# Full Label Space Reduction in MPLS Networks: Asymmetric Merged Tunneling

Fernando Solano, *Student Member, IEEE*, Ramon Fabregat, Jose Luis Marzo, *Member, IEEE*

***Abstract*— Traffic Engineering objective is to optimize network resource utilization. Although several works have been published about minimizing network resource utilization in MPLS networks, few of them have been focused in LSR label space reduction. This letter studies Asymmetric Merged Tunneling (AMT) as a new method for reducing the label space in MPLS network. The proposed method may be regarded as a combination of *label merging* (proposed in the MPLS architecture) and *asymmetric tunneling* (proposed recently in our previous works). Finally, simulation results are performed by comparing AMT with both ancestors. They show a great improvement in the label space reduction factor.**

***Index Terms*— Label space reduction, MP2P, label merging, asymmetric tunnels, label stack, NHLFE, MPLS traffic engineering.**

## I. INTRODUCTION

Traffic Engineering (TE) is concerned in improving performance of operational networks, usually considering Quality of Service (QoS) requirements. Multi Protocol Label Switching (MPLS) aims to work with these TE schemes by setting up Label Switched Paths (LSPs) as needed to transmit efficiently customer's flows with their QoS requirements. Customer QoS requirements, i.e. delay, packet loss, jitter, etc, are flow dependent. Although this can be achieved in many ways using different algorithms, Internet Service Providers (ISP) must be aware of Label Switched Router (LSR) resources utilization such as the label space.

Once a LSP is established, all the involved LSRs should use a label in order to identify the LSP. In other words, every packet of a LSP must be marked with a label that uniquely identifies the LSP in the LSR. When a packet is received by a LSR, the LSR looks for the packet label and then search for a Next Hop Label Forwarding Entry (NHLFE) in its memory that refers to this label. The NHLFE gives information about which interface will be used to reach the next hop in the network [1]. Clearly, the more LSPs a LSR support, the more NHLFEs are needed.

Among many reasons and motivations to reduce the label space we outline the following: a) large, but finite label space [1] for label encoding (specially considering MPLS multicast [1] , fast protection [2] or VPN support [3] [4]); b) increased look up delay [3], [5] in MPLS forwarding; c) support for MP$\lambda$S (low multiplexing capabilities); d) LSP protection response time improvement [6].

One of the basis of the method presented here to reduce the label space is based on the use of the label stack, which was originally stated for LSP tunneling across domains. LSP tunneling in MPLS networks is a feature that allows a set of LSP to be *joint* into a single one. To support LSP tunneling and forwarding, IETF defined a label stack for MPLS packets [7] and a set of feasible operations over this stack [1]: a) replace the label at top for a new one (swap), b) pop the stack (pop), c) replace the label at top for a new one and then push one or more onto stack (push). Each NHLFE associates an incoming label with one of these operations (which will be done over packets stack with the incoming label) an outgoing forwarding port. LSRs first decide where to forward the packets and then perform the operation stored in its NHFLE.

Taking advantage of the different possible operations a NHLFE may have, the number of labels used (label space) could be increased or reduced depending on how NHLFEs are configured, as explained in further sections. Since one label may store forwarding information for more than one LSP, there is a label space reduction. Therefore, the general Label Space Reduction problem can be stated as:

> *how can the NHLFEs be set up for a set of LSRs in a network so that the total number of labels used in the network is minimized?*

The acronym LASPARED is used along the letter to denote the LAbel SPAce REDuction problem.

So far, all published works dealing with the LASPARED problem have their basis on the *label merging* scheme (swap operations). On the other hand, our previous and innovative works ([8] and [9]) have contemplated the label stack (push and pop operations) as a way to solve the LASPARED problem under the new concept of *asymmetric tunneling*. These two existing schemes are summarized in section II. In section III a new scheme to solve the LASPARED problem is presented. The results of some experiments done in random generated flows and networks are analyzed in section IV. Finally, in section V, conclusions and new directions in the LASPARED problem are discussed.

The authors would like to remark that all presented reduction schemes are fully compliant with current MPLS architecture, i.e. no changes are needed in the current standard.

## II. FOUNDATIONS

In MPLS networks, labels are upstream assigned. Hence, LASPARED methods aim to minimize the number of incom-

---

Manuscript received March 29, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Iakovos Venieris. The work of Fernando Solano is partially supported by the Government of Catalonia under contract 2005FIR-00379. This work is also partially supported by both the Spanish Science and Technology Ministry under contract TIC2003-05567 and the European COST project action number 293 (GRAAL).

The authors are with the Institute of Informatics and Applications, University of Girona, Spain (e-mail: fsolanod@eia.udg.es).

Digital Object Identifier 10.1109/LCOMM.2005.11016.

ing labels per LSR[1] and, therefore, the number of NHLFEs in LSRs forwarding tables [3], [5], [8], [9], [10].

