On the calculation of inter-domains point to multipoint paths in MPLS networks

Mohamad Chaitou

Department of Computer Science,
Faculty of Science,
Lebanese University,
Nabatieh, Lebanon
Email: mohamad.chaitou@ul.edu.lb

Abstract: We propose a method to compute point to multipoint (P2MP) traffic engineered paths that cross several MPLS domains. It relies on the use of multiple path computation elements (PCEs), in the goal of calculating a P2MP tree along a given tree of domains by using a recursive path computation technique. Indeed, the calculation is started by the PCEs of the destination domains, i.e., those ones that are not connected to any domain downstream and it is continued upstream from PCE to PCE until reaching the P2MP tree source. The trade-off between the tree optimality and the number of paths to be calculated is hence highlighted and discussed by means of simulations.

Keywords: multi protocol label switching; MPLS; optimisation; inter-domain networks; multicast; simulations.

Reference to this paper should be made as follows: Chaitou, M. (2016) 'On the calculation of inter-domains point to multipoint paths in MPLS networks', *Int. J. Network Science*, Vol. 1, No. 1, pp.80–100.

Biographical notes: Mohamad Chaitou received his MS in Computer and Communications Engineering from the Lebanese University, Lebanon in 2002, and MS in Networking from the University of Pierre et Marie Curie, France, in 2003. In 2006, he obtained his PhD in Computer Science from Telecom SudParis (TSP), France. From 2007 to 2008, he worked at Orange Labs where He contributed to the development of the MPLS traffic engineering technology. From 2008 to 2010, he joined Bouygues Telecom as a Consultant Engineer where he worked on the convergence of IP backbone networks. Since 2010, he is an Assistant Professor at the Lebanese University. His actual research interests include: traffic engineering of enterprise networks, cloud computing, multicast, performance evaluation and probabilistic modelling.

1 Introduction

Multi protocol label switching – traffic engineering (MPLS-TE) is widely used by networking operators in order to provide value added services such that bandwidth reservation for multimedia-based traffic and fast reroute in order to provide redundancy in less than 50 ms. This makes from MPLS-TE an excellent candidate to support real-time multicast traffic that requires high quality of service (QoS) guarantees such that IPTV, video conferencing, e-teaching, distributed processing, etc. This is due to the use of the RSVP-TE protocol (Awduche et al., 2001) for signalling.

In this paper, we emphasise on the support of multicast point to multipoint (P2MP) trees in inter-domain MPLS networks.

In the context of MPLS, a P2MP tree is called P2MP TE-LSP (Aggarwal et al., 2007) where LSP stands for label switched path.

A domain is a set of nodes belonging to the same address management space such as an interior gateway protocol (IGP) area or an autonomous system (AS).

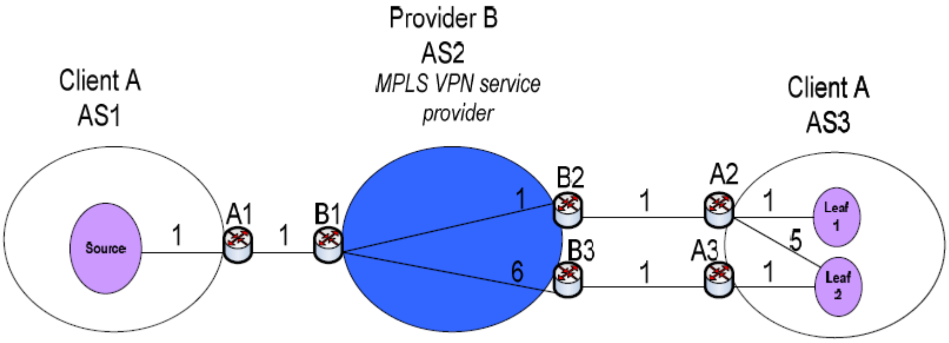
In an inter-domain P2MP TE-LSP, the root node and leaves are spread out over multiple domains, while in intra-domain P2MP TE-LSP these nodes reside in the same domain.

The calculation of a P2MP TE-LSP should take into account the minimisation of the tree cost (i.e., the sum of links' metrics composing the tree). This is called the P2MP TE-LSP optimisation.

The optimisation of an intra-domain P2MP TE-LSP is known in the literature as the Steiner problem in networks (Winter, 1987) which is NP-complete. Several heuristics have been proposed in order to find an approximate solution. The heuristic proposed by Takahashi and Matsuyama (1980) is characterised by a low computation time in the case where leaves are frequently added and/or pruned from the P2MP TE-LSP which is the case of several applications such that video on demand (VoD) or IPTV. We will rely on this heuristic for the procedures described later.

Inter-domain P2MP TE-LSPs are encountered typically in the context of MPLS-virtual private networks (VPN) (Andersson and Madsen, 2005). This is illustrated in Figure 1 where we show an MPLS-VPN service provider, called provider B and a client named A. In this case, the root node (i.e., the source) in AS1 should send the traffic to two leaves in AS3 via AS2, where AS1 and AS3 belong to client A and AS2 belongs to provider B. The number above each link represents its metric.

Figure 1 An example of an inter-domain P2MP TE-LSP (see online version for colours)



A serious challenge arises against the optimisation of the inter-domain P2MP TE-LSP. This is because each operator will hide the topology information of its domains from the other operators. The topology information of a domain includes the nodes, links and their metrics.

Without this information the optimisation of the inter-domain P2MP TELSP is not possible. For instance, in Figure 1, client A does not reveal the topology of AS1 and AS3 to provider B and the latter does not show the topology of AS2 to client A.

Moreover, the optimisation of the tree calculation in each domain separately will not lead to an optimal P2MP TE-LSP. This is illustrated in Figure 2 where we show the tree obtained after such optimisation. The total cost of this tree is 13. However, one may notice that the optimal tree is that shown in Figure 3 with a cost of 10.

Figure 2 The tree obtained by performing the optimisation inside each domain independently from the other ones (see online version for colours)

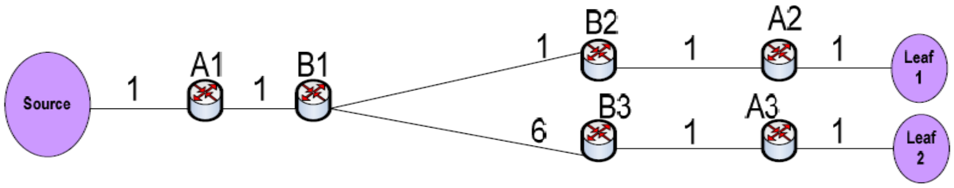
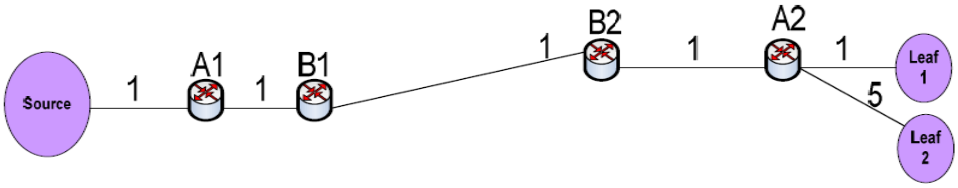


Figure 3 The optimal tree (see online version for colours)



In this paper, we propose a method called multicast backward recursive procedure (MBRP) in order to calculate an optimised inter-domain P2MP TE-LSP, called henceforth P2MP TE-LSP. It relies on the use of a calculation server called path computation element (PCE) and the PCEP protocol (Vasseur and Le Roux, 2009). We suppose that a PCE is responsible of the calculation of the P2MP TE-LSP segment in one domain. These segments joined together form the P2MP TE-LSP.

