On Optimal Computation of MPLS Label Binding for MultiPoint-to-Point Connections

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

***Abstract*—Most network operators have considered reducing Label Switched Routers (LSR) label spaces (i.e. the number of labels that can be used) as a means of simplifying management of underlaying Virtual Private Networks (VPNs) and, hence, reducing operational expenditure (OPEX). This letter discusses the problem of reducing the label spaces in MultiProtocol Label Switched (MPLS) networks using *label merging* - better known as MultiPoint-to-Point (MP2P) connections. Because of its origins in IP, MP2P connections have been considered to have tree-shapes with Label Switched Paths (LSP) as branches. Due to this fact, previous works by many authors affirm that the problem of minimizing the label space using MP2P in MPLS - the** MERGING PROBLEM **- cannot be solved optimally with a polynomial algorithm (NP-complete), since it involves a hard-decision problem. However, in this letter, the** MERGING PROBLEM **is analyzed, from the perspective of MPLS, and it is deduced that tree-shapes in MP2P connections are irrelevant. By overriding this tree-shape consideration, it is possible to perform label merging in polynomial time. Based on how MPLS signaling works, this letter proposes an algorithm to compute the minimum number of labels using label merging: the *Full Label Merging* algorithm. As conclusion, we reclassify the** MERGING PROBLEM **as Polynomial-solvable, instead of NP-complete. In addition, simulation experiments confirm that without the tree-branch selection problem, more labels can be reduced.**

***Index Terms*—Label space reduction, MP2P, label merging, NHLFE, MPLS, RSVP-TE.**

I. INTRODUCTION

MULTI PROTOCOL LABEL SWITCHING (MPLS) is a circuit-oriented technology developed for network Traffic Engineering (TE). MPLS aims to work with TE schemes by setting up Label Switched Paths (LSP) when necessary to transmit customer flows efficiently with their Quality of Service (QoS) requirements. Customer QoS requirements - e.g. delay, packet loss, jitter - are flow dependent. Many network operators (e.g. Bell [1], AT&T [2], NEC Japan [3], NEC USA [4]) aim to reduce the number of labels used in Label Switched Routers (LSR) not only because they are a finite resource, but because this simplifies network management, especially when Virtual Private Networks are considered. This could lead to reducing operation expenditures (OPEX). Other motivations include: a) reducing capital expenditures for the implementation of all-optical label swapping (see [5]) and b) the developing of the GMPLS controlled Ethernet Label Switching (GELS) project of the Internet Engineering Task Force (IETF).

When an LSP is created, all the LSRs involved must use a label to identify LSP packets going through. Initially, an LSR must employ one label for each of the forwarded LSPs. When a packet is received by an LSR, the LSR extracts the packet's label and then looks in the memory for a Next Hop Label Forwarding Entry (NHLFE) that refers to this label. NHLFEs give information about which interface must be used to reach the next hop in the network [6]. Clearly, the more LSPs an LSR supports, the more NHLFEs are needed.

Although MPLS packets may carry a stack of labels, which could be used for reducing label spaces even more [7], methods considering the stack are neither considered nor discussed in this letter. This letter focuses on the label merging method and in particular on the MERGING PROBLEM involved. Previous work classified the MERGING PROBLEM as NP-complete because of the *tree-shape consideration*. This consideration is analyzed in §II and §III, and in §IV it is shown to be unnecessary. Based on this, §V proposes a simple polynomial time (poly-time) algorithm, which achieves *at least* the same number of reduced labels as the optimal solution to the traditional scheme and is the main contribution of this letter. In §VI, preliminary simulations confirming this improvement are presented. Finally, the letter is concluded with a summary of its most noteworthy contributions in §VII.

II. MULTIPOINT-TO-POINT TREES: WHERE IS THE SO-CALLED MERGING PROBLEM?

As stated by the IETF working group, one of the advantages of MPLS label assignments is that it is possible to set the same outgoing label to many LSPs, when possible. In this way, it is said that a label is shared or that the respective LSPs are *merged* facing a single connection - a MultiPoint-to-Point(MP2P) LSP connection - using a single label per involved LSR.

Some authors' contributions - e.g. [2], [3] and [4] - consider that an MP2P connection looks like an *inverse tree* with leaves at the ingress LSRs and root in one egress LSR. This restriction is called the consideration of the *tree-shape* and it is the main point of discussion in this letter. Considering one MP2P connection, only one outgoing label is used by every LSR belonging to the tree, regardless of the number of LSPs it forwards. Hence, it is said that the MP2P connection reduces label spaces since LSRs could use less labels than forwarded LSPs.