So far, there are only two ways for dropping off the label space: a) *label merging*, and, b) *asymmetric tunneling* based on MPLS label stack.

For simplicity of notation, a LSP path $\alpha_0 \rightarrow \ldots \rightarrow \alpha_A$ is going to be rewritten as $\alpha_{0\ldots A}$.

### A. Label Merging: MultiPoint-to-Point Trees

The MPLS architecture allows labels merging in LSPs by assigning to many LSPs, which have a common egress LSR, the same label. In other words, let LSP A and B be two paths that share their final segment:

- $A : \alpha_{0\ldots A} \rightarrow \lambda_{0\ldots N}$, and
- $B : \beta_{0\ldots B} \rightarrow \lambda_{0\ldots N}$,

Then, this scheme allows shared LSRs $\lambda_{0\ldots N}$ to mark packets for A and B with the same label only if $\lambda_N$ is the egress node for both LSPs. If a set of LSPs having a common egress node are merged, then this reduction scheme builds an inverse tree rooted at an egress LSR with leaves at the ingress LSRs of the set of LSPs. MPLS architecture names such structure as MultiPoint-to-Point (MP2P) trees. MP2P lacks of scalability when LSPs do not share links attached to the same egress node.

Given a set of pre-computed LSPs routes, there are many ways of constructing MP2P trees. However, some of them lack in allowing LSPs rerouting for label space reduction, i.e. to compute a new path with the goal of best label merging (e.g. [4]), which can leads to QoS degeneration. Others try to compute a minimal set of MP2P trees (e.g. [5], [10]) but this will not always reduce as most the number of labels used in the network.

### B. Label Stacking: Asymmetric Tunneling

The asymmetric tunneling method (AT) ([8] and [9]) refers to pushing the same label in a set of LSPs, so LSRs can regard them as belonging to the same 'LSP'.

Consider two LSPs A and B that shares a segment placed elsewhere

- $A : \alpha_{0\ldots A} \rightarrow \lambda_{0\ldots N} \rightarrow \alpha'_{0\ldots A'}$, and
- $B : \beta_{0\ldots B} \rightarrow \lambda_{0\ldots N} \rightarrow \beta'_{0\ldots B'}$,

Then, this scheme allows shared LSRs $\{\lambda_0, \lambda_1, \ldots, \lambda_N\}$ to use the same label by pushing at $\lambda_0$ the same, but new, label into both LSPs label stack; and then do a pop of the stack at $\lambda_N$. Therefore, LSRs $\lambda_{1\ldots N}$ will regard only one label for both LSPs. In this way, it is said that LSP A and B are *stacked* or *tunneled* in $\lambda_{0\ldots N}$.

The method is scalable in the sense that if another LSP, $C : \gamma_{0\ldots C} \rightarrow \lambda_{i\ldots N} \rightarrow \gamma'_{0\ldots C'}$, shares the final sequence of LSRs of a tunnel, $\lambda_{i\ldots N}$, then LSR $\lambda_i$ can perform a push into C label stack in order to forward C packets through the tunnel. In this case, it is said that C is *partially tunneled*. MPLS architecture states that LSRs may only regard the label at the top of the stack, hence all LSP must be *unstacked* at the same time and therefore, asymmetric tunnels will be usually "bigger" at the end than at the beginning. Fig. 1 shows an example of an asymmetric tunnel for 3 LSPs (one of them partially tunneled) in which the label space has been dropped of from 13 to 9 (reduction factor around 30.7%).

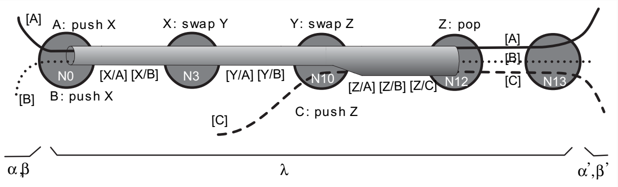

Fig. 1. Asymmetric Tunnel built for 3 LSPs in 4 LSRs. Packets are received at $N0$ with labels A and B. $N0$ push label $X$ onto both packet stacks. $N3$ performs swapping of label $X$ to $Y$. $N10$ swaps labels $Y$ to $Z$ and, in addition, push label $Z$ for packets marked with label $C$. $N12$ receives packets marked with label $Z$, computes the next hop, pop the stack, and forward them all to $N13$.

This reduction scheme is presented as a 2-depth stacking scheme, in other words, only one more label is used in label stacks to reduce the label space.

Due ATs can be built in many ways, some heuristics were proposed in previous works [8] and [9].

## III. ASYMMETRIC MERGED TUNNELING

So far, neither asymmetric tunneling gets advantage of labels merging feature, nor labels merging of asymmetric tunneling. In this section, a mixed version of both methods that preserves their advantages is proposed.