The calculation is recursive because it is started by the PCEs of the destination domains, i.e., those ones that are not connected to any domain downstream, such that domain AS3 in Figure 1. The calculation is continued upstream from PCE to PCE until reaching the P2MP tree source. We discuss the idea of reducing the calculation burden at the expense of increasing the cost of the calculated tree by introducing a mechanism called 'simplified MBRP'. We illustrate this trade-off by using extensive simulations.

The remainder of this paper is organised as follows. In Section 2, we present an overview of the related works. In Section 3, we illustrate how an optimal inter domain P2MP TE-LSP can be calculated. Section 4 provides a signalling example showing the PCEP message exchange and the interactions between RSVP-TE and PCEP in order to establish the P2MP TE-LSP. Section 5 presents the results of the simulation study. Finally, Section 6 concludes this paper.

2 Related works

The idea of recursive calculation has been previously defined in Vasseur et al. (2009) for point to point (P2P) path calculation. However, P2MP tree calculation has received a little attention in the literature.

In Wu et al. (2011), we found a comparison study between three existing PCE-based schemes for P2MP LSP computation and a proposal of a new algorithm called multi-domain minimum-cost path heuristic (MDMPH).

The three compared algorithms are:

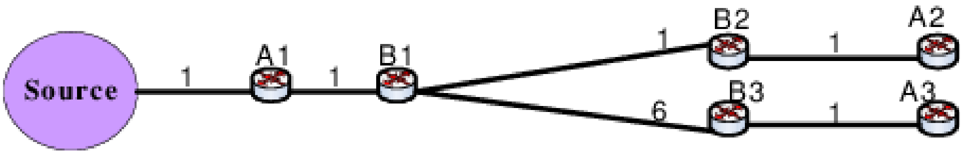
- Per-domain algorithm: it consists of performing computation in each domain separately (Vasseur et al., 2008). Clearly, this leads to a sub-optimal tree as indicated before.
- Extended BRPC-based (EBB) algorithm: It simply calculates a shortest path tree (SPT) by calculating each P2P path recursively using the method proposed in Vasseur et al. (2009). Intuitively, a SPT is a sub-optimal tree
- Core tree-based (CTB) algorithm: in this method proposed in Zhao et al. (2014), a core tree is a path tree which satisfies the following conditions:
  - a the root of the core tree is the source node
  - b the leaf of the core tree is the boundary node (BN) in the destination domain (destination domain is a domain which contains destination nodes, and it is not a transit domain)
  - c the transit and branch nodes of the core tree are BNs in the transit and branch domains.

  With CTB, computing the complete P2MP LSP path tree is done in two phases:

  1. build the P2MP LSP core tree
  2. graft destinations to the P2MP LSP core tree.

There are two drawbacks for this method. First, it assumes that a destination domain cannot be a transit domain. In other words, it assumes that a transit domain cannot contain a destination leaf. Second, the resulting tree is sub-optimal. To illustrate this refer to Figure 1. Using the CTB approach, the P2MP core tree will be as depicted in Figure 4 below because A2 and A3 are the BNs. Now, by grafting destinations (leaf1 and leaf2), we get the tree of Figure 2 which is suboptimal.

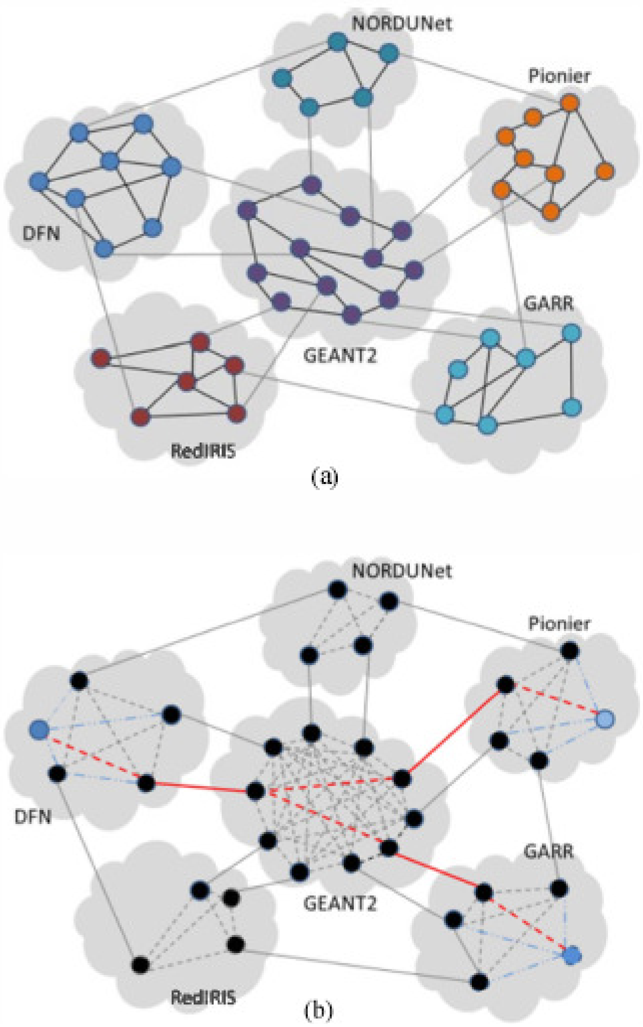
Figure 4 Core-based tree (see online version for colours)



The MDMPH algorithm proposed in Wu et al. (2011) works as follows. Each domain calculates full mesh of P2P shortest paths between all its BNs. The resulting topology is called full mesh virtual topology and it is exposed by the domain to all other domains using PCEs. Then, the heuristic presented in Takahashi and Matsuyama (1980) is applied on the interconnected virtual topologies to find the tree.

Figure 5 (Wu et al., 2011) illustrates this concept.

Figure 5 The MDMPH algorithm (see online version for colours)



In Figure 5(a), the physical topology is presented, while in Figure 5(b) the interconnected virtual topology is illustrated with an example of an interdomain tree.

This algorithm leads also to sub-optimal tree because it prohibits having branching nodes inside a domain.

3 The MBRP

3.1 Assumptions and notations

We consider establishing an optimised P2MP MPLS TE-LSP connecting a 'root node', i.e., a source node, located in a domain called the 'root domain', to multiple destinations called 'leaf nodes', or simply «leaves», located in one or several domains. All of these domains and their interconnection are expected to be known a priori. In the general case, we can see the interconnection between these domains as a P2MP tree, called 'domain tree'. The root domain of the 'domain tree' is the one that contains the 'root node'. A 'transit domain' is a domain that is connected simultaneously to a single domain upstream and to one or more domains downstream. The transit domain receives traffic belonging to the MPLS TE-LSP from the upstream domain and transmits it to downstream domains. A 'destination domain' is a domain that has no downstream domains. The 'root domain' has no upstream domain. All domains, that is, the 'root domain', the 'transit domains' and 'destination domains' may contain 'leaves' of the MPLS TE-LSP.