Paper approved by G. S. Kuo, the Editor for Communication Architectures of the IEEE Communications Society. Manuscript received December 13, 2005; revised March 10, 2007. This work was supported by the Department of Universities, Research and Information Society (DURSI) of the Government of Catalonia and the European Social Funds under grants 2005-SGR-00296, 2006-BE-00272 and, 2006-FIR-00109, and with the support from the Spanish Science and Technology Ministry under project M2R3 - TEC2006-03883/TCM, and the European COST project action number 293: Graphs & Algorithms in Communication Networks.

The authors are with the Broadband and Distributed Systems Group (BCDS) of the Institute of Informatics and Applications (IIiA) at the University of Girona, Spain (e-mail: fs@tele.pw.edu.pl, {ramon, marzo}@atc.udg.edu).

Digital Object Identifier 10.1109/TCOMM.2008.050601.

0090-6778/08$25.00 © 2008 IEEE

The consideration imposes that each possible merged LSP is taken as a feasible branch of an inverse tree. As a result, the problem of reducing label spaces using label merging is transformed into selecting a subset of LSP routes such that they become a tree-shape structure aiming at the greatest label space reduction. Formally, it could be stated as follows.

MERGING PROBLEM

INSTANCE: Given a graph $G(V, E)$, a positive integer K, a node $v \in V$, and a set of directed paths $\mathcal{P}_v$ all starting[1] at node v,

QUESTION: Is there a set of directed *trees* rooted at v, $\mathcal{T}_v$, such that every directed path $p \in \mathcal{P}_v$ is contained by at least one tree $t \in \mathcal{T}_v$ and

$$\sum_{t \in \mathcal{T}_v} |t| \leq K,$$

where $|t|$ is the number of nodes that tree t uses?

The most remarkable problem of this tree-shape consideration is that every pair of selected LSPs for an MP2P connection - *viz.* every pair of branches of the tree - must not be intersected in more than one segment ending at the egress LSR. In other words, considering the branches (LSP routes) of an MP2P connection,

1) there is *at most* one common path between any pair of LSP routes and,
2) the common path - if it exists - *ends in the egress* LSR.

From now on, this common path is referred to as the *ending segment* for a selected pair of LSPs.

Although weak, the most important reason behind this consideration is that the created MP2P trees store precisely one outgoing label at each LSR for all forwarded LSPs. Even though this consideration makes LSP management easier for a network operator, it makes the problem NP-complete since considering one LSP route would rule out considering others LSP routes for the same tree that interfere with the first. Selecting an appropriate set of routes as branches for an MP2P tree will make the label spaces increase more or less depending on how many LSRs are shared among these routes. This MERGING PROBLEM was initially analyzed by Saito *et. al.* for MPLS networks in [3] and formally claimed to be NP-complete by Bhatnagar *et. al.* in §III of [8].

The configuration shown in Fig. 1 is an example[2] to illustrate the MERGING PROBLEM. LSPs B and C follow the same path and, in the same way, LSPs D and E follow a new different path. For this example, at least 2 MP2P trees must be created for the 5 LSPs shown, since the link $N11 \rightarrow N10$ is crossed by 4 diverging LSPs.

By merging LSP A with both LSPs D and E - leaving LSPs B and C on a new tree (Fig. 2(a)) - there are 14 labels that are saved[3], whereas by merging LSP A with both LSPs B and C - leaving LSP D and E on a new tree (Fig. 2(b) - there are 15 labels that are saved.

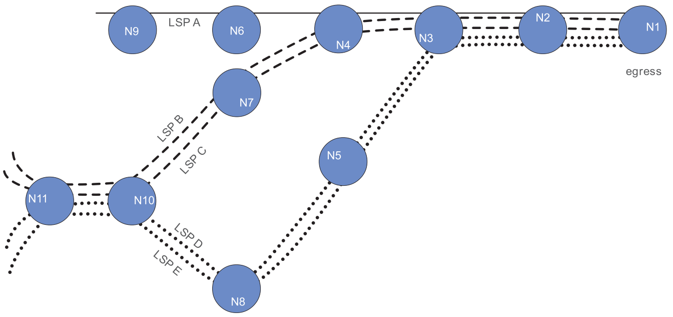
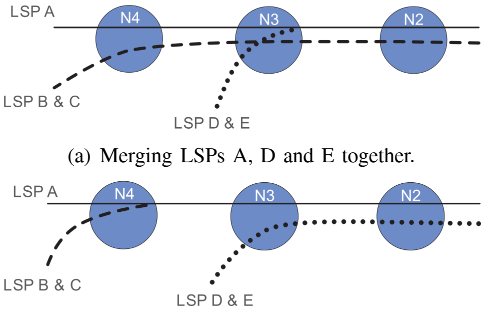


Fig. 1. MP2P scenario with 5 LSPs with the same egress LSR $N1$. There is more than one way to perform merging since $N11 \rightarrow N10$ is crossed by 4 LSPs (LSPs B, C, D and E).