Our proposal may be seen as a merging of asymmetric tunnels into a single MP2P connection, therefore decreasing even more the number of labels used in the network. Also, it may be seen as a way to create MP2P trees where the root LSR may not be the egress LSR of a set of LSPs, i.e. MP2P trees anywhere in the network.

Formally, given 4 LSPs (e.g. see Fig. 2):

- $A : \alpha_{0\ldots A} \rightarrow \mu_{0\ldots M} \rightarrow \lambda_{0\ldots L} \rightarrow \alpha'_{0\ldots A'}$,
- $B : \beta_{0\ldots B} \rightarrow \mu_{0\ldots M} \rightarrow \lambda_{0\ldots L} \rightarrow \beta'_{0\ldots B'}$,
- $C : \gamma_{0\ldots C} \rightarrow \nu_{0\ldots N} \rightarrow \lambda_{0\ldots L} \rightarrow \gamma'_{0\ldots C'}$, and
- $D : \delta_{0\ldots D} \rightarrow \nu_{0\ldots N} \rightarrow \lambda_{0\ldots L} \rightarrow \delta'_{0\ldots D'}$

Then, an Asymmetric Merged Tunnel (AMT) tree can be built when:

1) $\mu_0$ pushes a new label into LSPs A and B packet stacks,
2) $\nu_0$ pushes another new label into LSPs C and D stacks,
3) since labels are upstream assigned, $\lambda_0$ could ask both $\mu_M$ and $\nu_N$ to forward packets with the same label. Then, $\lambda_0$ may regard both flows as the same (using one NHLFE).
4) finally, $\lambda_{L-1}$ does a pop of packets stack, so $\lambda_L$ may receive packets with the original label (given by $\alpha_A$, $\beta_B$, $\gamma_C$ and $\delta_D$) and, hence, forward them to its correct destination (i.e. $\alpha'_0$, $\beta'_0$, $\gamma'_0$ or $\delta'_0$).

In Fig. 2 an example, in which the label space is dropped off from 16 to 11, for 4 LSPs is presented.

Note that all LSRs (except $\lambda_L$) have only one NHLFE to forward the 4 LSPs. The results can be even more dramatic if it is taken into account that the number of NHLFE (always one) is not affected by the number of LSPs inside an AMT

---

[1] Or the total number of incoming labels since the number of LSRs is constant.

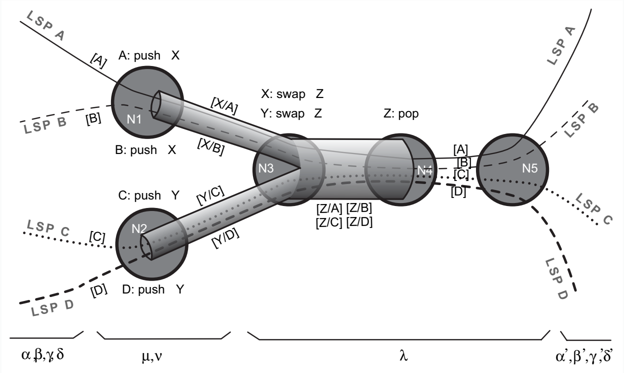

Fig. 2. AMT tree for 4 LSPs. LSR $N1$ pushes label $X$ for packet marked with labels $A$ and $B$ (belonging to LSPs A and B respectively). In the same way, LSR $N2$ pushes label $Y$ to packets marked with labels $C$ and $D$. Node $N3$ merges them by swapping their labels to the same label $Z$. LSR $N4$, then, receives pop the stack for packets marked with label $Z$.

tree. Moreover, this scheme also uses only one pushed label to reduce the label space, therefore nested tunnels are avoided.

Comparing it with its ancestors, it should be remarked that *label merging* can not perform such type of reduction since it demands the same egress node for all LSPs. AT can be used here, but it can not achieve the same reduction factor. In the example of Fig. 2, the best AT feasible solution will set up 3 NHLFEs for $\lambda_0$: two to do partial tunnels for LSPs C and D respectively, and another to tunnel LSPs A and B.

To explore heuristics to compute AMT trees given a set of LSPs is out of the scope of the letter.

## IV. PRELIMINARY SIMULATION RESULTS

AMT trees are a general case for both MP2P trees and ATs; therefore AMT trees may achieve in the worst case the same reduction factor as its ancestors. This is supported by some experimental results regarded in this section.

ATs were left out of our simulation experiments because it requires the same network effort (one pushed label) and reaches a lower label space reduction comparing it with AMTs (as discussed in previous section).

Since the number of labels increases as the number of LSPs increases, we evaluate the reduction factor respect to the number of LSPs placed in a network. The network used was a 25-nodes with 75x2-links network randomly generated following Siganos et al. ”Power Laws” [11].