The interconnection between two domains is done through the 'edge nodes' or 'BNs'. A BN is called area border router (ABR) in the case where the domains are IGP areas and autonomous system border router (ASBR) in the case where the domains are ASs.

Figure 6 shows an example of the proposed scenario, where we illustrate a root domain associated with a PCE named 'PCE *R*', a transit domain associated with 'PCE *i*' and two destination domains associated with 'PCE *d*1' and 'PCE *d*2', respectively.

In the general case, we suppose that a transit domain, denoted as domain *i*, is connected to its upstream domain, called domain *i* – 1, with k_{0i} BNs belonging to domain *i*. These BNs are called entry BNs and are denoted as:

$$BN-en(j,i),\ j=1,2,...,k_{0i}.$$

For instance, in Figure 6, we have for domain *i*:

$$k_{0i}=2,$$

$$BN-en(1,i)=B1$$

and

$$BN-en(2,i)=B2.$$

In addition, we suppose that domain *i* is connected to $Di \geq 1$ downstream domains. For example, in Figure 6, $Di = 2$. Domain *i* is connected to each downstream domain *d*, $d = 1, 2, \ldots, Di$ by k_d BNs named exit BNs and denoted as:

$$BN-ex_d(j,i),\ j=1,2,...,k_d.$$

For instance, in Figure 6, we have for domain *i*:

- $k_1 = 2$ BNs, which are $B3$ and $B4$. That is:

  $$BN - ex_1(1, i) = B3$$

  and

  $$BN - ex_1(2, i) = B4$$

  and

- $k_2 = 2$ BNs, which are $B5$ and $B6$. That is:

  $$BN - ex_2(1, i) = B5$$

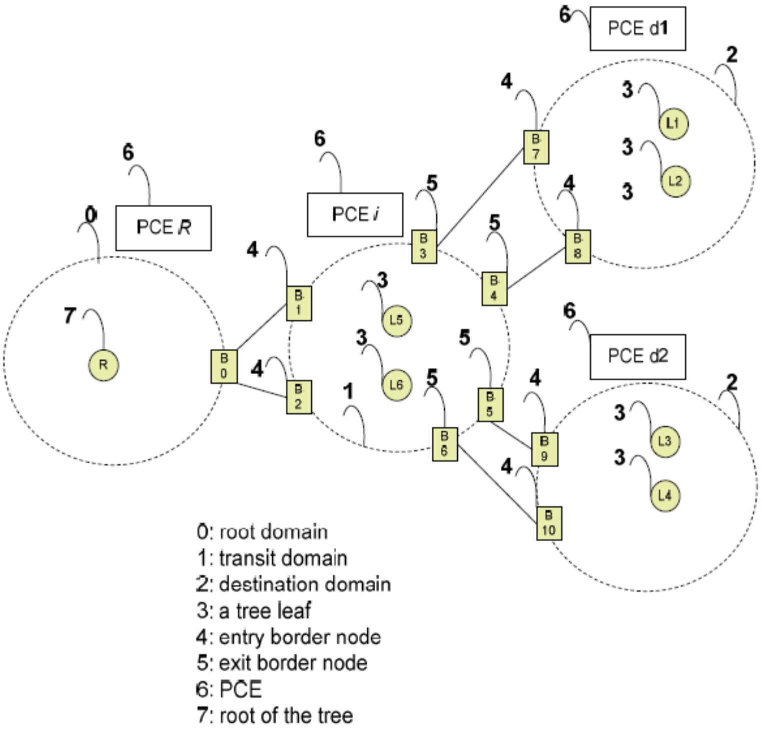
  and

  $$BN - ex_2(2, i) = B6.$$

Furthermore, as it has been mentioned previously, each domain may contain leaves. We assume that each domain i, contains $X_i \geq 0$ leaves. For example, in Figure 6, we have for domain R, $X_R = 0$. That is, the root domain contains no leaves. However, for domain i, we have $X_i = 2$ leaves, which are L5 and L6. Moreover, for domain $d1$, we have $X_{d1} = 2$ leaves, which are L1 and L2. Similarly for domain $d2$, we have $X_{d2} = 2$ leaves, which are L3 and L4.

We assume that the topology of a domain should not be revealed to its neighbours. As it has been previously mentioned, this assumption is interpreted by the fact that two adjacent domains may represent in the reality two different operators.

Figure 6 An example of the domain tree (see online version for colours)



3.2 The calculation procedure

We will consider that there is one PCE per domain.

Step 1 The first step consists of sending a request message from the root node to the PCE of the root domain. This message is called PCReq (Vasseur and Le Roux, 2009) and it contains an explicit request for calculating the P2MP TE-LSP via the recursive procedure. Hence, this message is relayed from a PCE to its downstream ones along the domain tree until reaching the PCEs responsible of the destination domains.

Step 2 Path calculation in a destination domain, denoted as domain *n*.

Using the previously mentioned notations, a domain *n* has k_{0n} entry BNs and X_n leaves. The PCE of domain *n*, called PCE *n*, should calculate a set of paths and send them to its upstream PCE. Each of these paths has one root selected among the entry BNs and a number of leaves ≥ 1 selected among the X_n leaves. Hence, such a path can be P2P if its number of leaves = 1 or P2MP its number of leaves > 1. In order to ensure the optimality of the P2MP TE-LSP, PCE *n* should compute the set of all the combinations of paths such that:

1 each combination contains a set of paths

2 each path inside a combination has one root selected randomly among the set of k_{0n} entry BNs

3 each path inside a combination has a set of leaves selected randomly among the X_n leaves

4 the sum of leaves of all paths in one combination = X_n, without having any common leaf between two paths of the same combination

5 two different combinations must not have any path in common.

In addition, PCE *n* determines the cost for each path using the cost definition in Section 1.

For instance, in Figure 6, suppose that domain $n = d1$, then $k_{0d1} = 2$ (B7 and B8) and $X_{d1} = 2$ (L1 and L2). The set of all combinations as described above are then illustrated by Table 1.

Table 1 The set of all combinations for domain *d*1

Combination identifier	*Combination cost*	*Path #*	*Path root*	*Path leaves*
1	cost d1 1	1	B7	L1, L2
2	cost_d1_2	1	B7	L1
		2	B8	L2
3	cost_d1_3	1	B7	L2
		2	B8	L1
4	cost_d1_4	1	B8	L1, L2

For $n = d2$ we get Table 2.

Table 2 The set of all combinations for domain *d*2

Combination identifier	*Combination cost*	*Path #*	*Path root*	*Path leaves*
1	cost_d2_1	1	B9	L3, L4
2	cost_d2_2	1	B9	L3
		2	B10	L4
3	cost_d2_3	1	B9	L4
		2	B10	L3
4	cost_d2_4	1	B10	L3, L4

The set of all the combinations of paths computed in this step is called $VMCT^d$ where VMCT stands for 'virtual minimum combination of trees'. PCE d stores this set in a database containing the following fields:

1 A unique identifier for each combination in the set. This identifier has a local significance between PCEs and it can be used to lookup for a specified combination

2 The cost of the combination: the term *cost_d_i* represents the cost of the combination with identifier *i* from domain *d*.