(a) Merging LSPs A, D and E together.

(b) Merging LSPs A, B and C together

Fig. 2. Label merging solutions for the network scenario depicted in Fig. 1.

It should be pointed out that the longer the branches on an MP2P tree, the lower the number of non-interfering LSPs as branches of this tree.

Fact 1: The number of reduced labels is proportional to the number of merged LSPs in a tree.

III. MPLS, RSVP-TE AND MP2P SIGNALING

The Resource ReSerVation Protocol (RSVP) [9] was proposed for IP networks. The purpose of RSVP is to provide a signaling protocol capable of reserving resources over a path in order to provide QoS for IP flows.

Due to the dynamism of IP routing, RSVP was thought of as a soft-state protocol, i.e. each router stores RSVP information in "states" that need to be refreshed periodically[4]. In RSVP, IP flows are associated with sessions, and sessions are identified according to the tuple: destination address, destination port and protocol id of the routed packets. Please note that the sender address is not a part of the the identifying tuple.

In RSVP, resources can be reserved according to one of the following three styles: Wildcard-Filter (WF), Fixed-Filter (FF) and Shared Explicit (SE). With the FF style, each sender has its own reservation. The WF and SE styles allow resource sharing among all senders (with the WF style) or among an explicit list of senders (with the SE style). As a consequence, the WF and SE styles maintain one reservation (hence one soft-state) for several senders, while the FF style maintains one per sender. The WF and SE reservation styles are appropriate for those applications that are unlikely to

[1] In order to simplify the problem notation in this definition only, a path is taken as an inverted LSP route in the network.

[2] The various paths that cross at $N11 \rightarrow N10$ in the figure are suitable as an example, since they could be a solution for a TE scheme.

[3] Taking into account that LSR N1 does not receive labeled packets due to the IETF recommendation: *penultimate hop popping* in §3.16 of [6].

[4] Otherwise, routers would consider them as old, having the right to discard them

transmit data simultaneously to the same destination, since the reservation is shared among many senders. Voice-conference applications are examples of these applications.

At the time RSVP was designed, every node - based on its routing protocol - selected the next hop for an IP flow. This fact simplified the design of the WF and SE styles, since most routing protocols would route IP flows to the same destination (hence belonging to the same RSVP session) over the same path regardless of who the senders were. As a result of using IP routing, RSVP maintains an MP2P connection among many senders that had specified an WF or SE reservation style for the same session.

With the emergence of MPLS, RSVP was extended for MPLS under the name of RSVP for Traffic Engineering (RSVP-TE) [10]. One of the purposes of MPLS is to provide traffic engineering in the network. To that end, RSVP-TE was designed to include a new feature: the Explicit Routing Object (ERO). With the ERO feature, an MPLS ingress LSR is capable of determining (fixing) the LSP route end-to-end, therefore preventing intermediate LSRs in the network from altering it (as occurred with RSVP for IP flows). In addition, RSVP-TE extended the concept of a session to a more generic one allowing several different flows to be handled between any pair of source-destination addresses/ports.

These two features made RSVP-TE stop supporting the WF and SE reservation styles (see §2.4.2 and §2.4.3 of [10]) and, hence, of the MP2P connections in MPLS.

IV. Full Label Merging: Merging without Tree Structures

The IETF has been proposing several signaling modifications not only to provide MP2P connections for the WF and SE styles, but also to support MP2P connections for FF styles [11]. As a general rule, regardless of the reservation style, it is being standardized that two (or more) LSPs may be merged into one MP2P connection at certain LSRs, only if the paths of the LSPs are the same starting from that LSR.

