. 49–54).
61. Nayak, P., & Murty, G. R. (2013). Survey on constrained based path selection QoS routing algortihms: MCP and MCOP Problems. *Journal of Information Systems and Communication, 4*(1), 6.
62. Farinacci, D., Fuller, V., Meyer, D., & Lewis, D. (2013). *The locator/ID separation protocol (LISP)*. IETF RFC6830.
63. Farinacci, D., Kowal, M., & Lahiri, P. (2016). *LISP traffic engineering use-cases*. draft-farinacci-lisp-te-11.
64. Saucez, D., Donnet, B., Iannone, L., & Bonaventure, O. (2008). Interdomain traffic engineering in a locator/identifier separation Context. In *IEEE internet network management workshop.*
65. Li, K., Wang, S., & Wang, X. (2011). Edge router selection and traffic engineering in LISP-capable networks. *IEEE Journal of Communications and Networks, 13*(6), 612–620.

66. Li, K., Wang, S., Xu, S., & Wang. (2011). ERMAO: An enhanced intradomain traffic engineering approach in LISP-capable Networks. In *IEEE Global Telecommunications Conference—GLOBECOM*.
67. Herrmann, D., Turba, M., Kuijper, A., & Schweizer, I. (2014). Inbound interdomain traffic engineering with LISP. In *39th annual IEEE conference on local computer networks* (pp. 458–461).
68. Jeong, T., Liy, J., Hyun, J., Yoo, J.-H., & Hong, J. W.-K. (2015). Experience on the development of LISP-enabled services: An ISP perspective. In *Proceedings of the 2015 1st IEEE conference on network softwarization* (pp. 1–9).
69. Nguyen, H. D. D., & Secci, S. (2016). LISP-EC: Enhancing LISP with egress control. In *IEEE conference on standards for communications and networking*.
70. Filsfils, C., Previdi, S., Decraene, B., Litkowski, S. & Shakir, R. (2017). *Segment routing architecture*. draft-ietf-spring-segment-routing-11.
71. Filsfils, C., et al. (2017). *Segment routing policy for traffic engineering*. raft-filsfils-spring-segment-routing-policy-00.
72. Bhatia, R., Hao, F., Kodialam, M., & Lakshman, T. V. (2015). Optimized network traffic engineering using segment routing. In *IEEE conference on computer communications*.
73. Rabah, G., Olivier, D., Samer, L., & Texier, G. (2016). Label encoding algorithm for MPLS segment routing. In *IEEE international symposium on network computing and application* (pp. 113–117).
74. Salsano, S., Siracusano, G., Luca, V., Luca, D. and Pier, L. (2016). PSMR-poor man's segment routing, a minimalistic approach to segment routing and a traffic engineering use case. In *Network operations and management Symposium* (pp. 598–604).
75. Moreno, E., Beghelli, A., & Cugini, F. (2017). Traffic engineering in segment routing networks. *Computer Networks, 114,* 23–31.
76. Balbinot, L., de Andrade, M., Tarouco, L., & Roesler, V. (2002). IP next generation label switching. In *IEEE Workshop on IP operations and management* (pp. 21–25).
77. Ishiguro, K., Manral, V., Davey, A., & Lindem, A. (2008). *Traffic engineering extensions to OSPF version 3*. IETF RFC5329.
78. Awduche, D. et al., (2001). *RSVP-TE: Extensions to RSVP for LSP tunnels*. IETF RFC3209.
79. Younis, O., & Fahmy, S. (2003). Constraint-based routing in the internet: Basic principles and recent research. *IEEE Communications Surveys & Tutorials*, *5*(1), 2–13.
80. Becerra, L. Y., Padilla, J. J., & Bañol, J. L. (2017). A survey on constraints-based routing algorithms: objectives traffic engineering and quality of service. *Revista Entre Ciencia e Ingeniería, 21,* 112–122.
81. Medhi, D., & Ramasay, K. (2007). *Network routing-algorithms, protocols and architectures* (pp. 166–191). San Francisco: Morgan Kaufmann.
82. Davie, B. S., & Farrel, A. (2008). *MPLS: Next steps*. San Francisco: Morgan Kaufmann Publishers.
83. Conta, A., & Deering, S. (1998). *Generic packet tunneling in IPv6 specification*. IETF RFC2473.
84. Mathworks. (1994). Retrieved June 8, 2016 from https://www.mathworks.com/products/matlab.html.
85. Cerrutti, I., & Castoldi, P. (2006). Influence of label stack depth on the performance of MPLS networks. In *IEEE Globecom* (pp. 1–5).
86. Vanaubel, Y., Mérindol, P., Pansiot, J.-J., & Donnet, B. (2016). A brief history of MPLS usage in IPv6. In *International conference on passive and active measurement link* (pp. 359–370).
87. AMPL Optimization Inc. (2013). AMPL. Retrieved May 15, 2016 from http://ampl.com.
88. CPLEX for AMPL. (2013). Retrieved May 20, 2016 from https://ampl.com/products/solvers/solvers-we-sell/cplex/.

Line Y. Becerra Sánchez is an electronic engineer and M.Sc in engineering in the telecommunications area, both from Universidad Pontificia Bolivariana. She is currently a Ph.D. student at Universidad Pontificia Bolivariana. She is an associate professor at the Universidad Católica de Pereira. Her recent research interests include Traffic Engineering, Routing, Internet, IPv6, MIPv6, and HMIPv6.

Jhon J. Padilla Aguilar is an electronic engineer from the Universidad de Cauca. He received his Master's degree in Computer Science from the Universidad Industrial de Santander in 1998, and Ph.D. in Telematics Engineering from the Polytechnic of Catalonia University in 2008. He currently teaches at the Faculty of Electronic Engineering at the Universidad Pontificia Bolivariana. His areas of interest are Traffic Engineering, Internet, Quality of Service on the Internet, wireless networks, IPv6.