3 The roots of paths constituting the combination.

4 The leaves of paths constituting the combination.

5 The intermediate nodes constituting each path of each combination.

A combination is a set of paths as it has been previously mentioned, and hence it can be represented within a domain by: the set of roots and leaves of its paths, the cumulative cost of its paths and a unique identifier. However, when the combination is sent from a PCE to its upstream, the leaves and intermediate nodes should not be included because they are part of the topology of the downstream domain that must not be revealed to the upstream one according to our assumption.

Now let N_n incarnates the number of combinations in domain *n*. For instance (see Figure 6, Table 1 and Table 2), $N_{d1} = N_{d2} = 4$. Next, we express mathematically N_n, $\forall k_{0n}$ and $\forall X_n$.

For $k_{0n} \geq 1$, we have:

$$N_n = k_{0n}^{X_n} \tag{1}$$

The proof of (1) is easy. Indeed, it suffices to take into account the calculation of all combinations as indicated by the points 1 to 5 above.

- Step 1:

This step concerns transit domains. A transit PCE, denoted as PCE *i*, receives from each downstream PCE *d* where $d = 1, 2, \ldots, Di$ the set of combination $VMCT^d$.

However, since the topology of domain *d* cannot be revealed to domain *i*, PCE *d* cannot include in the sent $VMCT^d$ the leaves or the nodes constituting the paths of the combinations. Hence, it sends $VMCT^d$ to PCE *i* as the set of roots of each combination in $VMCT^d$ in addition to the individual cost of each combination, and to the unique identifier per combination.

Note that the roots of $VMCT^d$ are known to domain *i* although they are part of domain d because domain *i* establishes the peering with these nodes.

In order to ensure that the combinations in $VMCT^i$ have minimum costs, $VMCT^i$ is determined as follows:

1 PCE *i* computes the set of combinations such that each combination covers the leaves X_i, i.e., the leaves inside domain *i*. The number of such combinations, denoted as N_i is given by (1) when replacing subscript *n* by *i*. Call this set $VMCT^i_{initial}$.

2 PCE *i* takes each combination in $VMCT^i_{initial}$ separately and adds to it the roots of *Di* combinations simultaneously, where:

- D_i is the number of domains downstream to domain *i*
- each of those D_i combinations is taken from a different downstream domain
- the addition of a root to a combination means that this root node joins the combination according to the heuristic used to calculate the P2MP TE-LSP.

The cost for each resulted combination is the cost of the one taken from $VMCT^i_{initial}$ added to the cost of the D_i combinations described above.

The total number of the resulted combinations per one combination taken from $VMCT^i_{initial}$ is simply:

$$\prod_{d=1}^{Di} N_d,$$

And since $VMCT^i_{initial}$ has N_i combinations, then the total number of combinations computed by PCE *i*, denoted by C_i, is expressed by:

$$C_i = N_i \times \prod_{d=1}^{Di} N_d, \tag{2}$$

where N_i is given by (1) when replacing subscript *n* by *i*.

3 For each combination from $VMCT^{i}_{initial}$, PCE *i* selects the combination with the minimal cost among the resulting $\prod_{d=1}^{Di} N_d$ combinations computed in Step 2 above and adds it to *VMCT*^{i}. Hence, the total number of combinations in *VMCT*^{i}, i.e., those to be sent to PCE $i - 1$ is given by N_i. As mentioned before, since the topology of domain *i* cannot be revealed to its upstream domain, PCE *i* sends *VMCT*^{i} to PCE $i - 1$ as the set of roots of each combination, the combination cost and the unique identifier of the combination.

Let us illustrate how PCE *i* in Figure 6 determines *VMCT*^{i}.

First, PCE *i* receives from PCE *d*1 the set of combinations *VMCT*^{d1} expressed by Table 1. Similarly, it receives from PCE *d*2 the set of combinations *VMCT*^{d2} expressed by Table 2.

Then PCE *i* applies the following steps:

1 PCE *i* computes the set of combinations covering the leaves inside domain *i* that is L5 and L6. This set of combinations is called $VMCT^{i}_{initial}$. It is given by Table 3.

2 PCE *i* takes each combination from $VMCT^{i}_{initial}$ separately and adds to it: the roots of one combination from *VMCT*^{d1} and the roots of one combination from *VMCT*^{d2} simultaneously. Hence, for each combination from $VMCT^{i}_{initial}$, the resulting number of combinations obtained is $N_{d1} \times N_{d2} = 4 \times 4 = 16$. PCE *i* determines the one with the minimum cost among these 16 combinations and adds it to *VMCT*^{i}.

3 Proceeding similarly for the remainder of combinations in $VMCT^{i}_{initial}$ we conclude that the total number of combinations to calculate in this step is: $C_i = 2^{X_i} \times N_{d1} \times N_{d2} = 2^2 \times 2^2 \times 2^2 = 64$.

Table 3 The set of all combinations in domain *i* including only the leaves inside domain *i*

Combination identifier	*Combination cost*	*Path #*	*Path root*	*Path leaves*
1	cost_i_1	1	B1	L5, L6
2	cost_i_2	1	B1	L5
		2	B2	L6
3	cost_i_3	1	B1	L6
		2	B2	L5
4	cost_i_4	1	B2	L5, L6

3.3 Final step

This last step is performed by PCE1, i.e., the root domain. The root domain is a particular case of a transit domain with k_{01} representing the root node. Since $k_{01} = 1$, $VMCT^1$ will contain only P2MP paths. The one among these paths with the minimal cost is the P2MP TE-LSP that we are searching for.

After the P2MP TE-LSP has been calculated, the root node initiates the control plane signalling procedures in order to establish the P2MP TE-LSP by using the protocol RSVP-TE. In addition, these procedures require some interactions between RSVP-TE and PCEP which will be explained in Section 4 next.

4 A signalling example

The steps for calculating the VMCTs, presented in Section 3, guarantee the optimality of P2MP MPLS TE-LSP. However, to reduce the complexity of the problem we can limit the number of combination of paths that a PCE should calculate by taking into account only a subset of all combinations. This implies a sub-optimality of the calculated P2MP tree in favour of a faster computation time and less overhead burden on the CPU of the PCE. Consequently, a trade-off optimality/complexity exists.

We introduce the ‘simplified MBRP’ mechanism which is the same as the MBRP but amended with the following modification: in each VMCT a combination can have only one root. That is an element of a VMCT is strictly a P2MP tree not a collection of P2MP and/or P2P paths. The aim of this modification is to reduce the number of elements of a VMCT which leads to reducing the calculation overhead. Indeed, using this assumption, equation (1) becomes:

$$N_n = k_{0n}$$

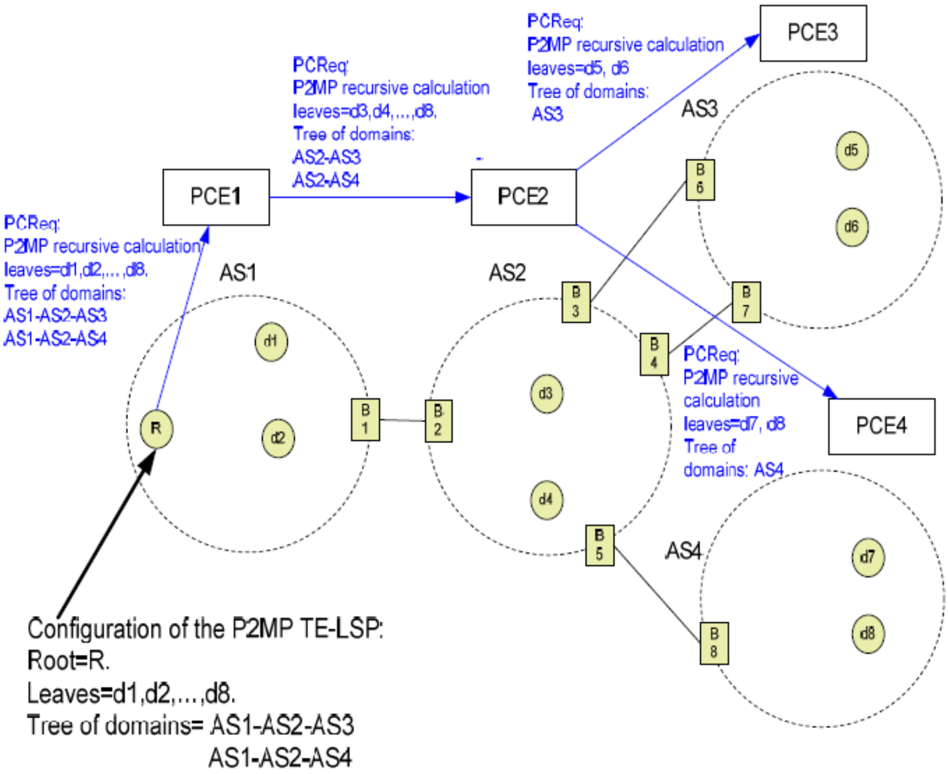
A performance evaluation study is presented in Section 5 next. However, in this section we illustrate a signalling example of the simplified MBRP mechanism.

Figure 7 through Figure 10 show an example of establishing the P2MP TE-LSP through multiple AS domains. We focus on the signalling aspects of the P2MP TE-LSP, that is, the PCEP (Vasseur and Le Roux, 2009) message exchange and the interaction between RSVP-TE and PCEP.

A network administrator configures the P2MP TE-LSP parameters on the root node *R*. These parameters include the domain tree in addition to the classical traffic engineering parameters (e.g., bandwidth).

The signalling of the P2MP TE-LSP is the responsibility of the root node *R* which sends a request message to PCE1 called PCReq. This message is relayed from PCE1 to PCE2 and then to the destination PCEs, i.e., PCE3 and PCE4. This is illustrated in Figure 7.

Figure 7 Exchange of the PCReq messages (see online version for colours)



4.1 Calculations by PCE3

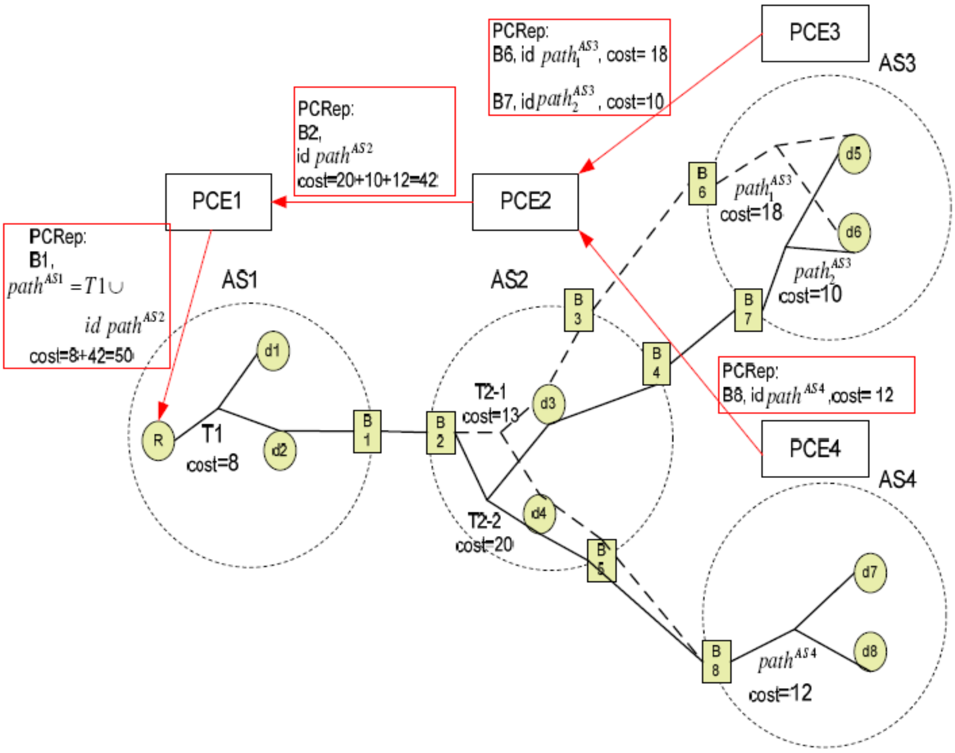
AS3 has two entry BNs, i.e., B6 and B7. Hence, according to the 'simplified MBRP' proposed in this section, PCE3 calculates two paths, denoted as $path_j^{AS3}$, $j = 1, 2$ such that $path_1^{AS3}$ has $B6$ as root and $d5$ and $d6$ as leaves while $path_2^{AS3}$ has $B7$ as root and $d5$ and $d6$ as leaves. These two paths constitute the set $VMCT^{AS3}$ which will be sent from PCE3 to PCE2 via a message called PCRep (Figure 8).

Because AS3 cannot reveal its internal topology to AS2, $VMCT^{AS3}$ is sent as:

$$\left\{\begin{matrix} \left\{root\ of\ \ path_1^{AS3}, \cos t\ of\ \ path_1^{AS3}, id\ of\ \ path_1^{AS3}\right\} \\ \left\{root\ of\ \ path_2^{AS3}, \cos t\ of\ \ path_2^{AS3}, id\ of\ \ path_2^{AS3}\right\} \end{matrix}\right\}$$

$$= \left\{\left\{B6, 18, id\ of\ \ path_1^{AS3}\right\}, \left\{B7, 10, id\ of\ \ path_2^{AS3}\right\}\right\}$$

Figure 8 Exchange of the PCRep messages (see online version for colours)



4.2 Calculations by PCE4

Similarly, PCE4 calculates $VMCT^{AS4}$ which will contain only one path called $path^{AS4}$ because AS4 has only one entry BN, i.e., BN8. $path^{AS4}$ has as root BN8, and as leaves $d7$ and $d8$ (Figure 8). $VMCT^{AS4}$ is sent to PCE2 via a PCRep message as $\{B8, 12, \textit{id of } path^{AS4}\}$.