This would override the consideration of the tree-shape (discussed previously in §II) with regard to label merging, i.e. labels may be merged for LSPs that are intersected in many places. This prevents merged LSPs from looking like inverse trees, although all of them end in the same egress LSR. Without the consideration, label binding must be performed according to the following rule. If two LSPs are intersected elsewhere except in their ending segment, they *must* use a different label; otherwise, they use the same label inside the intersected segment. This could require that LSRs use more than one outgoing label per MP2P connection, since merged LSPs could be intersected in non-ending segments of the MP2P connection. This type of label merging reduction - without the consideration of the tree-shape - is called *full label merging* so as to distinguish it from the traditional MP2P trees.

Consideration of the tree-shape may limit the possibilities of reducing label spaces with respect to this proposal (full label merging). For example, consider the LSP configuration shown in Fig. 1. The number of labels used is reduced by up to 16 labels considering that a) LSR N10 and N11 use two labels - one for LSPs B and C, and another for D and E - and b) each link in the path $N4 \rightarrow N2$ uses one label for all LSPs. With the consideration the number of reduced labels could be 14 or 15 at most, but never 16 as was just claimed. Refer to Fig. 3 for visualization. Note that we merge all LSPs into one single connection:

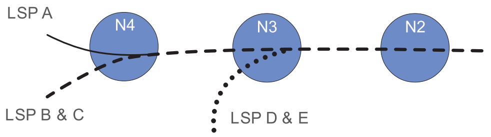


Fig. 3. Full Label Merging Solution

Fact 2: With full label merging, all the LSPs going to an egress LSR can be placed in only *one* MP2P connection, hence a better reduction is achieved.

In other words, the total number of MP2P connections in a network is equal to the number of LSRs, and all the LSPs with the same egress LSR belong to the same MP2P connection.

Since all of them can be joint together with a single MP2P connection, the problem of deciding which LSP routes must be chosen to create an MP2P connection no longer exists. As a result, the Merging Problem NP-complete proof presented in [8] lacks of basis.

V. The Full Label Merging Algorithm: An Optimal Solution in Polynomial-Time

Based on the ideas presented in the previous section, the proposed label binding rules for full label merging are formalized with an easy online polynomial algorithm using only label swapping operations in the NHLFEs.

The following algorithm, the Full Label Merging algorithm, assigns the label for a new LSP L. The algorithm is recursive in terms of the network LSRs. In each recursive call, an LSR N is queried for a label. The first queried LSR is the ingress LSR of the LSP.

```
Function FullLabelMerging(N,L)
Input: LSR $N$
Input: LSP $L$
Output: Requested Label
begin
    $i \leftarrow$ incomingInterface($N, L$) ;
    $o \leftarrow$ outgoingInterface($N, L$) ;
    if $N$ == egress($L$) then
        $e \leftarrow NHLFE[i, o, -]$ ;
    else
        $x \leftarrow$ FullLabelMerging($N + 1, L$) ;
        $e \leftarrow NHLFE[i, o, x]$ ;
    if $e = \emptyset$ then
        $l \leftarrow$ newLabel($N$) ;
        $e \leftarrow$ newEntry($i, o, l$) ;
        addEntry($N, e$);
    return incomingLabel($e$);
end
```

As can be seen in the algorithm, label binding takes place regardless of the LSP route itself, but mainly in the incoming and outgoing interfaces in the queried LSR N that the new LSP L traverses. Note that since two LSPs share an ending segment, they share the same outgoing and incoming labels as well. However, if they diverge, their labels become different.

If later they converge together again, their labels will still be different because they will have different previously assigned outgoing labels. Therefore, no incorrect label swapping or forwarding can take place here.

As there is no decision problem regarding label binding (Fact 2), the order in which LSP routes are given to the algorithm is not relevant.

Fact 3: The solutions found by the algorithm are optimal since a) there is no decision problem (Fact 2) and b) the reduction in label merging schemes depends on the selected branches and the full label merging solution takes them all at the same time (Fact 1); no possible improved solution could be computed. Since the algorithm finds the optimal solution to the problem in polynomial time, the MERGING PROBLEM is not NP-complete.

Mapping the proposed algorithm to a protocol would be simple, since: a) it is recursive on the LSRs and b) the output label of each iteration is computed only with the local information of the LSR. Moreover, the algorithm behaves similarly to RSVP-TE in terms of assigning labels.

VI. REMARKS ON THE IMPROVEMENT OBTAINED BY FULL LABEL MERGING

Although the main point of discussion of this letter is to show a simple way (Full Label Merging algorithm) to compute the optimal label binding with label merging, the number of reduced labels that the full label merging solution can improve (because of the absence of the tree consideration) with respect to MP2P trees was analyzed through simulations as well.