The network is loaded by querying a new LSP request; all having the same bandwidth, but differing in the ingress and egress LSRs. Then, a routing algorithm considering several ISPs considerations is used in order to determine LSPs routes. Finally, the best reduction achieved by using MP2P and AMTs is computed for the given set of paths.

Fig. 3 illustrates the reduction factor when the network is loaded until 128 LSPs. An approximation logarithmic curve shows that AMT reduction factor may converge up to a 55%. In the other hand, it shows that a poor reduction factor around 15% is achieved when MP2P trees method is used as well. Also, it should be stated that AMT trees methods converge faster to an appropriate reduction factor than MP2P trees.

Our tests were done up to 128 LSPs only. It was this way because they wanted to outline the *difference* between the MP2P and AMTs reductions (i.e. how much can it be improved). It is clear that the experiments done with thousands, or millions, of LSPs will reduce even more the label space (e.g. [3]).

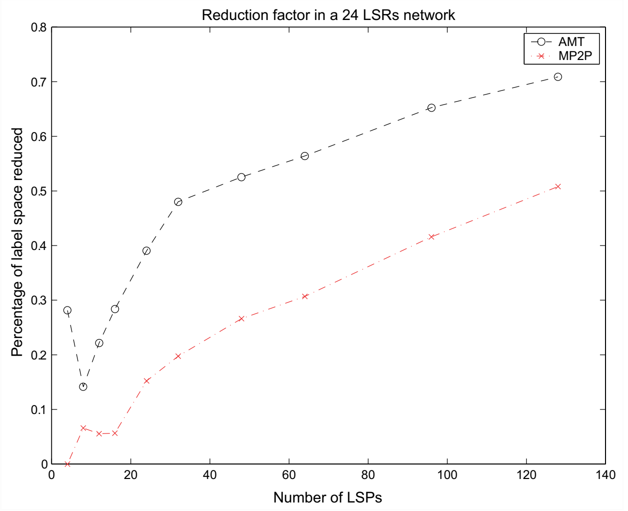

Fig. 3. Reduction factor experienced vs. number of placed LSPs in a network. MP2P: label merging - AT: asymmetric tunneling - AMT: asymmetric merged tunneling.

## V. CONCLUSIONS AND NEW DIRECTIONS

The presented letter proposed a new scheme for the LASPARED problem: Asymmetric Merged Tunneling (AMT). Preliminary results show great improvements respect to its predecessors: AT (previous works) and, specially, respect to the traditional and well-known MP2P trees.

Heuristics to build AMT trees, routing algorithms that aim to find LSPs routes taking into account AMT trees reduction, and signalling protocols to create AMT trees are left as new directions in the LASPARED problem area.

## REFERENCES

[1] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” RFC 3031, Jan. 2001.

[2] J. L. Marzo, E. Calle, C. Scoglio, and T. Anjali, “QoS on-line routing and MPLS multilevel protection: a survey,” *IEEE Commun. Mag.*, vol. 41, pp. 126-132, Oct. 2003.

[3] A. Gupta, A. Kumar, and R. Rastogi, “Exploring the trade-off between label size and stack depth in MPLS Routing,” in *Proc. IEEE INFOCOM 2003*, pp. 544-554.

[4] D. Applegate and M. Thorup, “Load optimal MPLS routing with N+M labels”. in *Proc. IEEE INFOCOM 2003*, pp. 555-565.

[5] S. Bhatnagar, S. Ganguly, and B. Nath, “Creating multipoint-to-point LSPs for traffic engineering,” *IEEE Commun. Mag.*, vol. 43, pp. 95-100, Jan. 2005.

[6] C. Neophytou and C. Phillips, “A scheme for the dynamic formation of robust multipoint to point LSPs,” in *Proc. IEEE CCNC 2004*, pp. 251-255.

[7] E. Rosen *et al.*, “MPLS Label Stack Encoding,” RFC 3032, Jan. 2001.

[8] F. Solano, R. Fabregat, Y. Donoso, and J. L. Marzo, “Asymmetric tunnels in P2MP LSPs as a label space reduction method,” in *Proc. IEEE ICC 2005*, pp. 43-47.

[9] F. Solano, R. Fabregat, Y. Donoso, and J. L. Marzo, “A label space reduction method for P2MP LSPs using asymmetric tunnels,” in *Proc. IEEE ISCC 2005*, pp. 746-751.

[10] H. Saito, Y. Miyao, and M. Yoshida, “Traffic engineering using multiple multipoint-to-point LSPs,” in *Proc. IEEE INFOCOM 2000*, pp. 894-901.

[11] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. ”Power-laws and the AS-level Internet topology,” *IEEE/ACM Trans. Networking*, vol. 11, pp. 514-524, Aug. 2003.