4.3 Calculations by PCE2

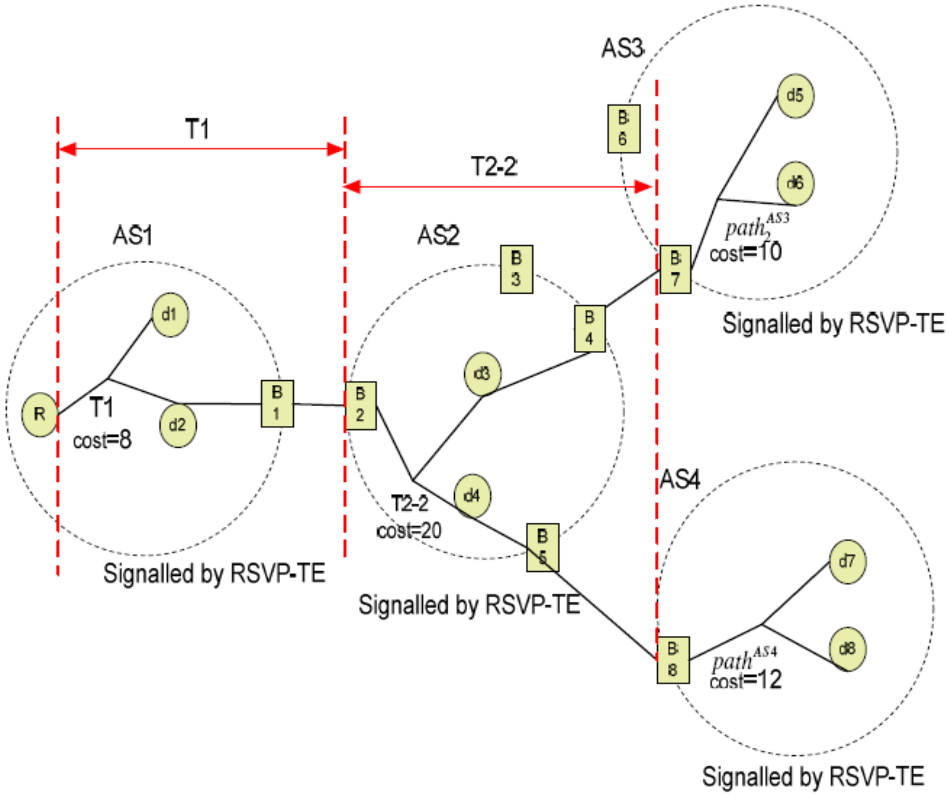
AS2 is a transit domain which has AS1 as an upstream domain and AS3 and AS4 as downstream domains. Hence, the calculation of $VMCT^{AS2}$ will use the following procedures.

Firstly, PCE2 calculates $VMCT_{initial}^{AS2}$ which contains only one path: that with root $B2$ and with leaves $d3$ and $d4$.

Then, PCE2 adds the root of $path_1^{AS3}$ (i.e., B6) and the root of $path^{AS4}$ (i.e., B8) to AS $VMCT_{initial}^{AS2}$ which yields a path called T_{2-1} in Figure 8. T_{2-1} has: as root B2, as leaves *d*3, *d*4, *B*6, *B*8 and as cost 13.

Another possibility consists of adding the root of $path_2^{AS3}$ (i.e., *B*7) and the root of $path^{AS4}$ (i.e., *B*8) to $VMCT_{initial}^{AS2}$ which yields a path called T_{2-2} in Figure 8 (see also Figure 9). T_{2-2} has as: root *B*2, as leaves *d*3, *d*4, *B*7, *B*8 and as cost 20.

Figure 9 The P2MP TE-LSP segments that are signalled by RSVP-TE (see online version for colours)



As for $VMCT^{AS2}$, it will contain only one path because AS2 has only one entry BN. This path is called $path^{AS2}$ and it is the one with the minimal path between the following two paths: $T_{2-1} \cup path_1^{AS2} \cup path^{AS4}$ and $T_{2-2} \cup path_2^{AS3} \cup path^{AS4}$.

Note that PCE2 does not know the topology of $path_1^{AS3}$, $path_2^{AS3}$ and $path^{AS4}$ but it knows their costs. Hence, it proceeds to the following calculation:

The cost of $T_{2-1} \cup path_1^{AS3} \cup path^{AS4} = 13+18+12 = 43$ while the cost of $T_{2-2} \cup path_2^{AS3} \cup path^{AS4} = 20+10+12 = 42$.

Hence, $VMCT^{AS2} = path^{AS2} = T_{2-2} \cup path_2^{AS3} \cup path^{AS4}$. This path is sent to PCE1 via a PCRep message as $VMCT^{AS2}$ = {*B*2, 42, *id of path*^{AS2}} (see Figure 8).

4.4 Calculations by PCE1

Lastly, PCE1 calculates a path that has as root node *R* and as leaves *d*1, *d*2 and *B*2. This path is called T_1 in Figure 8 (see also Figure 9). The final path, representing the P2MP TE-LSP is then $path^{AS1} = T_1 \cup path^{AS2}$ which has a cumulative cost = 8 + 42 = 50. Note that PCE1 does not know the topology of $path^{AS2}$ but it knows its cost.

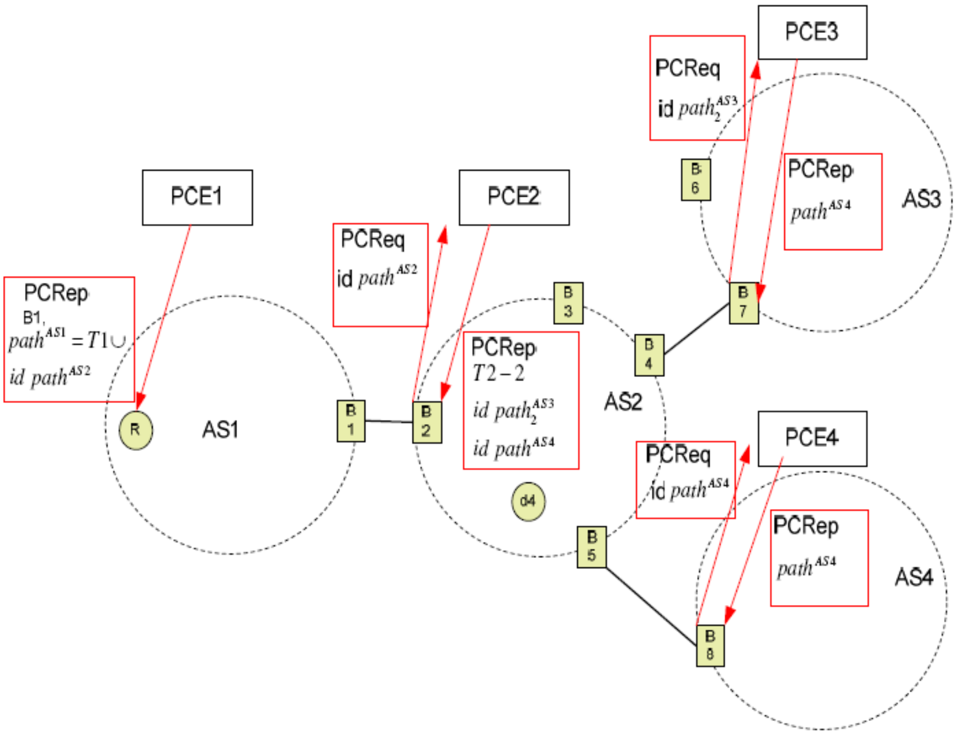
4.5 Interactions between RSVP-TE and PCEP

Now, an RSVP-TE message will be initiated by node *R* to establish T_1.