In all simulations, the network topologies are generated using *Power Laws* - described in [12] - with 20 nodes and rank exponent -0.7. In each simulation, 4 of the 20 LSRs are selected as edge routers (LER), i.e. assuming both ingress and egress functions.

The LSPs routes used are computed using a multi-objective evolutionary algorithm in which several QoS metrics are taken [13]. It should be mentioned that, since the routing solution is multi-objective, a set of possible routes could be computed for a single set of demands. In this case, the average of the reductions is taken. The number of generated demands (LSP routes) varies between the pairs of LERs from 12 to 100. Then, the number of used labels is computed using both Bhatnagar's algorithm and the one presented here; their relative difference is taken as the improvement factor.

Table I shows that 60% of the tests performed achieved a better reduction using full label merging, while the other 40% maintained the same reduction.

VII. CONCLUSIONS AND NEW DIRECTIONS

In this letter we analyze the MERGING PROBLEM from a practical point of view and arrive at the conclusion that it can be solved optimally in polynomial time - with the proposed Full Label Merging algorithm - since the decision problem it involves is incorrect and the MERGING PROBLEM is *not* NP-complete. The problem was not only solved optimally in an efficient way (poly-time algorithm), but also showed a better reduction factor: less labels were used in the network in most of the simulations and the same reduction was maintained in the rest of them.

TABLE I
IMPROVEMENT OF FULL LABEL MERGING WITH RESPECT TO MP2P TREES.

Percentage of Tests		Improvement Ratio
40%		not improved
60%	59%	$\geq 1\%$
	55%	$\geq 2\%$
	25%	$\geq 3\%$
	15%	$\geq 4\%$

As for future work, a scenario in which the label space is improved by means of the label stacks could be considered, e.g. [7]. This new consideration may convert the label space reduction problem to NP-complete and therefore new heuristics and optimization models may be proposed.

REFERENCES

[1] 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.*

[2] D. Applegate and M. Thorup, "Load optimal MPLS routing with N+M labels," in *Proc. IEEE INFOCOM 2003*, pp. 555–565.

[3] H. Saito, Y. Miyao, and M. Yoshida, "Traffic engineering using multiple MultiPoint-to-Point LSPs," in *Proc. IEEE INFOCOM 2000*, pp. 894–901.

[4] 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.

[5] F. Ramos, E. Kehayas, J. M. Martinez, R. Clavero, J. Marti, L. Stampoulidis, D. Tsiokos, H. Avramopoulos, J. Zhang, P. V. Holm-Nielsen, N. Chi, P. Jeppesen, N. Yan, I. T. Monroy, A. Koonen, M. Hill, Y. Liu, H. Dorren, R. V. Caenegem, D. Colle, M. Pickavet, and B. Riposati, "IST-LASAGNE: towards all-optical label swapping employing optical logic gates and optical flip-flops," *IEEE J. Select. Areas Commun.*, vol. 23, no. 10, pp. 2993–3011, Oct. 2005.

[6] E. Rosen, A. Viswanathan, and R. Callon, *Multiprotocol Label Switching Architecture*, IETF, Jan. 2001, RFC 3031.

[7] F. Solano, R. Fabregat, and J. Marzo, "Full label space reduction in MPLS networks: Asymmetric merged tunneling," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 1021–1023, Nov. 2005.

[8] S. Bhatnagar, S. Ganguly, and B. Nath, "Creating Multipoint-to-Point LSPs for traffic engineering," in *Proc. IEEE Workshop on High Performance Switching and Routing (HPSR 2003)*, June 2003, pp. 201–207.

[9] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, IETF, Sep. 1997, RFC 2750.

[10] D. Awduche, L. Berger, D.-H. Gan, T. Li, V. Srinivasan, and G. Swallow, *Extensions to RSVP for LSP Tunnels*, IETF, Dec. 2001, RFC 3209.

[11] S. Yasukawa, *Supporting Multipoint-to-Point Label Switched Paths in Multiprotocol Label Switching Traffic Engineering*, IETF, Feb. 2007, draft-yasukawa-mpls-mp2p-rsvpte-02.

[12] 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.

[13] Y. Donoso, R. Fabregat, and J. Marzo, "A multi-objective optimization scheme for multicast routing: A multitree approach," *Telecommunication Syst. J.*, vol. 27, no. 2–4, pp. 229–251, Oct. 2004, special issue: networks.