Node *R* incorporates in this message the path information it received from PCE1 that is (see Figure 8):

$$path^{AS1} = T_1 \cup id\ \ path^{AS2}.$$

Figure 10 Interaction between RSVP-TE and PCEP (see online version for colours)



When this message reaches *B*2, it will sends a PCReq message to PCE2 in order to obtain $path^{AS2}$. This is illustrated in Figure 10, where PCE2 replies by sending a PCRep message to *B*2 containing:

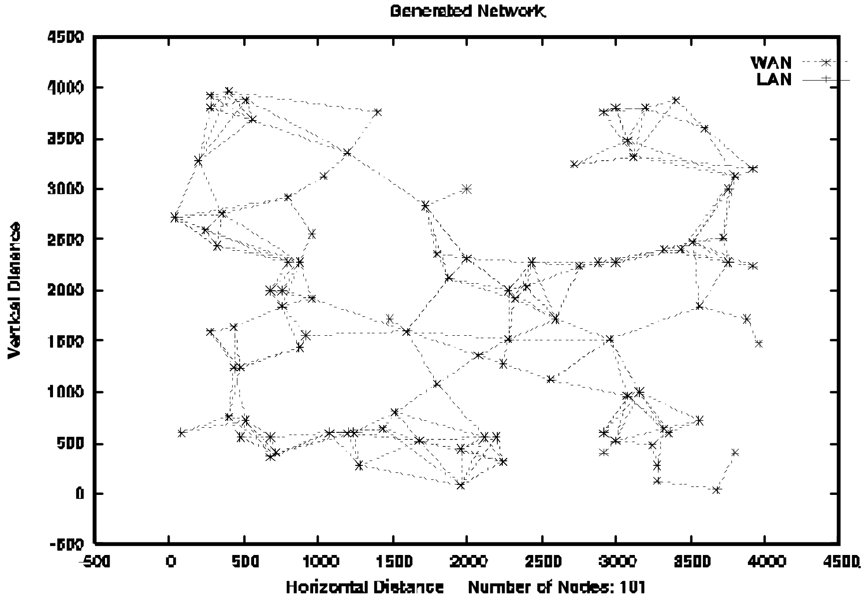
$$path^{AS2} = T_{2-2} \cup id\ \ path_2^{AS3} \cup id\ \ path^{AS4}$$

Hence, $B2$ initiates the establishment of $T_{2\text{–}2}$ in AS2 via an RSVP-TE message that will be sent through $T_{2\text{–}2}$. When this message reaches $B7$, the latter requests to obtain $path_2^{AS3}$ from PCE3, as shown in Figure 10. Then, $B7$ gets $path_2^{AS3}$ via a PCRep message and proceeds to signalling this path via RSVP-TE through AS3. Similarly, the same actions are performed by $B8$ in order to signalling $path^{AS4}$, which completes the establishment procedure of the inter-domain P2MP TE-LSP.

5 Simulation results

We performed a simulation study, using a simulator written in *C*. In this study, we used four ASs connected together according to Figure 6. Each AS network is composed from a numbers of PEs and Ps. The peering points are selected randomly between PEs. For each AS, we used ten network instances with a meshed topology generated by the Tiers topology generator (ns-topogen, http://www.isi.edu/nsnam/ns/nstopogen.html). Each instance has 101 nodes. The number of links for an instance is between 428 and 484. In each instance we assume that there are 80 PEs and 21 Ps distributed randomly around the network. Figure 11 provides the topology of one of the ten instances. The results presented next are the average over the ten instances. The leaves are selected randomly and are spread out over the four ASes. The root of the tree is also selected randomly.

Figure 11 A mesh network instance



We evaluated the following algorithms: CBT, MDMPH, MBRP and the simplified MBRP mechanisms.

The evaluation parameters are:

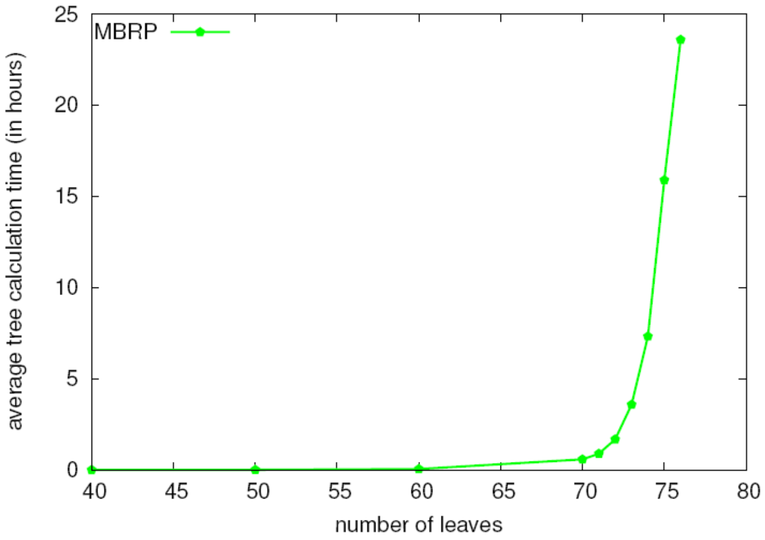
1 The average cost of a tree.

2 The average number of hops from the tree root to a tree leaf. This parameter is important in evaluating the average delay from the source to each destination.

3 the tree calculation time.

Figure 12 illustrates the average tree calculation time when using MBRP on a PC workstation with the following characteristics: 1.66 GHz, dual core, 2 GB RAM. It can be shown that the calculation time remains less than one hour if the number of leaves is below than 70. When the number of leaves increases above this value, the calculation time increases exponentially. For a number of leaves greater than 74 the calculation time becomes greater than one day (24 hours).

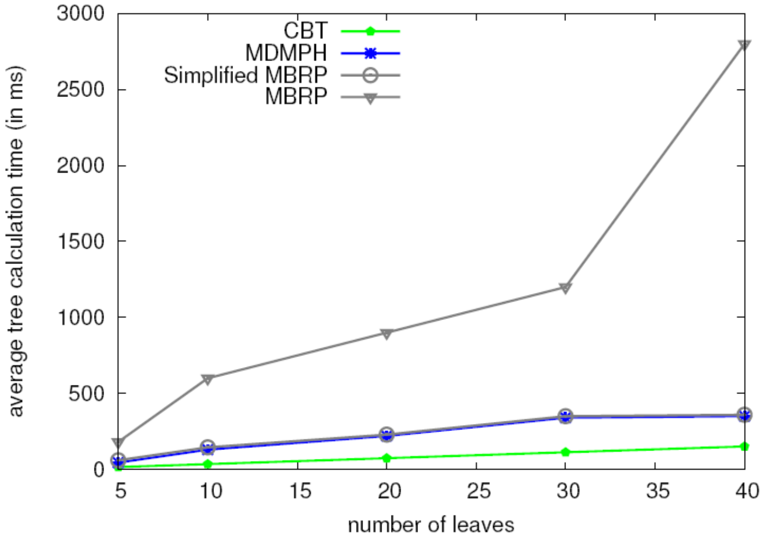
Figure 12 Average tree calculation time using MBRP (see online version for colours)



This is expected because as shown in Section 3, the number of combinations to calculate in the MBRP procedure increases exponentially as function of the number of leaves. This result indicates that the MBRP procedure is not useful for a large number of leaves. However, for a relatively small or medium number of leaves the calculation time remains acceptable for real-time tree calculation (less than three seconds for a number of leaves equal 40, see Figure 13). Hence, the MBRP procedure remains useful for inter-domain networks because the leaves there are routers (PEs) and not user terminals. Indeed, as the number of user terminals per multicast tree may increase rapidly, the number of PE leaves remains relatively small.

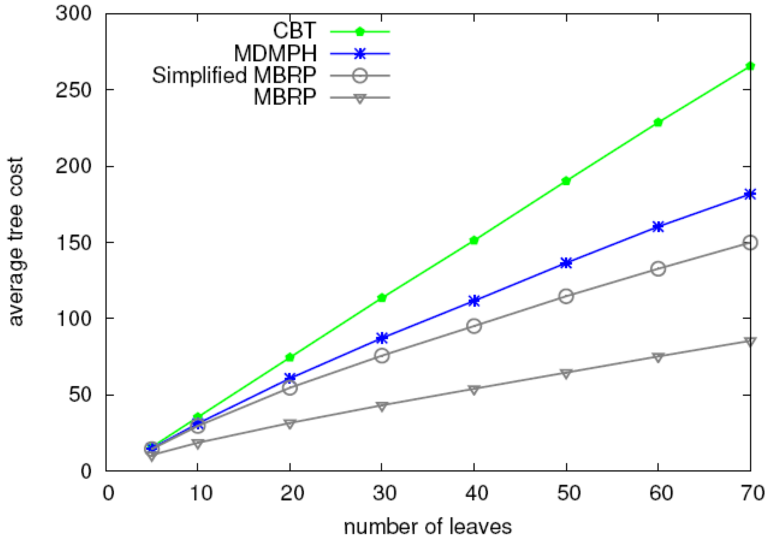
Figure 13 presents the average tree calculation time over the four studied algorithms. It can be shown that the time in the simplified MBRP procedure is in the same order of the CBT and MDMPH algorithms (less than 500 ms).

Figure 13 Comparison of average tree calculation time among different algorithms (see online version for colours)



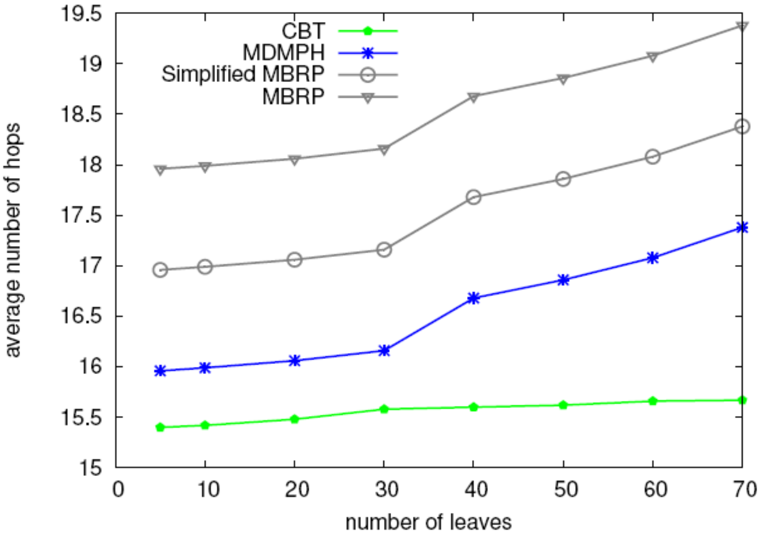
In Figure 14, we plot the average cost per tree. We conclude that the MBRP leads to the best cost and that the simplified MBRP leads to better cost than CBT and MDMPH. Hence, the simplified MBRP can be a good trade-off.

Figure 14 Average tree cost (see online version for colours)



Finally, in Figure 15, we present the average number of hops for each algorithm. Note that even MBRP and simplified MBRP lead to a higher number of hops, their values are still very close to that of CBT and MDMPH. Thus, the average delay between the tree source and a leaf for the four studied algorithms are of the same order.

Figure 15 Average number of hops (see online version for colours)



6 Conclusions

In this contribution, we provided a method in order to calculate an optimised inter-domain P2MP TE-LSP. The method is based on a recursive computation using PCE calculation servers. We assumed that the nodes of the P2MP TE-LSP are spread out over a tree of domains composed from a root domain, several transit and destination domains. Moreover, we supposed that the calculation starts on the destination domains and continues upstream until reaching the root domain. We highlighted the optimality/'calculation complexity' trade-off that aims at reducing the processing overhead on the PCEs at the expense of increasing the cost of the calculated P2MP TE-LSP. In addition, we provided a detailed signalling example showing how the inter-domain P2MP TE-LSP can be calculated and established. The provided method helps network operators to establish value added services such that multicast multimedia traffic while hiding the domains' topology between the different operators.

References

Aggarwal, R., Papadimitriou, D. and Yasukawa, S. (2007) *RFC 4875 – Extensions to Resource Reservation Protocol – Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*.

Andersson, L. and Madsen, T. (2005) *RFC 4026 – Provider Provisioned Virtual Private Network (VPN) Terminology* [online] https://tools.ietf.org/html/rfc4026 (accessed 2 March 2015).

Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and Swallow, G. (2001) *RFC 3209 – RSVP-TE Extensions to RSVP for LSP Tunnels* [online] https://tools.ietf.org/html/rfc3209 (accessed 2 March 2015)

ns-topogen [online] http://www.isi.edu/nsnam/ns/nstopogen.html (accessed 2 March 2015).

Takahashi, H. and Matsuyama, A. (1980) ‘An approximate solution for the Steiner problem in graphs’, *Math Jpn.*, Vol. 24, No. 6, pp.573–577.

Vasseur, J. and Le Roux, J. (2009) *RFC 5440 – Path Computation Element (PCE) Communication Protocol (PCEP)* [online] https://tools.ietf.org/html/rfc5440 (accessed 2 March 2015).

Vasseur, J., Ayyangar, A. and Zhang, R. (2008) *RFC 5152 – A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)* [online] https://tools.ietf.org/html/rfc5152 (accessed 2 March 2015).

Vasseur, J., Zhang, R., Bitar, N. and Le Roux, J. (2009) *RFC 5441 – A Backward Recursive PCE-based Computation (BRPC) Procedure to Compute Shortest Inter-domain Traffic Engineering Label Switched Paths* [online] https://tools.ietf.org/html/rfc5441 (accessed 2 March 2015).

Winter, P. (1987) ‘Steiner problem in networks a survey’, *Networks*, Vol. 17, No. 2, pp.129–167.

Wu, K., Zhao, Y., Yu, Z., Gu, W., Wang, D. and Cao, X. (2011) ‘A novel PCE-based algorithm for P2MP inter-domain traffic engineering in optical networks’, in *SPIE-OSA-IEEE Asia Communications and Photonics*, Vol. 8310, pp.1–7.

Zhao, Q., King, D., Ali, Z. and Casellas, R. (2014) *draft-ietf-pce-pcepinter-domain-p2mp-procedures-08 – PCE-Based Computation Procedure to Compute Shortest Constrained P2MP Inter-domain Traffic Engineering Label Switched Paths* [online] https://tools.ietf.org/html/draft-ietf-pce-pcep-inter-domain-p2mp-procedures-08 (accessed 2 March 2015).